RK611

DISKLESS CONTROL PART 1

CZR6ACO

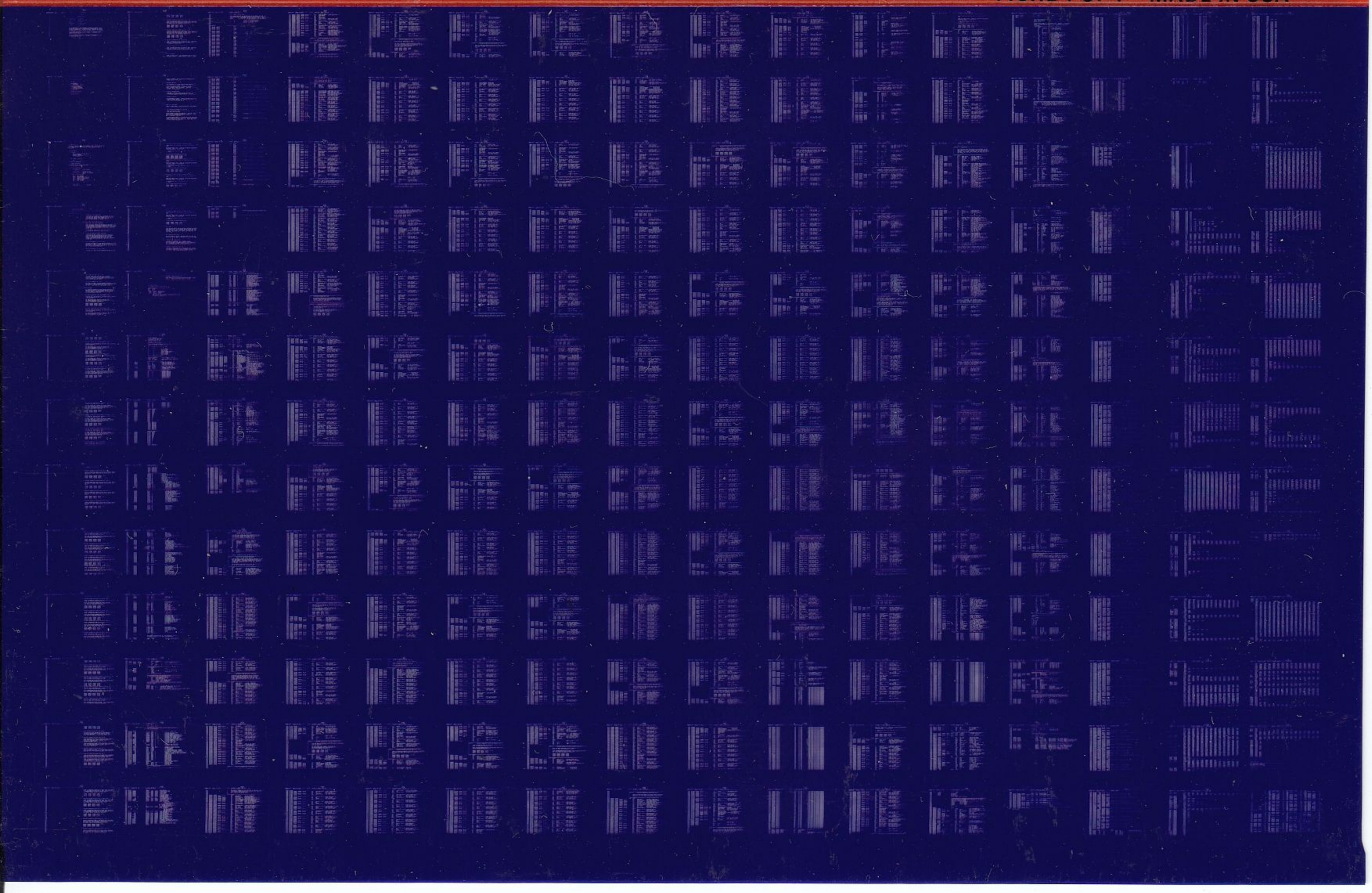
AH-9339C-MC

FICHE 1 OF 1

MAR 1978

digital

MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: AC-5338C-MC

PRODUCT NAME: CZR6ACO RK611 DSKLS CTR PRT 1

DATE: FEB 1978

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: ROY SPITZER

TABLE OF CONTENTS

```
                        1.0 ABSTRACT

                        THE RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 1 READS AND WRITES
                        EVERY RK611 REGISTER, TESTS THE INTERRUPT MECHANISM, AND TEST THE
                        SILO LOADING LOGIC.  NO RK06 DRIVE IS REQUIRED FOR PROGRAM
                        EXECUTION.


                        2.0 REQUIREMENTS

                        2.1 EQUIPMENT

                                PDP-11 SYSTEM (16K CORE MEMORY)
                                CONSOLE TERMINAL
                                DECTAPE, PAPER TAPE READER, OR DECDISK
                                RK611 CONTROLLER


                        2.2 PRELIMINARY PROGRAMS

                                NONE

                        3.0 OPERATING PROCEDURES

                        3.1 LOADING PROCEDURE

                        THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING ABSOLUTE LOADER
                        OR FROM XXDP MEDIA SUPPORTED BY XXDP.


                        3.2 STARTING PROCEDURE

                                LOCATION 200    - START PROGRAM

                                LOCATION 204    - RESTART PROGRAM

                                LOCATION 214    - REQUEST BUS ADDRESS, VECTOR ADDRESS, AND
                                                  PRIORITY MODIFICATION
                        3.3 OPTIONAL SWITCH SETTINGS

                                SW15 -  HALT PROGRAM
                                SW14 -  LOOP ON TEST
                                SW13 -  INHIBIT ERROR TYPE OUT
                                SW12 -  ABORT AFTER 20 ERRORS
                                SW11 -  INHIBIT ITERATION COUNT
                                SW10 -  BELL ON ERROR
                                SW9  -  LOOP ON ERROR
                                SW8  -  LOOP ON TEST IN SWITCHES 0-7


                        3.5 RUN TIME

                                FIRST PASS              7 SECONDS
                                SUBSEQUENT PASSES       2 MINUTES
```

4.0 OPERATING PROCEDURES

THE PROGRAM IS EXECUTED BY STARTING AT THE APPROPIATE ADDRESS.

5.0 PROGRAM DESCRIPTION

TEST 1 ADDRESS ALL RK611 REGISTERS

THIS TEST WILL ACCESS ALL RK611 REGISTERS AND CHECK TO MAKE SURE THAT NON-EXISTENT MEMORY DOES NOT OCCUR. A NON-EXISTENT MEMORY INDICATES EITHER THAT THE RK611 REGISTER BASE ADDRESS IS INCORRECT OR THAT THE RK611 DOES NOT RESPOND TO UNIBUS DIALOGUE.

**RESET, CONTROLLER CLEAR, AND TRI-STATE TESTS

TEST 2 RESET RK611 AND VERIFY REGISTERS

RESET THE RK611 CONTROLLER AND READ ALL REGISTER OF THE RK611 REGISTERS EXCEPT THE DATA BUFFER AND VERIFY THAT THEY ARE CORRECT. REEXAMINE COMMAND AND STATUS REGISTER TO MAKE SURE CONTROLLER ERROR DID NOT SET. REEXAMINE COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE DID NOT SET.

THE SUCCESSFUL EXECUTION OF THIS TEST VERIFIES THAT NO BIT OF THE TRI-STATE BUS IS STUCK TO ONE.

TEST 3 CONTROLLER CLEAR AND VERIFY REGISTERS

INITIALIZE THE RK611 CONTROLLER WITH A CONTROLLER CLEAR AND READ ALL REGISTER OF THE RK611 REGISTERS EXCEPT THE DATA BUFFER AND VERIFY THAT THEY ARE CORRECT. REEXAMINE COMMAND AND STATUS REGISTER 1 TO MAKE SURE CONTROLLER ERROR DID NOT SET. REEXAMINE COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE DID NOT SET.

TEST 4 WRITE BUS ADDRESS WITH 177777 (PART 1)

THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777 AND CHECK IF EQUAL TO 177776 AND THAT NO REGISTER INTERA TAKES PLACE. A RESET IS DONE AT THE END OF THE TEST TO MAKE SURE THE BUS ADDRESS CLEARS AND ALL RK611 REGISTERS ARE IN THEIR INITIALIZED STATE.

TEST 5 WRITE BUS ADDRESS WITH 177777 (PART 2)

THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777. A CONTROLLER CLEAR IS DONE. MAKE SURE THE BUS ADDRESS CL

```
                        TEST 6  WRITE WORD COUNT REG. WITH 177777

                                THIS TEST WILL WRITE THE WORD COUNT REGISTER
                                TO 0 AND 177777 AND CHECK IF EQUAL TO 0 AND 177777
                                RESPECTIVELY AND THAT NO REGISTER INTERACTION TAKES PLAC
                                ISSUE A CONTROLLER CLEAR AND MADE SURE THAT THE WORD
                                COUNT REGISTER DOES NOT CHANGE.


                        TEST 7  WRITE DISK ADDRESS WITH 177777

                                THIS TEST WILL WRITE THE DISK ADDRESS REGISTER TO 177777
                                AND CHECK IF EQUAL TO 003437 AND THAT NO REGISTER INTERA
                                TAKES PLACE. A CONTROLLER CLEAR IS DONE AT THE END OF
                                THE TEST TO MAKE SURE THE DISK ADDRESS CLEARS.


                **REGISTER INTERACTION TESTS

                                ALL REGISTER INTERACTION TESTS CONSISTS OF WRITING A
                                REGISTER AND CHECKING ITS CONTENTS AGAINST EXPECTED
                                CONTENTS. THEN ALL OTHER REGISTERS ARE READ EXCEPT
                                THE DATA BUFFER TO CHECK WHETHER THEY HAVE CHANGED FROM
                                THEIR INITIALIZED CONDITIONS.


                        TEST 10 REGISTER INTERACTION USING BUS ADDRESS (PART 1)

                                THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE
                                 CLEAR TO HE RK611 CONTROLLER. IT WILL THEN WRITE THE WO
                                COUNT REGISTER TO 0.

                                THE TEST ITSELF WILL CONSIST OF WRITING THE
                                BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A
                                TEST IF BUS ADDRESS IS CORRECT AND THAT NO
                                REGISTER INTERACTION TAKES PLACE.

                                000000  000010  000200  004000  100000
                                000001  000020  000400  010000
                                000002  000040  001000  020000
                                000004  000100  002000  040000


                TEST 11 REGISTER INTERACTION USING BUS ADDRESS (PART 2)

                                THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE
                                CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WO
                                COUNT REGISTER TO 0.

                                THE TEST ITSELF WILL CONSIST OF WRITING THE
                                BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A
                                TEST IF BUS ADDRESS IS CORRECT AND THAT NO
                                REGISTER INTERACTION TAKES PLACE.
```

```
                        177777  177767  177577  173777  077777
                        177776  177757  177377  167777
                        177775  177737  176777  157777
                        177773  177677  175777  137777


TEST 12 REGISTER INTERACTION USING BUS ADDRESS (PART 3)

        THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE
        CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WO
        COUNT REGISTER TO 0.

        THE TEST ITSELF WILL CONSIST OF WRITING THE
        BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A
        TEST IF BUS ADDRESS IS CORRECT AND THAT NO
        REGISTER INTERACTION TAKES PLACE.

        000001  000037  000777  017777  000000
        000003  000077  001777  037777
        000007  000177  003777  077777
        000017  000377  007777  177777
TEST 13 REGISTER INTERACTION USING BUS ADDRESS (PART 4)

        THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE
        CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WO
        COUNT REGISTER TO 0.

        THE TEST ITSELF WILL CONSIST OF WRITING THE
        BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A
        TEST IF BUS ADDRESS IS CORRECT AND THAT NO
        REGISTER INTERACTION TAKES PLACE.

        100000  174000  177600  177770  000000
        140000  176000  177700  177774
        160000  177000  177740  177776
        170000  177400  177760  177777


TEST 14 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)

        THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR
        BY WRITING THE WORD COUNT TO ZERO.

        THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
        AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
        SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
        THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
        REGISTER INTERACTION TAKES PLACE.

        000000  000020  000400  010000
        000002  000040  001000  020000
        000004  000100  002000  040000
        000010  000200  004000


TEST 15 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)
```

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR
BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
REGISTER INTERACTION TAKES PLACE.

077776	077756	077376	067776
077774	077736	076776	057776
077772	077676	075776	037776
077766	077576	073776	

TEST 16 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR
BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
REGISTER INTERACTION TAKES PLACE.

000002	000076	001776	037776
000006	000176	003776	077776
000016	000376	007776	000000
000036	000776	017776	

TEST 17 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR
BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
REGISTER INTERACTION TAKES PLACE.

000000	074000	077600	077770
040000	076000	077700	077774
060000	077000	077740	077776
070000	077400	077760	

TEST 20 REGISTER INTERACTION USING SPARE REG

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTE
WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE SPARE REGISTER WITH 177777 AND MAKE SURE

NO INTERACTION TAKES PLACE.

TEST 21 REGISTER INTERACTION USING WORD COUNT (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTE
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGU
AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 22 REGISTER INTERACTION USING WORD COUNT (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTE
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGU
AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 23 REGISTER INTERACTION USING WORD COUNT (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTE
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGU
AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 24 REGISTER INTERACTION USING WORD COUNT (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTE
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGU
AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	

```
                TEST 25 REGISTER INTERACTION USING DISK ADDRESS (PART 1)

                        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
                        WRITE THE WORD COUNT REGISTER WITH 0.

                        WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
                        THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AN
                        SURE NO INTERACTION TAKES PLACE.

                        000000  000010  000200  004000  100000
                        000001  000020  000400  010000
                        000002  000040  001000  020000
                        000004  000100  002000  040000


                TEST 26 REGISTER INTERACTION USING DISK ADDRESS (PART 2)

                        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
                        WRITE THE WORD COUNT REGISTER WITH 0.

                        WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
                        THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AN
                        SURE NO INTERACTION TAKES PLACE.

                        177777  177767  177577  173777  077777
                        177776  177757  177377  167777
                        177775  177737  176777  157777
                        177773  177677  175777  137777


                TEST 27 REGISTER INTERACTION USING DISK ADDRESS (PART 3)

                        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
                        WRITE THE WORD COUNT REGISTER WITH 0.

                        WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
                        THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AN
                        SURE NO INTERACTION TAKES PLACE.

                        000001  000037  000777  017777  000000
                        000003  000077  001777  037777
                        000007  000177  003777  077777
                        000017  000377  007777  177777
                TEST 30 REGISTER INTERACTION USING DISK ADDRESS (PART 4)

                        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
                        WRITE THE WORD COUNT REGISTER WITH 0.

                        WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
                        THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AN
                        SURE NO INTERACTION TAKES PLACE.


                        100000  174000  177600  177770  000000
                        140000  176000  177700  177774
                        160000  177000  177740  177776
```

```
                        170000  177400  177760  177777


TEST 31 REGISTER INTERACTION USING ATTN/OFFSET (PART 1)

        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
        WRITE THE WORD COUNT REGISTER WITH 0.

        WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
        WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTEN
        MAKE SURE NO INTERACTION TAKES PLACE.

        000000  000010  000200  004000  100000
        000001  000020  000400  010000
        000002  000040  001000  020000
        000004  000100  002000  040000


TEST 32 REGISTER INTERACTION USING ATTN/OFFSET (PART 2)

        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
        WRITE THE WORD COUNT REGISTER WITH 0.

        WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
        WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTEN
        MAKE SURE NO INTERACTION TAKES PLACE.

        177777  177767  177577  173777  077777
        177776  177757  177377  167777
        177775  177737  176777  157777
        177773  177677  175777  137777
TEST 33 REGISTER INTERACTION USING ATTN/OFFSET (PART 3)

        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
        WRITE THE WORD COUNT REGISTER WITH 0.

        WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
        WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTEN
        MAKE SURE NO INTERACTION TAKES PLACE.

        000001  000037  000777  017777  000000
        000003  000077  001777  037777
        000007  000177  003777  077777
        000017  000377  007777  177777


TEST 34 REGISTER INTERACTION USING ATTN/OFFSET (PART 4)

        ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTE
        WRITE THE WORD COUNT REGISTER WITH 0.

        WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
        WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTEN
        MAKE SURE NO INTERACTION TAKES PLACE.

        100000  174000  177600  177770  000000
```

```
                                        140000  176000  177700  177774
                                        160000  177000  177740  177776
                                        170000  177400  177760  177777


TEST 35 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1

        RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
        WRITE THE WORD COUNT REGISTER WITH ZERO.

        WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING C
        SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND ST
        REG. 2 AND CHECK FOR REGISTER INTERACTION.

        000000  000010  000400  010000
        000001  000020  001000  020000
        000002  000100  002000  040000
        000004  000200  004000  100000


TEST 36 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 2

        RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
        WRITE THE WORD COUNT REGISTER WITH ZERO.

        WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING C
        SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND ST
        REG. 2 AND CHECK FOR REGISTER INTERACTION.

        177737  177727  177337  167737
        177736  177717  176737  157737
        177735  177637  175737  137737
        177733  177537  173737  077737
TEST 37 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 3

        RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
        WRITE THE WORD COUNT REGISTER WITH ZERO.

        WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING C
        SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND ST
        REG. 2 AND CHECK FOR REGISTER INTERACTION.

        000001  000037  001737  037737
        000003  000137  003737  077737
        000007  000337  007737  177737
        000017  000737  017737  000000


TEST 40 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 4

        RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
        WRITE THE WORD COUNT REGISTER WITH ZERO.

        WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING C
        SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND ST
        REG. 2 AND CHECK FOR REGISTER INTERACTION.
```

```
                                100000  174000  177600  177734
                                140000  176000  177700  177736
                                160000  177000  177720  177737
                                170000  177400  177730  000000


TEST 41 CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS

        THIS TEST WILL TEST THE ABILITY OF THE SUBSYSTEM CLEAR T
        INITIALIZE THE BUS ADDRESS REGISTER AND COMMAND
        AND STATUS REGISTER 1.  IT WILL ALSO VERIFY THAT ALL
        OTHER REGISTERS REMAIN IN THE INITIALIZED STATE.


TEST 42 REGISTER INTERACTION USING DRIVE STATUS

        ISSUE A CONTROLLER CLEAR TO ININITALIZE RK611 REGISTERS.
        WRITE WORD COUNT TO 0, WRITE DRIVE STATUS REGISTER
        WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
        INTERACTION TAKES PLACE.
TEST 43 REGISTER INTERACTION USING ERROR REGISTER

        ISSUE A CONTRO  ER CLEAR TO INITIALIZE RK611 REGISTERS.
        WRITE THE WORD COUNT REGISTER TO ZERO. WRITE ERROR REGIS
        WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
        INTERACTION TAKES PLACE


TEST 44 REGISTER INTERACTION USING MAINT REG 2

        ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
        WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
        REGISTER 2 TO 177777 AND NO INTERACTION TAKES PLACE.


TEST 45 REGISTER INTERACTION USING MAINT REG 3

        ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
        WRITE THE WORD COUNT REGISTER TO ZERO, WRITE MAINTENANCE
        REGISTER 3 TO 177777 AND NO INTERACTION TAKES PLACE.
TEST 46 REGISTER INTERACTION WITH DISK CYLINDER (PART 1)

        ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
        WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
        CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHEC
        ARE CORRECT AND NO INTERACTION TAKES PLACE.

        000000  000010  000200  004000  100000
        000001  000020  000400  010000
        000002  000040  001000  020000
        000004  000100  002000  040000


TEST 47 REGISTER INTERACTION WITH DISK CYLINDER (PART 2)
```

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHEC
ARE CORRECT AND NO INTERACTION TAKES PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 50 REGISTER INTERACTION WITH DISK CYLINDER (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHEC
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 51 REGISTER INTERACTION WITH DISK CYLINDER (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHEC
ARE CORRECT AND NO INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	

TEST 52 REGISTER INTERACTION USING MAINT REG 1 (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 53 REGISTER INTERACTION USING MAINT REG 1 (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C
ARE CORRECT AND NO INTERACTION TAKES PLACE.

```
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
```

177777 177767 177577 173777 077777
177776 177757 177377 167777
177775 177737 176777 157777
177773 177677 175777 137777

TEST 54 REGISTER INTERACTION USING MAINT REG 1 (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000001 000037 000777 017777 000000
000003 000077 001777 037777
000007 000177 003777 077777
000017 000377 007777 177777

TEST 55 REGISTER INTERACTION USING MAINT REG 1 (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C
ARE CORRECT AND NO INTERACTION TAKES PLACE.

100000 174000 177600 177770 000000
140000 176000 177700 177774
160000 177000 177740 177776
170000 177400 177760 177777

TEST 56 REGISTER INTERACTION WITH PATTERN REG.

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE PATTER
REGISTER TO 177777 AND MAKE SURE IT REMAINS 0 AND
NO INTERACTION TAKES PLACE.

TEST 57 REGISTER INTERACTION WITH POSITION REG.

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE POSITI
REGISTER TO 177777 AND MAKE SURE IT STAYS AT THE
INITIALIZED CONDITION AND NO INTERACTION TAKES PLACE.

**INTERRUPT TESTS

TEST 60 RK611 INTERRUPT

STORE LOCATIONS 0-776, LOAD LOCATIONS 0-776 TO TRAP ALL
POSSIBLE INTERRUPTS. LOWER PROCESSOR PRIORITY TO ZERO.
MAKE SURE THAT NO INTERRUPT OCCURS. NOW SET INTERRUPT

```
                                        ENABLE AND READY.  VERIFY THAT THE INTERRUPT OCCURS AT
                                        PROPER VECTOR ADDRESS.  MAKE SURE THAT INTERRUPT IS
                                        CLEARED AFTER IT IS GIVEN.


                                TEST 61 INTERRUPT PRIORITY

                                        SET UP PRIORITY TO 1 LESS THAN INTERRUPT PRIORITY.
                                        WRITE READY WITH INTERRUPT ENABLE.  MAKE SURE INTERRUPT

                                        NOW SET UP PRIORITY EQUAL TO INTERRUPT PRIORITY.
                                        WRITE INTERRUPT ENABLE WITH READY.  MAKE SURE INTERRUPT
                                        DOES NOT  OCCUR.  NOW LOWER PRIOIRY AND MAKE
                                        INTERRUPT HAS BEEN STORED.


                                TEST 62 SETTING INTERRUPT ENABLE

                                        CLEAR RK611 CONTROLLER WITH CONTROLLER CLEAR. ALLOW RK61
                                        INTERRUPTS BY SETTING PROCESSOR PRIORITY TO ZERO.
                                        SET INTERRUPT ENABLE AND MAKE SURE NO INTERRUPTS OCCUR.


                                TEST 63 INTERRUPT CLEARING

                                        SET UP PRIORITY TO SEVEN.  CREATE INTERRUPT BY SETTING
                                        INTERRUPT ENABLE  READY.  AND CLEAR IT WITH CONTROLLER
                                        CLEAR. SET INTERRUPT ENABLE.  NOW LOWER PRIORITY
                                        TO MAKE SURE NO INTERRUPT OCCURS.


                        **SILO  TESTS

                                TEST 64 READ SILO WHEN EMPTY

                                        READ SILO WHEN EMPTY.  CHECK FOR DATA LATE AND CONTROLLE
                                        ERROR.  ISSUE CONTROLLER CLEAR AND CHECK IF ERROR RESET.


                                TEST 65 SILO LOADING AND UNLOADING OF ONE WORD

                                        ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 CONTROLLER.
                                        CLEAR WORD COUNT REGISTER.

                                        WRITE A WORD OF 177777 INTO THE SILO.  CHECK ALL OTHER
                                        REGISTERS FOR INTERACTION PROBLEMS.  CHECK THAT OUTPUT
                                        READY IS SET IN COMMAND AND STATUS REGISTER 2.  IF NOT
                                        WAIT A REASONABLE TIME FOR IT.

                                        IF OUTPUT READY COMES UP IN A REASONABLE TIME, READ BACK
                                        CONTENTS AND MAKE SURE IT IS 177777.  CHECK FOR NO CONTR
                                        ERROR, NO DATA LATE, INPUT READY SET, OUTPUT READY RESET
                                        NOW READ ANOTHER WORD FROM THE SILO TO MAKE SURE DATA
                                        LATE AND CONTROLLER ERROR SET.
```

```
TEST 66 ONE WORD SILO WRITE AND REG. INTERACTION (PART 1)

        ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
        ZERO.

        WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERAC
        PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
        STATUS REGISTER 2.  IF NOT WAIT.

        IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
        IF CORRECT.


        THE FOLLOWING CONFIGURATIONS ARE USED:

        000000  000010  000200  004000  100000
        000001  000020  000400  010000
        000002  000040  001000  020000
        000004  000100  002000  040000
TEST 67 ONE WORD SILO WRITE AND REG. INTERACTION (PART 2)

        ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
        ZERO.

        WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERAC
        PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
        STATUS REGISTER 2.  IF NOT WAIT.

        IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
        IF CORRECT.

        THE FOLLOWING CONFIGURATIONS ARE USED:

        177777  177767  177577  173777  077777
        177776  177757  177377  167777
        177775  177737  176777  157777
        177773  177677  175777  137777

TEST 70 ONE WORD SILO WRITE AND REG. INTERACTION (PART 3)

        ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
        ZERO.

        WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERAC
        PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
        STATUS REGISTER 2.  IF NOT WAIT.

        IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
        IF CORRECT.

        THE FOLLOWING CONFIGURATIONS ARE USED:

        000001  000037  000777  017777  000000
        000003  000077  001777  037777
        000007  000177  003777  077777
```

```
                                000017  000377  007777  177777


TEST 71 ONE WORD SILO WRITE AND REG. INTERACTION (PART 4)

        ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
        ZERO.

        WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERAC
        PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
        STATUS REGISTER 2.  IF NOT WAIT.

        IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
        IF CORRECT.

        THE FOLLOWING CONFIGURATIONS ARE USED:

        100000  174000  177600  177770  000000
        140000  176000  177700  177774
        160000  177000  177740  177776
        170000  177400  177760  177777


TEST 72 SILO FILL

        THIS TEST WILL WRITE THE SILO WITH 66 DIFFERENT PATTERNS
        CHECK INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH
        WORD WRITTEN.  IT WILL THEN READ ALL 66 WORDS BACK CHECK
        CONTENTS, INPUT READY, OUTPUT READY, AND DATA LATE FOR E
        WORD READ.  AN EXTRA READ IS THEN DONE TO MAKE SURE THE
        SILO IS EMPTY.


TEST 73 SILO CAPACITY DATA LATE

        WRITE 67 WORDS IN THE SILO AND MAKE SURE DATA LATE ONLY
        OCCURS ON THE 67TH WORD.  CLEAR RK611 WITH CONTROLLER CL
        CHECK INPUT READY AND OUTPUT READY FOR INITIALIZED STATE


TEST 74 INTERRUPT DUE TO DATA LATE

        ALLOW RK611 INTERRUPTS.  SET INTERRUPT ENABLE.
        NOW READ ONE WORD FROM DATA BUFFER AND MAKE SURE THAT
        DATA LATE CAUSES INTERRUPT.  BEFORE CLEARING ERROR ALLOW
        RK611 INTERRUPTS AND MAKE SURE IT DOES NOT OCCUR AGAIN.
        NOW CLEAR CONTROLLER WITH A CONTROLLER CLEAR.


TEST 75 INTERRUPT CLEARING AND DATA LATE

        CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. CREATE
        A CONTROLLER ERROR (DATA LATE) BY READING THE DATA BUFFE
        WHEN EMPTY.  CLEAR RK611 CONTROLLER WITH A CONTROLLER CL
        SET INTERRUPT ENABLE AND LOWER PROCESSOR PRIORITY.
```

```
                                        MAKE SURE AN INTERRUPT DOES NOT OCCUR.


                                TEST 76 INTERRUPT ENABLE AND DATA LATE

                                        CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.  ALLOW
                                        RK611 INTERRUPTS.  READ DATA BUFFER TO GENERATE INTERRUP
                                        PENDING.  MAKE SURE INTERRUPT DOES NOT OCCUR.

                                        NOW SET INTERRUPT ENABLE AND MAKE SURE INTERRUPTS OCCURS
```

6.0 ERROR REPORTING

THE GENERAL FORMAT OF ERROR REPORTS IS:

```
OPERATION DESCRIPTION AND ERROR DESCRIPTION
TEST     ERROR
NUM      PC
XXXXXX   YYYYYY
  EXPECT   ACTUAL   OTHER PERTENANT
  REG      REG      INFORMATION
  ZZZZZZ   WWWWWW   AAAAAA
```

NOTE: MORE THAN ONE SET OF EXPECT/ACTUAL REGISTERS MAY BE PRINTED OUT. OTHER PERTENANT INFORMATION MAY CONSIST OF MORE THAN ONE WORD.

[END OF DOCUMENT]

%

```
                        ;*** REV 003 ***
                        .TITLE  CZR6ACO RK611 DSKLS CTRL PRT1
                        ;*COPYRIGHT (C) 1976,1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ROY SPITZER
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            ABORT PROGRAM AFTER 20 ERRORS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
```

```
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7


                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  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)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
```

```
1040                                    .EQUIV  BIT06,BIT6
1041                                    .EQUIV  BIT05,BIT5
1042                                    .EQUIV  BIT04,BIT4
1043                                    .EQUIV  BIT03,BIT3
1044                                    .EQUIV  BIT02,BIT2
1045                                    .EQUIV  BIT01,BIT1
1046                                    .EQUIV  BIT00,BIT0
1047
1048                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1049            000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1050            000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1051            000014                  TBITVEC=14              ;;"T" BIT
1052            000014                  TRTVEC= 14              ;;TRACE TRAP
1053            000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1054            000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1055            000024                  PWRVEC= 24              ;;POWER FAIL
1056            000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1057            000034                  TRAPVEC=34              ;;"TRAP" TRAP
1058            000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
1059            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1060            000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1061            000114                  MEMVEC= 114                     ;MEMORY PARITY VECTOR
1062            172100                  MEMBAS= 172100                  ;MEMORY PARITY BASE ADDDRESS
1063            000001                  PAR.EN= 1                       ;ALLOW MEMORY PARITY OPTION
1064            120210                  AVECT1= 120210                  ;DEFINE RK611 VECTOR ADDRESS
1065            000005                  APRIOR= 5                       ;DEFINE RK611 PRIORITY
1066            177440                  ABASE=  177440                  ;DEFINE BASE OF RK611 REGISTERS
1067
1068                                    .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
1069
1070            000000                  RKCS1=  0                       ;CONTROL AND STATUS REGISTER 1
1071            000002                  RKWC=   2                       ;WORD COUNT REGISTER
1072            000004                  RKBA=   4                       ;BUS ADDRESS REGISTER
1073            000006                  RKDA=   6                       ;DESIRED TRACK SECTOR REGISTER
1074            000010                  RKCS2=  10                      ;CONTROL AND STATUS REGISTER 2
1075            000012                  RKDS=   12                      ;DRIVE STATUS REGISTER
1076            000014                  RKER=   14                      ;ERROR REGISTER
1077            000016                  RKASOF= 16                      ;ATTENTION SUMMARY AND OFFSET REGISTER
1078            000020                  RKDCYL= 20                      ;DESIRED CYLINDER REGISTER
1079            000024                  RKDB=   24                      ;DATA BUFFER
1080            000026                  RKMR1=  26                      ;MAINTENANCE REGISTER 1
1081            000034                  RKMR2=  34                      ;MAINTENANCE REGISTER 2
1082            000036                  RKMR3=  36                      ;MAINTENANCE REGISTER 3
1083            000030                  RKECPS= 30                      ;ECC POSITION INFORMATION
1084            000032                  RKECPT= 32                      ;ECC PATTERN INFORMATION
1085            000022                  RKSPAR= 22                      ;SPARE REGISTER
1086
1087                                    .SBTTL  DRIVE COMMANDS
1088
1089            000001                  SELDRV= 01                      ;SELECT DRIVE
1090            000003                  PACK=   03                      ;PACK ACKNOWLEDGE
1091            000005                  CLEAR=  05                      ;DRIVE CLEAR
1092            000007                  UNLOAD= 07                      ;UNLOAD
1093            000011                  SRTSPL= 11                      ;START SPINDLE
1094            000013                  RECAL=  13                      ;RECALIBRATE
1095            000015                  OFFSET= 15                      ;OFFSET
```

```
                        DRIVE COMMANDS

1096    000017          SEEK=   17                      ;SEEK
1097    000021          RDDATA= 21                      ;READ DATA
1098    000023          WRDATA= 23                      ;WRITE DATA
1099    000025          RDHEAD= 25                      ;READ HEADER
1100    000027          WRHEAD= 27                      ;WRITE HEADER AND DATA
1101    000031          WRTCHK= 31                      ;WRITE CHECK
1102    000300          INTR=   300                     ;GENERATE INTERRUPT TO CPU
1103
1104                    .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
1105
1106    000001          GO=     BIT0                    ;GO BIT
1107    000100          IE=     BIT6                    ;INTERRUPT ENABLE
1108    000200          RDY=    BIT7                    ;CONTROLLER READY
1109    000400          BA16=   BIT8                    ;BUS ADDRESS BIT 16
1110    001000          BA17=   BIT9                    ;BUS ADDRESS BIT 17
1111    002000          CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06)
1112    004000          CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
1113                                                    ;  DRIVE RESPONSE
1114    010000          CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
1115    020000          SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
1116    040000          DI=     BIT14                   ;DRIVE INTERRUPT
1117    100000          CERR=   BIT15                   ;CONTROLLER ERROR
1118    100000          CCLR=   BIT15                   ;CONTROLLER CLEAR
1119
1120                    .SBTTL  CONTROL AND STATUS REGISTER 2 BITS
1121
1122    000007          DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
1123    000010          RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1124    000020          BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
1125    000040          SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
1126    000100          IR=     BIT6                    ;INPUT READY
1127    000200          OR=     BIT7                    ;OUTPUT READY
1128    000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1129    001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1130    002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1131    004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1132    010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1133    020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1134    040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1135    100000          DLT=    BIT15                   ;DATA LATE ERROR
1136
1137                    .SBTTL  ERROR REGISTER BIT DEFINITION
1138
1139    000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1140    000002          SKI=    BIT1                    ;SEEK INCOMPLETE
1141    000004          NXF=    BIT2                    ;NON-EXECUTABLE DRIVE FUNCTION
1142    000010          DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
1143    000020          FMTE=   BIT4                    ;FORMAT ERROR
1144    000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1145    000100          ECH=    BIT6                    ;ECC HARD
1146    000200          BSE=    BIT7                    ;BAD SECTOR ERROR
1147    000400          HVRC=   BIT8                    ;HEADER VRC ERRROR
1148    001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1149    002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
1150    004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1151    010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
```

```
                        ERROR REGISTER BIT DEFINITION

  1152  020000          OPI=     BIT13                ;OPERATION (SEARCH) INCOMPLETE
  1153  040000          UNS=     BIT14                ;DRIVE UNSAFE
  1154  100000          DCK=     BIT15                ;DATA CHECK
  1155
  1156                  .SBTTL   STATUS REGISTER BIT DEFINITION
  1157
  1158  000001          DRA=     BIT0                 ;DRIVE AVAILABLE (CONTROLLER IS SET IF
  1159                                                ;  THIS BIT IS RESET)
  1160  000004          OFST=    BIT2                 ;DRIVE OFFSET
  1161  000010          ACLO=    BIT3                 ;AC LOW
  1162  000020          SPDLSS=  BIT4                 ;SPEED LOSS
  1163  000040          DROT=    BIT5                 ;DRIVE OFF TRACK
  1164  000100          VV=      BIT6                 ;VOLUME VALID
  1165  000200          DRDY=    BIT7                 ;DRIVE READY
  1166  000400          DDT=     BIT8                 ;DRIVE TYPE (0=RK06)
  1167  004000          WRL=     BIT11                ;WRITE LOCK
  1168  020000          PIP=     BIT13                ;POSITIONING IN PROGRESS
  1169  040000          DSC=     BIT14                ;DRIVE STATUS CHANGE
  1170  100000          SVAL=    BIT15                ;STATUS VALID
  1171
  1172                  .SBTTL   MAINTENANCE REGISTER 1 BIT DEFINITION
  1173
  1174  000017          MESMSK=  17                   ;MESSAGE MASK
  1175
  1176  000020          PAT=     BIT4                 ;FORCE EVEN PARITY ON DRIVE MESSAGE LINES
  1177  000040          DMD=     BIT5                 ;DIAGNOSTIC MODE
  1178  000100          MSP=     BIT6                 ;MAINTENANCE SECTOR PULSE
  1179  000200          MIND=    BIT7                 ;MAINTENANCE INDEX
  1180  000400          MCLK=    BIT8                 ;MAINTENANCE CLOCK
  1181  001000          MERD=    BIT9                 ;MAINTENANCE ENCODED READ DATA
  1182  002000          MEWD=    BIT10                ;MAINTENACNE ENCODED WRITE DATA
  1183  004000          PCA=     BIT11                ;PRECOMPENSATION ADVANCE
  1184  010000          PCD=     BIT12                ;PRECOMPENSATION DELAY
  1185  020000          ECCW=    BIT13                ;ECC WORD IS BEING READ OR WRITTEN
  1186  040000          WRTGAT=  BIT14                ;WRITE GATE
  1187  100000          RDGATE=  BIT15                ;READ GATE
  1188
  1189                  .SBTTL   TRANSMITTED MESSAGE A
  1190
  1191  000020          S.SEEK=  BIT4                 ;SEEK COMMAND
  1192  000040          S.RECL=  BIT5                 ;RECALIBRATE COMMAND
  1193  000100          S.STSP=  BIT6                 ;START SPINDLE COMMAND
  1194  000200          S.RTC=   BIT7                 ;DRIVE RETURN TO CENTERLINE COMMAND
  1195  000400          S.CLR=   BIT8                 ;CLEAR ERROR AND DSC
  1196  001000          S.FMT=   BIT9                 ;FORMAT
  1197  002000          S.UNLD=  BIT10                ;UNLOAD
  1198  004000          S.PACK=  BIT11                ;SET VOLUME VALID (PACK ACKNOWNLEDGE)
  1199                  .SBTTL   TRAP CATCHER
  1200
  1201  000000                   .=0
  1202                  ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
  1203                  ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
  1204                  ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
  1205  000174                   .=174
  1206  000174  000000  DISPREG: .WORD  0              ;;SOFTWARE DISPLAY REGISTER
  1207  000176  000000  SWREG:   .WORD  0              ;;SOFTWARE SWITCH REGISTER
```

```
1208
1209  000200  000137  002050          .SBTTL  STARTING ADDRESS(ES)
                                      JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1210  000204  000137  002040          JMP     RESTRT          ;JUMP TO RESTART ROUTINE
1211          000214                  .=214
1212  000214  000137  002030          JMP     PARM            ;JUMP TO OPERATOR ASSIGNED PARMETERS
1213                                  .SBTTL  ACT11 HOOKS
1214
1215                                  ;;*****************************************************************
1216                                  ;HOOKS REQUIRED BY ACT11
1217          000220                  $SVPC=.                 ;SAVE PC
1218          000046                  .=46
1219  000046  055244                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1220          000052                  .=52
1221  000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
1222          000220                  .=$SVPC                 ;; RESTORE PC
1223          001000                  .=1000
1224                                  .SBTTL  APT PARAMETER BLOCK
1225
1226                                  ;;*****************************************************************
1227                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1228                                  ;;*****************************************************************
1229          001000                  .$X=.   ;;SAVE CURRENT LOCATION
1230          000024                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1231  000024  000200                  200     ;;FOR APT START UP
1232          000044                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1233  000044  001000                  $APTHDR ;;POINT TO APT HEADER BLOCK
1234          001000                  .=.$X   ;;RESET LOCATION COUNTER
1235                                  ;;*****************************************************************
1236                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1237                                  ;INTERFACE SPEC.
1238
1239  001000                          $APTHD:
1240  001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1241  001002  001214                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1242  001004  000001                  $TSTM:  .WORD   1       ;;RUN TIM OF LONGEST TEST
1243  001006  000007                  $PASTM: .WORD   7       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1244  001010  000007                  $UNITM: .WORD   7       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1245  001012  000032                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPCCR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001166  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001170  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001172  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001174  000000          $TMP6:  .WORD   0               ;;USER DEFINED
001176  000000          $TMP7:  .WORD   0               ;;USER DEFINED
001200  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001202  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001204  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001210     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001211     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001212  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
```

```
1302  001214                  $MAIL:                          ;;APT MAILBOX
1303  001214  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
1304  001216  000000          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
1305  001220  000000          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
1306  001222  000000          $PASS:  .WORD   APASS           ;;PASS COUNT
1307  001224  000000          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
1308  001226  000000          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
1309  001230  000000          $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
1310  001232  000000          $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
1311  001234                  $ETABLE:                        ;;APT ENVIRONMENT TABLE
1312  001234     000          $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
1313  001235     000          $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
1314  001236  000000          $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
1315  001240  000000          $USWR:  .WORD   AUSWR           ;;USER SWITCHES
1316  001242  000000          $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
1317                                                          BITS 15-11=CPU TYPE
1318                                                          ;*
1319                                                          ;*
1320                                                          ;*
1321                                                          BIT 10=REAL TIME CLOCK
1322                                                          BIT  9=FLOATING POINT PROCESSOR
1323  001244     000          $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
1324  001245     000          $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
1325                                                          MEM.TYPE BYTE    --  (HIGH BYTE)
1326                                                          ;*
1327                                                          ;*
1328                                                          ;*
1329  001246  000000          $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
1330                                                          ;*
1331  001250     000          $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
1332  001251     000          $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
1333  001252  000000          $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
1334  001254     000          $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
1335  001255     000          $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
1336  001256  000000          $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
1337  001260     000          $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
1338  001261     000          $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
1339  001262  000000          $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
1340  001264  120210          $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1341  001266  000000          $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1342  001270  177440          $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1343  001272  000000          $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
1344  001274  000000          $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
1345  001276  000000          $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
1346  001300                  $ETEND:
1347                          .MEXIT
```

```
                                .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


001300                          $ERRTB:
                                ;       ERROR 1:  NEM ACCESSING RK611 REGISTER
001300  063251                          EM1
001302  000000                          0
001304  061416                          DT001
001306  061562                          DF001
                                ;       ERROR 2:   ATTEMPTING TO INITIALIZE RK611
001310  000000                  EM2N:   0
001312  000000                          0
001314  061424                          DT002
001316  061602                          DF002
                                ;       ERROR 3:   ATTEMPTING TO WRITE REG
001320  000000                  EM3N:   0
001322  000000                          0
001324  061434                          DT003
001326  061626                          DF003
                                ;       ERROR 4:   ATTEMPTING TO CLEAR RK611
001330  000000                  EM4N:   0
001332  000000                          0
001334  061446                          DT004
001336  061652                          DF004
                                ;       ERROR 5:        INTERRUPT DID NOT OCCUR
001340  064766                          EM24
001342  000000                          0
001344  061460                          DT005
001346  061676                          DF005
                                ;       ERROR 6:        UNEXPECTED INTERRUPT WHEN LOWERING PROCESSOR PRIORITY
001350  065060                          EM25
001352  000000                          0
001354  061464                          DT006
001356  061712                          DF006
                                ;       ERROR 7:        VECTOR ADDRESS INCORRECT
001360  065140                          EM26
001362  000000                          0
001364  061464                          DT006
001366  061712                          DF006
                                ;       ERROR 10:       INTERRUPT DID NOT CLEAR IN RK611
001370  065176                          EM27
001372  000000                          0
001374  061464                          DT006
001376  061712                          DF006
                                ;       ERROR 11:       EXPECTED INTERRUPT DID NOT OCCUR AT PROCESSOR PRIORITY
```

```
1404  001400  065232                    EM28
1405  001402  000000                    0
1406  001404  061474                    DT011
1407  001406  061736                    DF011
1408                          ;         ERROR 12:       UNEXPECTED INTERRUPT OCCURRED AT PROCESSOR PRIORITY
1409  001410  065314                    EM29
1410  001412  000000                    0
1411  001414  061474                    DT011
1412  001416  061736                    DF011
1413                          ;         ERROR 13:       INTERRUPT DID NOT OCCUR WHEN PRIORITY LOWERED
1414  001420  065373                    EM30
1415  001422  000000                    0
1416  001424  061474                    DT011
1417  001426  061736                    DF011
1418                          ;         ERROR 14:       SETTENG INTERRUPT ENABLE CAUSED EXPECTED INTERRUPT
1419  001430  065444                    EM31
1420  001432  000000                    0
1421  001434  061460                    DT005
1422  001436  061676                    DF005
1423                          ;         ERROR 15:       CONTROLLER CLEAR DID NOT CLEAR STORED INTERRUPT
1424  001440  065515                    EM32
1425  001442  000000                    0
1426  001444  061460                    DT005
1427  001446  061676                    DF005
1428                          ;         ERROR 16:       ATTEMPTING TO READ SILO WHEN EMPTY - CS2 INCORRECT
1429  001450  065632                    EM34
1430  001452  067072                    EM1009
1431  001454  061502                    DT016
1432  001456  061756                    DF016
1433                          ;         ERROR 17:       ATTEMPTING TO READ SILO WHEN EMPTY - CS1 INCORRECT
1434  001460  065632                    EM34
1435  001462  066574                    EM1000
1436  001464  061502                    DT016
1437  001466  061756                    DF016
1438                          ;         ERROR 20:       ATTEMPTING TO CLEAR DATA LATE - CS' INCORRECT
1439  001470  065675                    EM35
1440  001472  066574                    EM1000
1441  001474  061502                    DT016
1442  001476  061756                    DF016
1443                          ;         ERROR 21:       ATTEMPTING TO CLEAR DATA LATE - CS2 INCORRECT
1444  001500  065675                    EM35
1445  001502  066704                    EM1004
1446  001504  061502                    DT016
1447  001506  061756                    DF016
1448                          ;         ERROR 22:       ATTEMPTING TO READ SILO CONTAINING ONE WORD
1449                          ;                         CS1 INCORRECT
1450  001510  065733                    EM36
1451  001512  066574                    EM1000
1452  001514  061502                    DT016
1453  001516  061756                    DF016
1454                          ;         ERROR 23:       ATTEMPTING TO READ SILO CONTAINING ONE WORD
1455                          ;                         CS2 INCORRECT
1456  001520  065733                    EM36
1457  001522  066704                    EM1004
1458  001524  061502                    DT016
1459  001526  061756                    DF016
```

```
1460                                    ;       ERROR 24:       ATTEMPTING TO LOAD SILO - CS1 INCORRECT
1461  001530  066007                            EM37
1462  001532  066574                            EM1000
1463  001534  061516                            DT024
1464  001536  062002                            DF024
1465                                    ;       ERROR 25:       ATTEMPTING TO LOAD SILO - CS2 INCORRECT
1466  001540  066007                            EM37
1467  001542  066704                            EM1004
1468  001544  061516                            DT024
1469  001546  062002                            DF024
1470                                    ;       ERROR 26:       ATTEMPTING TO READ SILO - CS1 INCORRECT
1471  001550  066037                            EM38
1472  001552  066574                            EM1000
1473  001554  061534                            DT026
1474
1475  001556  062026                            DF026
1476                                    ;       ERROR 27:       ATTEMPTING TO READ SILO - CS2 INCORRECT
1477  001560  066037                            EM38
1478  001562  066704                            EM1004
1479  001564  061534                            DT026
1480  001566  062026                            DF026
1481                                    ;       ERROR 30:       ATTEMPTING TO READ SILO - DATA INCORRECT
1482  001570  066037                            EM38
1483  001572  067044                            EM1008
1484  001574  061534                            DT026
1485  001576  062026                            DF026
1486                                    ;       ERROR 31:       ATTEMPTING TO LOAD FULL SILO - CS2 INCORRECT
1487  001600  066067                            EM39
1488  001602  066704                            EM1004
1489  001604  061502                            DT016
1490  001606  061756                            DF016
1491                                    ;       ERROR 32:       ATTEMPTING TO LOAD FULL SILO - CS1 INCORRECT
1492  001610  065314                            EM29
1493  001612  066574                            EM1000
1494  001614  061502                            DT016
1495  001616  061756                            DF016
1496                                    ;       ERROR 33:       DATA LATE DID NOT CAUSE EXPECTED INTERRUPT
1497  001620  066137                            EM40
1498  001622  000000                            0
1499  001624  061460                            DT005
1500  001626  061676                            DF005
1501                                    ;       ERROR 34:       UNEXPECTED INTERRUPT DUE TO UNCLEARED CONTROLLER ERROR
1502  001630  066205                            EM41
1503  001632  000000                            0
1504  001634  061460                            DT005
1505  001636  061676                            DF005
1506                                    ;       ERROR 35:       CONTROLLER CLEAR DID NOT CLEAR PENDING INTERRUPT DUE
1507                                    ;                       TO CONTROLLER ERROR
1508  001640  066267                            EM42
1509  001642  000000                            0
1510  001644  061460                            DT005
1511  001646  061676                            DF005
1512                                    ;       ERROR 36:       CONTROLLER ERROR CAUSED INTERRUPT WITH
1513                                    ;                       INTERRUPT ENABLE RESET
1514  001650  066374                            EM43
1515  001652  000000                            0
```

E03


```
1516  001654  061460                    DT005
1517  001656  061676                    DF005
1518                          ;         ERROR 37:       INTERRUPT DID NOT OCCUR WHEN INTERRUPT ENABLE
1519                          ;                         SET WITH INTERRUPT PENDING DUE TO DATA LATE
1520  001660  066460                    EM44
1521  001662  062251                    DH000A
1522  001664  061464                    DT006
1523  001666  061712                    DF006
1524                          ;         ERROR 40:       UNEXPECTED MEMORY PARITY ENABLE TRAP
1525  001670  063204                    EM000
1526  001672  062302                    DH000C
1527  001674  061412                    DT000
1528  001676  061556                    DF000
```

```
                                        TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER

1529                                    .SBTTL  TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
1530
1531  001700  000000                    T.CS1:  .WORD   0               ;CONTROL AND STATUS REGISTER 1
1532  001702  000000                    T.WC:   .WORD   0               ;WORD COUNT REGISTER
1533  001704  000000                    T.BA:   .WORD   0               ;BUS ADDRESS REGISTER
1534  001706  000000                    T.DA:   .WORD   0               ;DESIRED TRACK SECTOR REGISTER
1535  001710  000000                    T.CS2:  .WORD   0               ;CONTROL AND STATUS REGISTER 2
1536  001712  000000                    T.DS:   .WORD   0               ;DRIVE STATUS REGISTER
1537  001714  000000                    T.ER:   .WORD   0               ;ERROR REGISTER
1538  001716  000000                    T.ASOF: .WORD   0               ;ATTENTION SUMMARY AND OFFSET REGISTER
1539  001720  000000                    T.DCYL: .WORD   0               ;DESIRED CYLINDER REGISTER
1540  001722  000000                    T.DB:   .WORD   0               ;DATA BUFFER
1541  001724  000000                    T.MR1:  .WORD   0               ;MAINTENANCE REGISTER 1
1542  001726  000000                    T.MR2:  .WORD   0               ;MAINTENANCE REGISTER 2
1543  001730  000000                    T.MR3:  .WORD   0               ;MAINTENANCE REGISTER 3
1544  001732  000000                    T.ECPS: .WORD   0               ;ECC POSITION INFORMATION
1545  001734  000000                    T.ECPT: .WORD   0               ;ECC PATTERN INFORMATION
1546  001736  000000                    T.SPAR: .WORD   0               ;SPARE REGISTER
1547
1548                                    .SBTTL  EXPECTED RK611 CONTROLLER REGISTERS
1549
1550  001740  000000                    E.CS1:  .WORD   0               ;CONTROL AND STATUS REGISTER 1
1551  001742  000000                    E.WC:   .WORD   0               ;WORD COUNT REGISTER
1552  001744  000000                    E.BA:   .WORD   0               ;BUS ADDRESS REGISTER
1553  001746  000000                    E.DA:   .WORD   0               ;DESIRED TRACK SECTOR REGISTER
1554  001750  000000                    E.CS2:  .WORD   0               ;CONTROL AND STATUS REGISTER 2
1555  001752  000000                    E.DS:   .WORD   0               ;DRIVE STATUS REGISTER
1556  001754  000000                    E.ER:   .WORD   0               ;ERROR REGISTER
1557  001756  000000                    E.ASOF: .WORD   0               ;ATTENTION SUMMARY AND OFFSET REGISTER
1558  001760  000000                    E.DCYL: .WORD   0               ;DESIRED CYLINDER REGISTER
1559  001762  000000                    E.DB:   .WORD   0               ;DATA BUFFER
1560  001764  000000                    E.MR1:  .WORD   0               ;MAINTENANCE REGISTER 1
1561  001766  000000                    E.MR2:  .WORD   0               ;MAINTENANCE REGISTER 2
1562  001770  000000                    E.MR3:  .WORD   0               ;MAINTENANCE REGISTER 3
1563  001772  000000                    E.ECPS: .WORD   0               ;ECC POSITION INFORMATION
1564  001774  000000                    E.ECPT: .WORD   0               ;ECC PATTERN INFORMATION
1565  001776  000000                    E.SPAR: .WORD   0               ;SPARE REGISTER
1566
1567                                    .SBTTL  PROGRAM DEFINED VARIABLES
1568
1569  002000  000210                    RkVEC:  .WORD   210             ;RK611 VECTOR ADDRESS
1570  002002  000240                    RKPRI:  .WORD   PR5             ;RK611 PRIORITY
1571  002004  000000                    SRTFLG: .WORD   0               ;START FLAG
1572                                                                    ;   0 = 200
1573                                                                    ;   1 = 214
1574                                                                    ;  -1 = 204
1575  002006  000000                    ERRCNT: .WORD   0               ;ERROR COUNT FOR SWITCH 12 ABORT
1576  002010  000000                    CONFIG: .WORD   0               ;DATA WRITTEN IN REGISTER
1577  002012  000000                    CONFG1: .WORD   0               ;DATA GENERATION WORD
1578  002014  000000                    PREREG: .WORD   0               ;PREVIOUS REGISTER CONTENTS
1579  002016  000000                    SAVFLG: .WORD   0               ;TRAP CATCHER SAVED
1580  002020  000000                    PRIOR:  .WORD   0               ;PROCESSOR PRIORITY
1581  002022  000000                    SILCNT: .WORD   0               ;NUMBER OF WORDS READ OR WRITTEN IN SILO
1582  002024  000000                    TRAPPC: .WORD   0               ;TRAP PC FOR UNEXPECTED MAIN MEMORY CHECK
1583  002026  000000                    SAVSWR: .WORD   0               ;RESTORE SWITCH REGISTER
```

```
1584                                        .SBTTL  PROGRAM SETUP
1585
1586  002030  012737  000001  002004  PARM:   MOV     #1,SRTFLG       ;LOAD START FLAG FOR PARMETER START
1587  002036  000406                          BR      START1
1588
1589  002040  012737  177777  002004  RESTRT: MOV     #-1,SRTFLG      ;LOAD START FLAG FOR RESTART
1590  002046  000402                          BR      START1
1591
1592  002050  005037  002004          START:  CLR     SRTFLG          ;CLEAR START FLAG
1593  002054  000005                  START1: RESET                   ;RESET THE WHOLE SYSTEM
1594  002056  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
1595  002062  012746  000340                  MOV     #PR7,-(SP)      ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
1596  002066  012746  002074                  MOV     #1$,-(SP)       ;LOAD START OF PROGRAM
1597  002072  000002                          RTI                     ;LOAD PSW
1598
1599  002074                          1$:
1600                                        .SBTTL  INITIALIZE THE COMMON TAGS
1601                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1602  002074  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1603  002100  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1604  002102  022706  001140                  CMP     #SWR,R6 ;;DONE?
1605  002106  001374                          BNE     .-6             ;;LOOP BACK IF NO
1606  002110  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1607                                        ;;INITIALIZE A FEW VECTORS
1608  002114  012737  055416  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1609  002122  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1610  002130  012737  056416  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1611  002136  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1612  002144  012737  061322  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1613  002152  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1614  002160  012737  061172  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1615  002166  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1616  002174  013737  055110  055102          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1617  002202  005037  001200                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1618  002206  005037  001202                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1619  002212  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1620  002220  012737  002220  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1621  002226  012737  002226  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1622                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1623                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1624  002234  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1625  002240  012737  002274  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1626  002246  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1627  002254  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1628  002262  022777  177777  176650          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1629  002270  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1630                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1631  002272  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1632  002274  012716  002302          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1633  002300  000002                          RTI
1634  002302  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1635  002310  012737  000174  001142          MOV     #DISPREG,DISPLAY
1636  002316  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1637
1638  002322  005037  001222                  CLR     $PASS           ;;CLEAR PASS COUNT
1639  002326  132737  000200  001235          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
```

```
1640  002334  001403                        BEQ     67$                ;;YES,USE NON-APT SWITCH
1641  002336  012737  001236  001140        MOV     #$SWREG,SWR        ;;NO,USE APT SWITCH REGISTER
1642  002344                        67$:
1643  002344  005037  002006              CLR     ERRCNT             ;CLEAR ERROR COUNT
1644                                .SBTTL  TYPE PROGRAM NAME
1645                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1646  002350  005227  177777              INC     #-1                ;;FIRST TIME?
1647  002354  001055                        BNE     68$                ;;BRANCH IF NO
1648  002356  022737  055244  000042        CMP     #$ENDAD,@#42       ;;ACT-11?
1649  002364  001451                        BEQ     68$                ;;BRANCH IF YES
1650  002366  104401  002434              TYPE    ,69$               ;;TYPE ASCIZ STRING
1651                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1652  002372  005737  000042              TST     @#42               ;;ARE WE RUNNING UNDER XXDP/ACT?
1653  002376  001012                        BNE     70$                ;;BRANCH IF YES
1654  002400  123727  001234  000001        CMPB    $ENV,#1            ;;ARE WE RUNNING UNDER APT?
1655  002406  001406                        BEQ     70$                ;;BRANCH IF YES
1656  002410  023727  001140  000176        CMP     SWR,#SWREG         ;;SOFTWARE SWITCH REG SELECTED?
1657  002416  001005                        BNE     71$                ;;BRANCH IF NO
1658  002420  104406                        GTSWR                      ;;GET SOFT-SWR SETTINGS
1659  002422  000403                        BR      71$
1660  002424  112737  000001  001134  70$:  MOVB    #1,$AUTOB          ;;SET AUTO-MODE INDICATOR
1661  002432                        71$:
1662  002432  000426                        BR      68$                ;;GET OVER THE ASCIZ
1663                                ;;69$:  .ASCIZ  <CRLF>/RK611 DISKLESS DIAGNOSTIC: PART 1 CZR6AC0/<CRLF>
1664  002510                        68$:
1665  002510  022737  000001  002004        CMP     #1,SRTFLG          ;CHECK IF PARAMETER START
1666  002516  001122                        BNE     15$                ;NO, CONTINUE SETUP
1667  002520  104401  062052        5$:     TYPE    ,OPR001            ;TYPE "RK611 BUS ADDRESS (   ) ="
1668  002524  013746  001270              MOV     $BASE,-(SP)        ;;SAVE $BASE FOR TYPEOUT
1669  002530  104402                        TYPOC                      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1670  002532  104401  062101              TYPE    ,OPR002
1671  002536  104412                        RDOCT                      ;GET VALUE
1672  002540  012637  001160              MOV     (SP)+,$TMP0
1673  002544  001407                        BEQ     7$                 ;CHECK IF <CR>
1674  002546  022737  160000  001160        CMP     #160000,$TMP0      ;CHECK IF IN I/O PAGE
1675  002554  101361                        BHI     5$
1676  002556  013737  001160  001270        MOV     $TMP0,$BASE        ;LOAD NEW BUS ADDRESS
1677  002564  104401  062107        7$:     TYPE    ,OPR003            ;TYPE "RK611 VECTOR ADDRESS (   ) ="
1678  002570  013746  001264              MOV     $VECT1,-(SP)       ;TYPE OUT CURRENT VECTOR ADDRESS
1679  002574  042716  160000              BIC     #160000,(SP)
1680  002600  104402                        TYPOC
1681  002602  104401  062101              TYPE    ,OPR002
1682  002606  104412                        RDOCT                      ;GET VALUE
1683  002610  012637  001160              MOV     (SP)+,$TMP0
1684  002614  001412                        BEQ     10$                ;CHECK IF <CR>
1685  002616  022737  001000  001160        CMP     #1000,$TMP0        ;CHECK IF LEGAL
1686  002624  101757                        BLOS    7$
1687  002626  042737  017777  001264        BIC     #17777,$VECT1      ;LOAD NEW VECTOR ADDRESS
1688  002634  053737  001160  001264        BIS     $TMP0,$VECT1
1689  002642  104401  062137        10$:    TYPE    ,OPR004            ;TYPE "RK611 PRIORITY (   ) ="
1690  002646  005046                        CLR     -(SP)              ;MAKE ROOM ON THE STACK
1691  002650  113716  001265              MOVB    $VECT1+1,(SP)
1692  002654  006216                        ASR     (SP)               ;SHIFT 5 BITS RIGHT
1693  002656  006216                        ASR     (SP)
1694  002660  006216                        ASR     (SP)
1695  002662  006216                        ASR     (SP)
```

```
 1696  002664  006216                        ASR     (SP)
 1697  002666  104402                        TYPOC
 1698  002670  104401  062101                TYPE    ,OPR002
 1699  002674  104412                        RDOCT                   ;GET VALUE
 1700  002676  012637  001160                MOV     (SP)+,$TMP0
 1701  002702  001430                        BEQ     15$             ;CHECK FOR DEFAULT
 1702  002704  022737  000007  001160        CMP     #7,$TMP0        ;CHECK IF LEGAL
 1703  002712  103753                        BLO     10$
 1704  002714  022737  000004  001160        CMP     #4,$TMP0
 1705  002722  101347                        BHI     10$
 1706  002724  006337  001160                ASL     $TMP0           ;SHIFT 5 BITS LEFT
 1707  002730  006337  001160                ASL     $TMP0
 1708  002734  006337  001160                ASL     $TMP0
 1709  002740  006337  001160                ASL     $TMP0
 1710  002744  006337  001160                ASL     $TMP0
 1711  002750  042737  160000  001264        BIC     #160000,$VECT1  ;STORE NEW PRIORITY
 1712  002756  153737  001160  001265        BISB    $TMP0,$VECT1+1
 1713  002764  013737  001264  002000  15$:  MOV     $VECT1,RKVEC    ;STORE RK611 VECTOR
 1714  002772  042737  160000  002000        BIC     #160000,RKVEC   ;CLEAR PRIORITY BITS
 1715  003000  113737  001265  002002        MOVB    $VECT1+1,RKPRI  ;STORE PRIORITY
 1716  003006  005737  002016                TST     SAVFLG          ;CHECK IF TRAP CATCHER IS TO BE RESTORED
 1717  003012  001412                        BEQ     NEWPAS          ;NO, GO TO FIRST TEST
 1718  003014  012701  067754                MOV     #SAVVEC,R1      ;RESTORE TRAP CATCHER
 1719  003020  005000                        CLR     R0
 1720  003022  012703  000400                MOV     #400,R3         ;STORE COUNT
 1721  003026  012120                  16$:  MOV     (R1)+,(R0)+
 1722  003030  005303                        DEC     R3
 1723  003032  001375                        BNE     16$
 1724  003034  005037  002016                CLR     SAVFLG          ;INDICATE THAT TRAP CATCHER HAS BEEN RESTORED
 1725
 1726  003040  004737  055264  NEWPAS: JSR     PC,CHKPAR       ;CHECK FOR MEMORY CHECK ENABLE
 1727  003044  013702  001270                MOV     $BASE,R2        ;LOAD RK611 BASE ADDR
 1728  003050  012746  000340                MOV     #PR7,-(SP)      ;LOCK OUT INTERRUPTS
 1729  003054  012746  003062                MOV     #TST1,-(SP)
 1730  003060  000002                        RTI
```

```
                                        ;;*****************************************************************
                                        ;*TEST 1         ADDRESS ALL RK611 REGISTERS
                                        ;*
                                        ;*       THIS TEST WILL ACCESS ALL RK611 REGISTERS AND CHECK TO
                                        ;*       MAKE SURE THAT NON-EXISTENT MEMORY DOES NOT OCCUR.
                                        ;*       A NON-EXISTENT MEMORY INDICATES EITHER THAT THE RK611
                                        ;*       REGISTER BASE ADDRESS IS INCORRECT OR THAT THE
                                        ;*       RK611 DOES NOT RESPOND TO UNIBUS DIALOGUE.
                                        ;*
                                        ;*****************************************************************
003062  000004                    TST1:   SCOPE
003064  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
003072  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE
003076  012737  003130  000004            MOV     #10$,ERRVEC     ;LOAD VECTOR FOR NEM
003104  012737  000340  000006            MOV     #PR7,ERRVEC+2
003112  012703  000020                    MOV     #16.,R3         ;LOAD NUMBER OF REGISTER
003116  005712                    5$:     TST     (R2)            ;ADDRESS RK611 REGISTER
003120  005722                            TST     (R2)+           ;INCREMENT TO NEXT REGISTER
003122  005303                            DEC     R3              ;CHECK IF ALL REGISTERS ADDRESS
003124  001374                            BNE     5$              ;NO,CONTINUE
003126  000405                            BR      15$             ;RESTORE TRAP CATCHER

003130  062706  000004            10$:    ADD     #4,SP           ;ADJUST STACK
003134  010237  001122                    MOV     R2,$BDADR       ;LOAD ADDRESS PRINT OUT
003140  104001                            ERROR   1
003142  012737  000006  000004    15$:    MOV     #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
003150  005037  000006                    CLR     ERRVEC+2


                                        .SBTTL  **RESET, CONTROLLER CLEAR, AND TRI-STATE TESTS

                                        ;;*****************************************************************
                                        ;*TEST 2         RESET RK611 AND VERIFY REGISTERS
                                        ;*
                                        ;*       RESET THE RK611 CONTROLLER AND READ ALL REGISTER OF THE
                                        ;*       RK611 REGISTERS EXCEPT THE DATA BUFFER AND VERIFY THAT
                                        ;*       THEY ARE CORRECT.  REEXAMINE COMMAND AND STATUS REGISTER 1
                                        ;*       TO MAKE SURE CONTROLLER ERROR DID NOT SET.  REEXAMINE
                                        ;*       COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE
                                        ;*       DID NOT SET.
                                        ;*
                                        ;*       THE SUCCESSFUL EXECUTION OF THIS TEST VERIFIES THAT NO
                                        ;*       BIT OF THE TRI-STATE BUS IS STUCK TO ONE.
                                        ;*
                                        ;*****************************************************************
003154  000004                    TST2:   SCOPE
003156  012737  000012  001200            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
003164  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE REGISTER
003170  012737  063326  001310            MOV     #EM2,EM2N       ;LOAD ERROR MESSAGE FOR PRINT OUT
003176  000005                            RESET                   ;CLEAR RK611 WITH  UNIBUS INIT
003200  004737  055264                    JSR     PC,CHKPAR       ;CHECK FOR MEMORY CHECK ENABLE
003204  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;SAVE COMMAND AND STATUS REG.7
003212  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
003220  001407                            BEQ     1$              ;YES,CHECK OTHER REGISTERS
003222  012737  000200  001124            MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
003230  012737  066574  001312            MOV     #EM1000,EM2N+2  ;LOAD ERROR MESSAGE
```

```
1787  003236  104002                        ERROR   2
1788  003240  016237  000004  001126  1$:   MOV     RKBA(R2),$BDDAT ;SAVE BUS ADD REG
1789  003246  001406                        BEQ     2$              ;CHECK IF ZERO
1790  003250  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1791  003254  012737  066637  001312        MOV     #EM1002,EM2N+2  ;LOAD ERROR MESSAGE
1792  003262  104002                        ERROR   2
1793  003264  016237  000006  001126  2$:   MOV     RKDA(R2),$BDDAT ;SAVE DISK ADDRESS ERROR
1794  003272  001406                        BEQ     3$              ;CHECK IF ZERO
1795  003274  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1796  003300  012737  066661  001312        MOV     #EM1003,EM2N+2  ;LOAD ERROR MESSAGE
1797  003306  104002                        ERROR   2
1798  003310  016237  000016  001126  3$:   MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION/OFFSET REG.
1799  003316  001406                        BEQ     4$              ;CHECK IF ZERO
1800  003320  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1801  003324  012737  067001  001312        MOV     #EM1007,EM2N+2  ;LOAD ERROR MESSAGE
1802  003332  104002                        ERROR   2
1803  003334  016237  000010  001126  4$:   MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2
1804  003342  022737  000100  001126        CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
1805  003350  001407                        BEQ     5$              ;YES,CONTINUE
1806  003352  012737  000100  001124        MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
1807  003360  012737  066704  001312        MOV     #EM1004,EM2N+2  ;LOAD ERROR MESSAGE
1808  003366  104002                        ERROR   2
1809  003370  016237  000012  001126  5$:   MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
1810  003376  001406                        BEQ     6$              ;CHECK IF ZERO
1811  003400  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1812  003404  012737  066722  001312        MOV     #EM1005,EM2N+2  ;LOAD ERROR MESSAGE
1813  003412  104002                        ERROR   2
1814  003414  016237  000014  001126  6$:   MOV     RKER(2),$BDDAT  ;STORE ERROR STATUS REG
1815  003422  001406                        BEQ     7$              ;CHECK IF ZERO
1816  003424  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1817  003430  012737  066755  001312        MOV     #EM1006,EM2N+2  ;LOAD ERROR MESSAGE
1818  003436  104002                        ERROR   2
1819  003440  016237  000020  001126  7$:   MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
1820  003446  001406                        BEQ     9$              ;CHECK IF EQUAL ZERO
1821  003450  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1822  003454  012737  067272  001312        MOV     #EM1016,EM2N+2  ;LOAD ERROR MESSAGE
1823  003462  104002                        ERROR   2
1824  003464  016237  000026  001126  9$:   MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
1825  003472  012737  002000  001124        MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
1826  003500  032737  020000  001126        BIT     #ECCW,$BDDAT
1827  003506  001403                        BEQ     10$
1828  003510  052737  020000  001124        BIS     #ECCW,$GDDAT
1829  003516  023737  001124  001126  10$:  CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
1830  003524  001407                        BEQ     11$             ;YES,CONTINUE TEST
1831  003526  012737  022000  001124        MOV     #ECCW!MEWD,$GDDAT ;LOAD EXPECTED CONTENTS
1832  003534  012737  067072  001312        MOV     #EM1009,EM2N+2  ;LOAD ERROR MESSAGE
1833  003542  104002                        ERROR   2
1834  003544  016237  000032  001126  11$:  MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
1835  003552  001406                        BEQ     12$             ;CHECK IF ZERO
1836  003554  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED RESULTS
1837  003560  012737  067142  001312        MOV     #EM1013,EM2N+2  ;LOAD ERROR MESSAGE
1838  003566  104002                        ERROR   2
1839  003570  016237  000030  001126  12$:  MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
1840  003576  022737  004066  001126        CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
1841  003604  001407                        BEQ     13$             ;YES,CONTINUE
1842  003606  012737  004066  001124        MOV     #4066,$GDDAT    ;LOAD EXPECTED RESULTS
```

```
 1843  003614  012737  067120  001312          MOV     #EM1012,EM2N+2  ;LOAD ERROR MESSAGE
 1844  003622  104002                          ERROR   2
 1845  003624  016237  000000  001126  13$:    MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG 1
 1846  003632  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
 1847  003640  001407                          BEQ     14$             ;YES, CONTINUE
 1848  003642  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
 1849  003650  012737  067164  001312          MOV     #EM1014,EM2N+2  ;LOAD ERROR MESSAGE
 1850  003656  104002                          ERROR   2
 1851  003660  016237  000010  001126  14$:    MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2.
 1852  003666  022737  010100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
 1853  003674  001407                          BEQ     TST3            ;;GO ON TO NEXT TEXT
 1854  003676  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
 1855  003704  012737  067227  001312          MOV     #EM1015,EM2N+2  ;LOAD ERROR MESSAGE
 1856  003712  104002                          ERROR   2
 1857
 1858                                  ;;*****************************************************************
 1859                                  ;*TEST 3         CONTROLLER CLEAR AND VERIFY REGISTERS
 1860                                  ;*
 1861                                  ;*       INITIALIZE THE RK611 CONTROLLER WITH A CONTROLLER
 1862                                  ;*       CLEAR AND READ ALL REGISTER OF THE RK611 REGISTERS
 1863                                  ;*       EXCEPT THE DATA BUFFER AND VERIFY THAT THEY ARE
 1864                                  ;*       CORRECT.  REEXAMINE COMMAND AND STATUS REGISTER 1
 1865                                  ;*       TO MAKE SURE CONTROLLER ERROR DID NOT SET.  REEXAMINE
 1866                                  ;*       COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE
 1867                                  ;*       DID NOT SET.
 1868                                  ;*
 1869                                  ;;*****************************************************************
 1870  003714  000004                  TST3:   SCOPE
 1871  003716  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
 1872  003724  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE REGISTER
 1873  003730  012737  063403  001310          MOV     #EM3,EM2N       ;LOAD ERROR MESSAGE FOR PRINT OUT
 1874  003736  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
 1875  003744  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;SAVE COMMAND AND STATUS REG.7
 1876  003752  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
 1877  003760  001407                          BEQ     1$              ;YES,CHECK OTHER REGISTERS
 1878  003762  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
 1879  003770  012737  066574  001312          MOV     #EM1000,EM2N+2  ;LOAD ERROR MESSAGE
 1880  003776  104002                          ERROR   2
 1881  004000  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;SAVE BUS ADD REG
 1882  004006  001406                          BEQ     2$              ;CHECK IF ZERO
 1883  004010  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 1884  004014  012737  066637  001312          MOV     #EM1002,EM2N+2  ;LOAD ERROR MESSAGE
 1885  004022  104002                          ERROR   2
 1886  004024  016237  000006  001126  2$:     MOV     RKDA(R2),$BDDAT ;SAVE DISK ADDRESS ERROR
 1887  004032  001406                          BEQ     3$              ;CHECK IF ZERO
 1888  004034  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 1889  004040  012737  066661  001312          MOV     #EM1003,EM2N+2  ;LOAD ERROR MESSAGE
 1890  004046  104002                          ERROR   2
 1891  004050  016237  000016  001126  3$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION/OFFSET REG.
 1892  004056  001406                          BEQ     4$              ;CHECK IF ZERO
 1893  004060  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 1894  004064  012737  067001  001312          MOV     #EM1007,EM2N+2  ;LOAD ERROR MESSAGE
 1895  004072  104002                          ERROR   2
 1896  004074  016237  000010  001126  4$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2
 1897  004102  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
 1898  004110  001407                          BEQ     5$              ;YES,CONTINUE
```

```
1899  004112  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
1900  004120  012737  066704  001312          MOV     #EM1004,EM2N+2  ;LOAD ERROR MESSAGE
1901  004126  104002                          ERROR   2
1902  004130  016237  000012  001126  5$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
1903  004136  001406                          BEQ     6$              ;CHECK IF ZERO
1904  004140  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1905  004144  012737  066722  001312          MOV     #EM1005,EM2N+2  ;LOAD ERROR MESSAGE
1906  004152  104002                          ERROR   2
1907  004154  016237  000014  001126  6$:     MOV     RKER(2),$BDDAT  ;STORE ERROR STATUS REG
1908  004162  001406                          BEQ     7$              ;CHECK IF ZERO
1909  004164  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1910  004170  012737  066755  001312          MOV     #EM1006,EM2N+2  ;LOAD ERROR MESSAGE
1911  004176  104002                          ERROR   2
1912  004200  016237  000020  001126  7$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
1913  004206  001406                          BEQ     9$              ;CHECK IF EQUAL ZERO
1914  004210  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1915  004214  012737  067272  001312          MOV     #EM1016,EM2N+2  ;LOAD ERROR MESSAGE
1916  004222  104002                          ERROR   2
1917  004224  016237  000026  001126  9$:     MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
1918  004232  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
1919  004240  032737  020000  001126          BIT     #ECCW,$BDDAT
1920  004246  001403                          BEQ     10$
1921  004250  052737  020000  001124          BIS     #ECCW,$GDDAT
1922  004256  023737  001124  001126  10$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
1923  004264  001407                          BEQ     11$             ;YES,CONTINUE TEST
1924  004266  012737  022000  001124          MOV     #ECCW!MEWD,$GDDAT ;LOAD EXPECTED CONTENTS
1925  004274  012737  067072  001312          MOV     #EM1009,EM2N+2  ;LOAD ERROR MESSAGE
1926  004302  104002                          ERROR   2
1927  004304  016237  000032  001126  11$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
1928  004312  001406                          BEQ     12$             ;CHECK IF ZERO
1929  004314  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
1930  004320  012737  067142  001312          MOV     #EM1013,EM2N+2  ;LOAD ERROR MESSAGE
1931  004326  104002                          ERROR   2
1932  004330  016237  000030  001126  12$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
1933  004336  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
1934  004344  001407                          BEQ     13$             ;YES,CONTINUE
1935  004346  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED RESULTS
1936  004354  012737  067120  001312          MOV     #EM1012,EM2N+2  ;LOAD ERROR MESSAGE
1937  004362  104002                          ERROR   2
1938  004364  016237  000000  001126  13$:    MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG 1
1939  004372  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
1940  004400  001407                          BEQ     14$             ;YES, CONTINUE
1941  004402  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
1942  004410  012737  067164  001312          MOV     #EM1014,EM2N+2  ;LOAD ERROR MESSAGE
1943  004416  104002                          ERROR   2
1944  004420  016237  000010  001126  14$:    MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2.
1945  00442E  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
1946  004434  001407                          BEQ     TST4            ;;GO ON TO NEXT TEXT
1947  004436  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
1948  004444  012737  067227  001312          MOV     #EM1015,EM2N+2  ;LOAD ERROR MESSAGE
1949  004452  104002                          ERROR   2
1950
1951                                  ;;*****************************************************************
1952                                  ;*TEST 4         WRITE BUS ADDRESS WITH 177777 (PART 1)
1953                                  ;*
1954                                  ;*      THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777
```

```
1955                                    ;*      AND CHECK IF EQUAL TO 177776 AND THAT NO REGISTER INTERACTION
1956                                    ;*      TAKES PLACE. A RESET IS DONE AT THE END OF THE TEST TO
1957                                    ;*      MAKE SURE THE BUS ADDRESS CLEARS AND ALL RK611 REGISTERS
1958                                    ;*      ARE IN THEIR INITIALIZED STATE.
1959                                    ;*
1960                                    ;**************************************************************
1961  004454  000004                   T$T4:   SCOPE
1962  004456  012737  000012  001200           MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1963  004464  013702  001270                   MOV     $BASE,R2        ;LOAD RK611 BASE
1964  004470  012737  177777  002010           MOV     #177777,CONFIG  ;LOAD VALUE WRITTEN FOR PRINT OUT
1965  004476  012737  063465  001320           MOV     #EM4,EM3N       ;LOAD ERROR MESSAGE
1966  004504  012762  100000  000000           MOV     #CCLR,RKCS1(R2) ;INITIALIZE RK611 CONTROLLER
1967  004512  012762  177777  000004           MOV     #177777,RKBA(R2);WRITE BUS ADDRESS WITH 177777
1968  004520  016237  000004  001126           MOV     RKBA(R2),$BDDAT ;STORE BUS ADDRESS REG
1969  004526  022737  177776  001126           CMP     #177776,$BDDAT  ;CHECK IF BUS ADDRESS CORRECT
1970  004534  001407                           BEQ     1$              ;YES, CHECK IF ANY REGISTER MODIFIED
1971  004536  012737  177776  001124           MOV     #177776,$GDDAT  ;LOAD EXPECTED RESULTS
1972  004544  012737  066637  001322           MOV     #EM1002,EM3N+2  ;LOAD ERROR MESSAGE
1973  004552  104003                           ERROR   3
1974  004554  016237  000000  001126  1$:      MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
1975  004562  022737  000200  001126           CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
1976  004570  001407                           BEQ     2$              ;YES, CHECK OTHER REGISTERS
1977  004572  012737  000200  001124           MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
1978  004600  012737  067314  001322           MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
1979  004606  104003                           ERROR   3
1980  004610  016237  000006  001126  2$:      MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG.
1981  004616  001406                           BEQ     3$              ;CHECK IF ZERO
1982  004620  005037  001124                   CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1983  004624  012737  067412  001322           MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
1984  004632  104003                           ERROR   3
1985  004634  016237  000016  001126  3$:      MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
1986  004642  001406                           BEQ     4$              ;CHECK IF ZERO
1987  004644  005037  001124                   CLR     $GDDAT          ;LOAD EXPECTED CONTENTSQ
1988  004650  012737  067556  001322           MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
1989  004656  104003                           ERROR   3
1990  004660  016237  000010  001126  4$:      MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2
1991  004666  022737  000100  001126           CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
1992  004674  001407                           BEQ     5$              ;YES,CONTINUE
1993  004676  012737  000100  001124           MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
1994  004704  012737  067450  001322           MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
1995  004712  104003                           ERROR   3
1996  004714  016237  000012  001126  5$:      MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
1997  004722  001406                           BEQ     6$              ;CHECK IF ZERO
1998  004724  005037  001124                   CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
1999  004730  012737  067471  001322           MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
2000  004736  104003                           ERROR   3
2001  004740  016237  000014  001126  6$:      MOV     RKER(R2),$BDDAT ;STORE ERROR REG.
2002  004746  001406                           BEQ     7$              ;CHECK IF ZERO
2003  004750  005037  001124                   CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2004  004754  012737  067527  001322           MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
2005  004762  104003                           ERROR   3
2006  004764  016237  000020  001126  7$:      MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD. REG..
2007  004772  001406                           BEQ     9$              ;CHECK IF EQUAL ZERO
2008  004774  005037  001124                   CLR     $GDDAT          ;LOAD EXPECTED RESULTS
2009  005000  012737  067624  001322           MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
2010  005006  104003                           ERROR   3
```

```
2011  005010  016237  000026  001126  9$:     MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG. 1
2012  005016  012737  002000  001124          MOV     #MEWD,$GDDAT     ;LOAD EXPECTED MR1
2013  005024  032737  020000  001126          BIT     #ECCW,$BDDAT
2014  005032  001403                          BEQ     10$
2015  005034  052737  020000  001124          BIS     #ECCW,$GDDAT
2016  005042  023737  001124  001126  10$:    CMP     $GDDAT,$BDDAT    ;CHECK IF MR1 CORRECT
2017  005050  001404                          BEQ     11$              ;YES,CONTINUE TEST
2018  005052  012737  067651  001322          MOV     #EM1026,EM3N+2   ;LOAD ERROR MESSAGE
2019  005060  104003                          ERROR   3
2020  005062  016237  000032  001126  11$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2021  005070  001406                          BEQ     12$              ;CHECK IF ZERO
2022  005072  005037  001124                  CLR     $GDDAT           ;LOAD EXPECTED RESULTS
2023  005076  012737  067727  001322          MOV     #EM1030,EM3N+2   ;LOAD ERROR MESSAGE
2024  005104  104003                          ERROR   3
2025  005106  016237  000030  001126  12$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REGC
2026  005114  022737  004066  001126          CMP     #4066,$BDDAT     ;CHECK IF ECC POSITION CRRECT
2027  005122  001407                          BEQ     13$              ;YES, CLEAR RK611
2028  005124  012737  004066  001124          MOV     #4066,$GDDAT     ;LOAD EXPECTED RESULTS
2029  005132  012737  067702  001322          MOV     #EM1029,EM3N+2   ;LOAD ERROR MESSAGE
2030  005140  104003                          ERROR   3
2031  005142  012737  177776  002014  13$:    MOV     #177776,PREREG   ;LOAD PREVIOUS CONTENTS
2032  005150  012737  063521  001330          MOV     #EM5,EM4N        ;LOAD ERROR MESSAGE
2033  005156  000005                          RESET                    ;CLEAR RK611 WITH UNIBUS INIT
2034  005160  004737  055264                  JSR     PC,CHKPAR        ;CHECK FOR MEMORY CHECK ENABLE
2035  005164  016237  000004  001126          MOV     RKBA(R2),$BDDAT  ;STORE BUS ADDRESS REG.
2036  005172  001406                          BEQ     TST5             ;;CHECK IF CLEARED
2037  005174  005037  001124                  CLR     $GDDAT           ;LOAD EXPECTED RESULTS
2038  005200  012737  066637  001332          MOV     #EM1002,EM4N+2   ;LOAD ERROR MESSAGE
2039  005206  104004                          ERROR   4
2040
2041                                    ;;**********************************************************************
2042                                    ;*TEST 5        WRITE BUS ADDRESS WITH 177777 (PART 2)
2043                                    ;*
2044                                    ;*      THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777.
2045                                    ;*      A CONTROLLER CLEAR IS DONE. MAKE SURE THE BUS ADDRESS CLEARS.
2046                                    ;*
2047                                    ;;**********************************************************************
2048  005210  000004                    TST5:   SCOPE
2049  005212  012737  000144  001200            MOV     #100.,$TIMES     ;;DO 100. ITERATIONS
2050  005220  013702  001270                    MOV     $BASE,R2         ;LOAD RK611 BASE
2051  005224  012762  100000  000000            MOV     #CCLR,RKCS1(R2)  ;CLEAR KK611 WITH CONTROLLER CLEAR
2052  005232  012762  177777  000004            MOV     #177777,RKBA(R2) ;LOAD BUS ADDRESS WITH ALL ONES
2053  005240  012762  100000  000000            MOV     #CCLR,RKCS1(R2)  ;CLEAR RK611 WITH CONTROLLER CLEAR
2054  005246  016237  000004  001126            MOV     RKBA(R2),$BDDAT  ;STORE BUS ADDRESS
2055  005254  001414                           BEQ     TST6             ;;CHECK IF ZERO
2056  005256  012737  177776  002014            MOV     #177776,PREREG   ;LOAD PREVIOUS CONTENTS
2057  005264  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2058  005270  012737  063600  001330            MOV     #EM6,EM4N        ;LOAD ERROR MESSAGE
2059  005276  012737  066637  001332            MOV     #EM1002,EM4N+2
2060  005304  104004                           ERROR   4
2061
2062                                    ;;**********************************************************************
2063                                    ;*TEST 6        WRITE WORD COUNT REG. WITH 177777
2064                                    ;*
2065                                    ;*      THIS TEST WILL WRITE THE WORD COUNT REGISTER
2066                                    ;*      TO 0 AND 177777 AND CHECK IF EQUAL TO 0 AND 177777
```

```
                                        T6      WRITE WORD COUNT REG. WITH 177777

                                        ;*      RESPECTIVELY AND THAT NO REGISTER INTERACTION TAKES PLACE.
                                        ;*
                                        ;*      ISSUE A CONTROLLER CLEAR AND MADE SURE THAT THE WORD
                                        ;*    . COUNT REGISTER DOES NOT CHANGE.
                                        ;*
                                        ;**************************************************************
005306  000004                          T$T6:   SCOPE
005310  012737  000764  001200                  MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
005316  013702  001270                          MOV     $BASE,R2        ;LOAD RK611 BASE
005322  005037  002010                          CLR     CONFIG          ;CLEAR CONFIGURATION WORD
005326  012737  063663  001320                  MOV     #EM7,EM3N       ;LOAD ERROR MESSAGE
005334  012737  005342  001110                  MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                        ;  SUBTEST LOOP

005342                                  1$:
005342  012762  100000  000000                  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WTIH CONTROLLER CLEAR
005350  013762  002010  000002                  MOV     CONFIG,RKWC(R2) ;WRITE WORD COUNT REG.
005356  016237  000002  001126                  MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
005364  023737  002010  001126                  CMP     CONFIG,$BDDAT   ;CHECK IF WORD COUNT CORRECT
005372  001407                                  BEQ     2$              ;YES, CHECK IF ANY REGISTER MODIFIED
005374  013737  002010  001124                  MOV     CONFIG,$GDDAT   ;LOAD EXPECTED RESULTS
005402  012737  066612  001322                  MOV     #EM1001,EM3N+2  ;LOAD ERROR MESSAGE
005410  104003                                  ERROR   3
005412  016237  000000  001126          2$:     MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
005420  022737  000200  001126                  CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
005426  001407                                  BEQ     3$              ;YES, CONTINUE TEST
005430  012737  000200  001124                  MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
005436  012737  067314  001322                  MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
005444  104003                                  ERROR   3
005446  016237  000004  001126          3$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
005454  001406                                  BEQ     4$              ;CHECK IF ZERO
005456  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
005462  012737  067365  001322                  MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
005470  104003                                  ERROR   3
005472  016237  000006  001126          4$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG.
005500  001406                                  BEQ     5$              ;CHECK IF ZERO
005502  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTD CONTENTS
005506  012737  067412  001322                  MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
005514  104003                                  ERROR   3
005516  016237  000016  001126          5$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
005524  001406                                  BEQ     6$              ;CHECK IF ZERO
005526  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
005532  012737  067556  001322                  MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
005540  104003                                  ERROR   3
005542  016237  000010  001126          6$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2
005550  022737  000100  001126                  CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
005556  001407                                  BEQ     7$              ;YES,CONTINUE
005560  012737  000100  001124                  MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
005566  012737  067450  001322                  MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
005574  104003                                  ERROR   3
005576  016237  000012  001126          7$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
005604  001406                                  BEQ     8$              ;CHECK IF ZERO
005606  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
005612  012737  067471  001322                  MOV     #EM1022, EM3N+2 ;LOAD ERROR MESSAGE
005620  104003                                  ERROR   3
005622  016237  000014  001126          8$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
```

```
2123  005630  001406                        BEQ     9$                      ;CHECK IF ZERO
2124  005632  005037  001124                CLR     $GDDAT                  ;LOAD EXPECTED CONTENTS
2125  005636  012737  067527  001322        MOV     #EM1023,EM3N+2          ;LOAD ERROR MESSAGE
2126  005644  104003                        ERROR   3
2127  005646  016237  000020  001126  9$:   MOV     RKDCYL(R2),$BDDAT       ;STORE CYLINDER ADD. REG
2128  005654  001406                        BEQ     11$                     ;CHECK IF EQUAL ZERO
2129  005656  005037  001124                CLR     $GDDAT                  ;LOAD EXPECTED RESULTS
2130  005662  012737  067624  001322        MOV     #EM1025,EM3N+2          ;LOAD ERROR MESSAGE
2131  005670  104003                        ERROR   3
2132  005672  016237  000026  001126  11$:  MOV     RKMR1(R2),$BDDAT        ;STORE MAINTENANCE REG 1
2133  005700  012737  002000  001124        MOV     #MEWD,$GDDAT            ;LOAD EXPECTED MR1
2134  005706  032737  020000  001126        BIT     #ECCW,$BDDAT
2135  005714  001403                        BEQ     12$
2136  005716  052737  020000  001124        BIS     #ECCW,$GDDAT
2137  005724  023737  001124  001126  12$:  CMP     $GDDAT,$BDDAT           ;CHECK IF MR1 CORREGT
2138  005732  001404                        BEQ     13$                             ;YES, CONTINUE TEST
2139  005734  012737  067651  001322        MOV     #EM1026,EM3N+2          ;LOAD ERROR MESSAGE
2140  005742  104003                        ERROR   3
2141  005744  016237  000032  001126  13$:  MOV     RKECPT(R2),$BDDAT       ;STORE ECC PATTERN REG.
2142  005752  001406                        BEQ     14$                     ;CHECK IF ZERO
2143  005754  005037  001124                CLR     $GDDAT                  ;LOAD EXPECTED RESULTS
2144  005760  012737  067727  001322        MOV     #EM1030,EM3N+2          ;LOAD ERROR MESSAGE
2145  005766  104003                        ERROR   3
2146  005770  016237  000030  001126  14$:  MOV     RKECPS(R2),$BDDAT       ;STORE ECC POSITION REG.
2147  005776  022737  004066  001126        CMP     #4066,$BDDAT            ;CHECK IF ECC POSITION CORRECT
2148  006004  001407                        BEQ     15$                     ;YES,CLEAR RK611
2149  006006  012737  004066  001124        MOV     #4066,$GDDAT            ;LOAD EXPECTED RESULTS
2150  006014  012737  067702  001322        MOV     #EM1029,EM3N+2          ;LOAD ERROR MESSAGE
2151  006022  104003                        ERROR   3
2152  006024  012762  100000  000000  15$:  MOV     #CCLR,RKCS1(R2)         ;CLEAR RK611
2153  006032  016237  000002  001126        MOV     RKWC(R2),$BDDAT         ;STORE WORD COUNT REG.
2154  006040  023737  002010  001126        CMP     CONFIG,$BDDAT           ;CHECK IF WORD COUNT NOT CHANGED
2155                                                                        ;  BY COTROLLER CLEAR
2156  006046  001412                        BEQ     20$                     ;YES, CONTINUE
2157  006050  013737  002010  001124        MOV     CONFIG,$GDDAT           ;LOAD EXPECTED DATA
2158  006056  012737  063403  001310        MOV     #EM3,EM2N               ;LOAD ERROR MESSAGE
2159  006064  012737  067335  001312        MOV     #EM1018,EM2N+2
2160  006072  104002                        ERROR   2
2161  006074  104415                20$:    SCOP1                           ;TEST IF LOOP ON ERROR
2162  006076  005737  002010                TST     CONFIG                  ;CHECK IF FINISHED
2163  006102  001005                        BNE     TST7                    ;;YES, GO TO NEXT TEST
2164  006104  012737  177777  002010        MOV     #177777,CONFIG                  ;USE 177777 FOR SECOND PASS
2165  006112  000137  005342                JMP     1$
2166
2167                                        ;;*****************************************************************
2168                                        ;*TEST 7         WRITE DISK ADDRESS WITH 177777
2169                                        ;*
2170                                        ;*      THIS TEST WILL WRITE THE DISK ADDRESS REGISTER TO 177777
2171                                        ;*      AND CHECK IF EQUAL TO 003437 AND THAT NO REGISTER INTERACTION
2172                                        ;*      TAKES PLACE. A CONTROLLER CLEAR IS DONE AT THE END OF
2173                                        ;*      THE TEST TO MAKE SURE THE DISK ADDRESS CLEARS.
2174                                        ;*
2175                                        ;*****************************************************************
2176  006116  000004                TST7:   SCOPE
2177  006120  012737  000144  001200        MOV     #100.,$TIMES            ;;DO 100. ITERATIONS
2178  006126  013702  001270                MOV     $BASE,R2                ;LOAD RK611 BASE
```

```
2179  006132  012737  177777  002010          MOV     #177777,CONFIG  ;LOAD CONFIGURAION WORD
2180  006140  012737  063722  001320          MOV     #EM8,EM3N       ;LOAD ERROR MESSAGE
2181  006146  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
2182  006154  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG
2183  006160  012762  177777  000006          MOV     #177777,RKDA(R2) ;WRITE DISK ADDRESS WITH
2184                                                                  ;  177777
2185  006166  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG.
2186  006174  022737  003437  001126          CMP     #3437,$BDDAT    ;CHECK IF DISK ADDRESS REG CORRECT
2187  006202  001407                          BEQ     1$              ;YES, CHECK IF AY REGISTER MODIFIED
2188  006204  012737  003437  001124          MOV     #3437,$GDDAT    ;LOAD EXPECTED RESULTS
2189  006212  012737  066661  001322          MOV     #EM1003,EM3N+2  ;LOAD ERROR MESSAGE
2190  006220  104003                          ERROR   3
2191  006222  016237  000000  001126  1$:     MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
2192  006230  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
2193  006236  001407                          BEQ     2$              ;YES, CONTINUE
2194  006240  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
2195  006246  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
2196  006254  104003                          ERROR   3
2197  006256  016237  000004  001126  2$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
2198  006264  001406                          BEQ     3$              ;CHECK IF ZERO
2199  006266  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2200  006272  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
2201  006300  104003                          ERROR   3
2202  006302  016237  000002  001126  3$:     MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
2203  006310  001406                          BEQ     4$              ;CHECK IF ZERO
2204  006312  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2205  006316  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
2206  006324  104003                          ERROR   3
2207  006326  016237  000016  001126  4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2208  006334  001406                          BEQ     5$              ;CHECK IF ZERO
2209  006336  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2210  006342  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
2211  006350  104003                          ERROR   3
2212  006352  016237  000010  001126  5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
2213  006360  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
2214  006366  001407                          BEQ     6$              ;YES,CONTINUE
2215  006370  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
2216  006376  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
2217  006404  104003                          ERROR   3
2218  006406  016237  000012  001126  6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STAUS REG.
2219  006414  001406                          BEQ     7$              ;CHECK IF ZERO
2220  006416  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2221  006422  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
2222  006430  104003                          ERROR   3
2223  006432  016237  000014  001126  7$:     MOV     RKER(R2),$BDDAT ;STORE ERRPR REG.
2224  006440  001406                          BEQ     8$              ;CHECK IF ZERO
2225  006442  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2226  006446  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
2227  006454  104003                          ERROR   3
2228  006456  016237  000020  001126  8$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2229  006464  001406                          BEQ     10$             ;CHECK IF EQUAL ZERO
2230  006466  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
2231  006472  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
2232  006500  104003                          ERROR   3
2233  006502  016237  000026  001126  10$:    MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
2234  006510  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
```

```
2235  006516  032737  020000  001126          BIT     #ECCW,$BDDAT
2236  006524  001403                          BEQ     11$
2237  006526  052737  020000  001124          BIS     #ECCW,$GDDAT
2238  006534  023737  001124  001126  11$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
2239  006542  001404                          BEQ     12$             ;YES,CONTINUE TEST
2240  006544  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
2241  006552  104003                          ERROR   3
2242  006554  016237  000032  001126  12$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2243  006562  001406                          BEQ     13$             ;CHECK IF ZERO
2244  006564  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
2245  006570  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
2246  006576  104003                          ERROR   3
2247  006600  016237  000030  001126  13$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2248  006606  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
2249  006614  001407                          BEQ     14$             ;YES, ISSUE CONTROLLER CLEAR
2250  006616  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPE  ED RESULTS
2251  006624  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERR   MESSAGE
2252  006632  104003                          ERROR   3
2253  006634  012762  100000  000000  14$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK11
2254  006642  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG.
2255  006650  001414                          BEQ     TST10           ;;YES, GO ON TO NEXT TEST
2256  006652  005037  001124                  CLR     $GDDAT                  ;LOAD EXPECTED CONTENTS
2257  006656  012737  067403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
2258  006664  012737  066661  001332          MOV     #EM1003,EM4N+2
2259  006672  012737  003437  002014          MOV     #3437,PREREG    ;LOAD PREVIOUS VALUE
2260  006700  104004                          ERROR   4
2261
2262
2263                                  ;;*****************************************************************
2264                                  .SBTTL  **REGISTER INTERACTION TESTS
2265                                  ;*
2266                                  ;*      ALL REGISTER INTERACTION TESTS CONSISTS OF WRITING A
2267                                  ;*      REGISTER AND CHECKING ITS CONTENTS AGAINST EXPECTED
2268                                  ;*      CONTENTS. THEN ALL OTHER REGISTERS ARE READ EXCEPT
2269                                  ;*      THE DATA BUFFER TO CHECK WHETHER THEY HAVE CHANGED FROM
2270                                  ;*      THEIR INITIALIZED CONDITIONS.
2271                                  ;*
2272                                  ;*****************************************************************
2273
2274                                  ;*****************************************************************
2275                                  ;*TEST 10       REGISTER INTERACTION USING BUS ADDRESS (PART 1)
2276                                  ;*
2277                                  ;*      THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
2278                                  ;*       CLEAR TO HE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
2279                                  ;*      COUNT REGISTER TO 0.
2280                                  ;*
2281                                  ;*      THE TEST ITSELF WILL CONSIST OF WRITING THE
2282                                  ;*      BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
2283                                  ;*      TEST IF BUS ADDRESS IS CORRECT AND THAT NO
2284                                  ;*      REGISTER INTERACTION TAKES PLACE.
2285                                  ;*
2286                                  ;*      000000  000010  000200  004000  100000
2287                                  ;*      000001  000020  000400  010000
2288                                  ;*      000002  000040  001000  020000
2289                                  ;*      000004  000100  002000  040000
2290                                  ;*
```

```
                                ;*****************************************************************
                        TST10:  SCOPE
002200 000144 006702 000004             MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
002202 000021 006704 012701             MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
002204 002010 006712 012737 000001      MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
002206 001320 006716 012737 063465      MOV     #EM4,EM3N       ;LOAD ERROR MESSAGE
002210 000000 006724 012762 100000      MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
002212 001110 006732 012737 006746      MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
                        1$:
002214 000002 006746 005062             CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
002216 000004 006752 013762 002010      MOV     CONFIG,RKBA(R2) ;WRITE RKBA
002220 001126 006760 016237 000004      MOV     RKBA(R2),$BDDAT ;STORE RKBA
002222 001124 006766 013737 002010      MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
002224 001124 006774 042737 000001      BIC     #1,$GDDAT       ;INITIALIZE READ ONLY BITS
002226 001126 007002 023737 001124      CMP     $GDDAT,$BDDAT   ;CHECK IF RKBA CORRECT
002230 007010 001404                    BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
002232 001322 007012 012737 066637      MOV     #EM1002,EM3N+2  ;LOAD ERROR MESSAGE
002234 007020 104003                    ERROR   3
                        2$:
002236 001126 007022 016237 000000      MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
002240 007030 022737 000200 001126      CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
002242 007036 001407                    BEQ     3$              ;YES, CONTINUE
002244 001124 007040 012737 000100      MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
002246 001322 007046 012737 067314      MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
002250 007054 104003                    ERROR   3
                        3$:
002252 001126 007056 016237 000002      MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
002254 007064 001406                    BEQ     5$              ;CHECK IF ZERO
002256 001124 007066 005037             CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
002260 001322 007072 012737 067335      MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
002262 007100 104003                    ERROR   3
                        5$:
002264 001126 007102 016237 000006      MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
002266 007110 001406                    BEQ     6$              ;CHECK IF ZERO
002270 001124 007112 005037             CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
002272 001322 007116 012737 067412      MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
002274 007124 104003                    ERROR   3
                        6$:
002276 001126 007126 016237 000016      MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
002300 007134 001406                    BEQ     7$              ;CHECK IF ZERO
002302 001124 007136 005037             CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
002304 001322 007142 012737 067556      MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
002306 007150 104003                    ERROR   3
                        7$:
002310 001126 007152 016237 000010      MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
002312 007160 022737 000100 001126      CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
002314 007166 001407                    BEQ     8$              ;YES,CONTINUE
002316 001124 007170 012737 000100      MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
002320 001322 007176 012737 067450      MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
002322 007204 104003                    ERROR   3
                        8$:
002324 001126 007206 016237 000012      MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
002326 007214 001406                    BEQ     9$              ;CHECK IF ZERO
002330 001124 007216 005037             CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
```

```
2347  007222  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
2348  007230  104003                          ERROR   3
2349  007232  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
2350  007240  001406                          BEQ     10$             ;CHECK IF ZERO
2351  007242  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2352  007246  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
2353  007254  104003                          ERROR   3
2354  007256                          10$:
2355  007256  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2356  007264  001406                          BEQ     12$             ;CHECK IF ZERO
2357  007266  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2358  007272  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
2359  007300  104003                          ERROR   3
2360  007302                          12$:
2361  007302  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
2362  007310  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
2363  007316  032737  020000  001126          BIT     #ECCW,$BDDAT
2364  007324  001403                          BEQ     13$
2365  007326  052737  020000  001124          BIS     #ECCW,$GDDAT
2366  007334  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
2367  007342  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
2368  007344  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
2369  007352  104003                          ERROR   3
2370  007354                          14$:
2371  007354  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2372  007362  001406                          BEQ     15$             ;CHECK IF ZERO
2373  007364  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2374  007370  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
2375  007376  104003                          ERROP   3
2376  007400  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2377  007406  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
2378  007414  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
2379  007422  001404                          BEQ     18$             ;YES,INITIALIZE RK611
2380  007424  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
2381  007432  104003                          ERROR   3
2382  007434  016237  000004  002014  18$:    MOV     RKBA(R2),PREREG ;GET PREVIOUS CONTENTS
2383  007442  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
2384  007450  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;GET CURRENT VALUE
2385  007456  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2386  007462  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKBA CORRECT
2387  007470  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
2388  007472  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
2389  007500  012737  066637  001332          MOV     #EM1002,EM4N+2
2390  007506  104004                          ERROR   4
2391  007510  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
2392  007512  000241                          CLC                     ;SHIFT IN ZERO
2393  007514  006137  002010                  ROL     CONFIG
2394  007520  005301                          DEC     R1              ;CHECK IF FINISHED
2395  007522  001402                          BEQ     TST11           ;;YES, GO ON TO NEXT TEST
2396  007524  000137  006746                  JMP     1$
2397
2398                                  ;;******************************************************************
2399                                  ;*TEST 11        REGISTER INTERACTION USING BUS ADDRESS (PART 2)
2400                                  ;*
2401                                  ;*      THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
2402                                  ;*      CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
```

```
                                        ;*      COUNT REGISTER TO 0.
                                        ;*
                                        ;*      THE TEST ITSELF WILL CONSIST OF WRITING THE
                                        ;*      BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
                                        ;*      TEST IF BUS ADDRESS IS CORRECT AND THAT NO
                                        ;*      REGISTER INTERACTION TAKES PLACE.
                                        ;*
                                        ;*      177777  177767  177577  173777  077777
                                        ;*      177776  177757  177377  167777
                                        ;*      177775  177737  176777  157777
                                        ;*      177773  177677  175777  137777
                                        ;*
                                        ;**********************************************************************
007530  000004                    TST11:  SCOPE
007532  012737  000144  001200            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
007540  012701  000021                    MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
007544  012737  177776  002010            MOV     #177776,CONFIG  ;LOAD INITIAL CONFIGURATION
007552  012737  063465  001320            MOV     #EM4,EM3N       ;LOAD ERROR MESSAGE
007560  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
007566  012737  007574  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                  ;  SUBTEST LOOP
007574                                  1$:
007574  005062  000002                    CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
007600  013762  002010  000004            MOV     CONFIG,RKBA(R2) ;WRITE RKBA
007606  016237  000004  001126            MOV     RKBA(R2),$BDDAT ;STORE RKBA
007614  013737  002010  001124            MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
007622  042737  000001  001124            BIC     #1,$GDDAT       ;INITIALIZE READ ONLY BITS
007630  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKBA CORRECT
007636  001404                            BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
007640  012737  066637  001322            MOV     #EM1002,EM3N+2  ;LOAD ERROR MESSAGE
007646  104003                            ERROR   3
007650                                  2$:
007650  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
007656  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
007664  001407                            BEQ     3$              ;YES, CONTINUE
007666  012737  000100  001124            MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
007674  012737  067314  001322            MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
007702  104003                            ERROR   3
007704                                  3$:
007704  016237  000002  001126            MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
007712  001406                            BEQ     5$              ;CHECK IF ZERO
007714  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
007720  012737  067335  001322            MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
007726  104003                            ERROR   3
007730                                  5$:
007730  016237  000006  001126            MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
007736  001406                            BEQ     6$              ;CHECK IF ZERO
007740  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
007744  012737  067412  001322            MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
007752  104003                            ERROR   3
007754                                  6$:
007754  016237  000016  001126            MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
007762  001406                            BEQ     7$              ;CHECK IF ZERO
007764  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
007770  012737  067556  001322            MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
```

```
2459  007776  104003                        ERROR   3
2460  010000                        7$:
2461  010000  016237  000010  001126        MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
2462  010006  022737  000100  001126        CMP     #IR,$BDDAT       ;CHECK IF CS2 CORRECT
2463  010014  001407                        BEQ     8$               ;YES,CONTINUE
2464  010016  012737  000100  001124        MOV     #IR,$GDDAT       ;LOAD EXPECTED CONTENTS
2465  010024  012737  067450  001322        MOV     #EM1021,EM3N+2   ;LOAD ERROR MESSAGE
2466  010032  104003                        ERROR   3
2467  010034                        8$:
2468  010034  016237  000012  001126        MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
2469  010042  001406                        BEQ     9$               ;CHECK IF ZERO
2470  010044  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2471  010050  012737  067471  001322        MOV     #EM1022,EM3N+2   ;LOAD ERROR MESSAGE
2472  010056  104003                        ERROR   3
2473  010060  016237  000014  001126  9$:   MOV     RKER(R2),$BDDAT ;STORE ERROR REG
2474  010066  001406                        BEQ     10$              ;CHECK IF ZERO
2475  010070  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2476  010074  012737  067527  001322        MOV     #EM1023,EM3N+2   ;LOAD ERROR MESSAGE
2477  010102  104003                        ERROR   3
2478  010104                        10$:
2479  010104  016237  000020  001126        MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2480  010112  001406                        BEQ     12$              ;CHECK IF ZERO
2481  010114  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2482  010120  012737  067624  001322        MOV     #EM1025,EM3N+2   ;LOAD ERROR MESSAGE
2483  010126  104003                        ERROR   3
2484  010130                        12$:
2485  010130  016237  000026  001126        MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
2486  010136  012737  002000  001124        MOV     #MEWD,$GDDAT     ;LOAD EXPECTED MR1
2487  010144  032737  020000  001126        BIT     #ECCW,$BDDAT
2488  010152  001403                        BEQ     13$
2489  010154  052737  020000  001124        BIS     #ECCW,$GDDAT
2490  010162  023737  001124  001126  13$:  CMP     $GDDAT,$BDDAT    ;CHECK IF MR1 CORRECT
2491  010170  001404                        BEQ     14$              ;YES,ISSUE CONTROLLER CLEAR
2492  010172  012737  067651  001322        MOV     #EM1026,EM3N+2   ;LOAD ERROR MESSAGE
2493  010200  104003                        ERROR   3
2494  010202                        14$:
2495  010202  016237  000032  001126        MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2496  010210  001406                        BEQ     15$              ;CHECK IF ZERO
2497  010212  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2498  010216  012737  067727  001322        MOV     #EM1030,EM3N+2   ;LOAD ERROR MESSAGE
2499  010224  104003                        ERROR   3
2500  010226  016237  000030  001126  15$:  MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2501  010234  012737  004066  001124  16$:  MOV     #4066,$GDDAT     ;USE 4066
2502  010242  023737  001124  001126  17$:  CMP     $GDDAT,$BDDAT    ;CHECK IF ECC POSITION CORRECT
2503  010250  001404                        BEQ     18$              ;YES,INITIALIZE RK611
2504  010252  012737  067702  001322        MOV     #EM1029,EM3N+2   ;LOAD ERROR MESSAGE
2505  010260  104003                        ERROR   3
2506  010262  016237  000004  002014  18$:  MOV     RKBA(R2),PREREG ;GET PREVIOUS CONTENTS
2507  010270  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
2508  010276  016237  000004  001126        MOV     RKBA(R2),$BDDAT ;GET CURRENT VALUE
2509  010304  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2510  010310  023737  001124  001126        CMP     $GDDAT,$BDDAT    ;CHECK IF RKBA CORRECT
2511  010316  001407                        BEQ     19$              ;YES, CHECK IF FINISHED
2512  010320  012737  063403  001330        MOV     #EM3,EM4N        ;LOAD ERROR MESSAGE
2513  010326  012737  066637  001332        MOV     #EM1002,EM4N+2
2514  010334  104004                        ERROR   4
```

```
2515  010336  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
2516  010340  000261                          SEC                     ;SHIFT IN ONE
2517  010342  006137  002010                  ROL     CONFIG
2518  010346  005301                          DEC     R1              ;CHECK IF FINISHED
2519  010350  001402                          BEQ     TST12           ;;YES, GO ON TO NEXT TEST
2520  010352  000137  007574                  JMP     1$
2521
2522                                  ;;*****************************************************************
2523                                  ;*TEST 12       REGISTER INTERACTION USING BUS ADDRESS (PART 3)
2524                                  ;*
2525                                  ;*      THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
2526                                  ;*      CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
2527                                  ;*      COUNT REGISTER TO 0.
2528                                  ;*
2529                                  ;*      THE TEST ITSELF WILL CONSIST OF WRITING THE
2530                                  ;*      BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
2531                                  ;*      TEST IF BUS ADDRESS IS CORRECT AND THAT NO
2532                                  ;*      REGISTER INTERACTION TAKES PLACE.
2533                                  ;*
2534                                  ;*      000001  000037  000777  017777  000000
2535                                  ;*      000003  000077  001777  037777
2536                                  ;*      000007  000177  003777  077777
2537                                  ;*      000017  000377  007777  177777
2538                                  ;*
2539                                  ;;*****************************************************************
2540  010356  000004                  TST12:  SCOPE
2541  010360  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
2542  010366  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
2543  010372  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
2544  010376  012737  063465  001320          MOV     #EM4,EM3N       ;LOAD ERROR MESSAGE
2545  010404  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
2546  010412  012737  010420  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
2547                                                                  ;  SUBTEST LOOP
2548
2549  010420                          1$:
2550  010420  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
2551  010424  013762  002010  000004          MOV     CONFIG,RKBA(R2) ;WRITE RKBA
2552  010432  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE RKBA
2553  010440  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
2554  010446  042737  000001  001124          BIC     #1,$GDDAT       ;INITIALIZE READ ONLY BITS
2555  010454  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKBA CORRECT
2556  010462  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
2557  010464  012737  066637  001322          MOV     #EM1002,EM3N+2  ;LOAD ERROR MESSAGE
2558  010472  104003                          ERROR   3
2559  010474                          2$:
2560  010474  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
2561  010502  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
2562  010510  001407                          BEQ     3$              ;YES, CONTINUE
2563  010512  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
2564  010520  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
2565  010526  104003                          ERROR   3
2566  010530                          3$:
2567  010530  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
2568  010536  001406                          BEQ     5$              ;CHECK IF ZERO
2569  010540  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2570  010544  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
```

```
2571  010552  104003                          ERROR   3
2572  010554                          5$:
2573  010554  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
2574  010562  001406                          BEQ     6$              ;CHECK IF ZERO
2575  010564  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2576  010570  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
2577  010576  104003                          ERROR   3
2578  010600                          6$:
2579  010600  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2580  010606  001406                          BEQ     7$              ;CHECK IF ZERO
2581  010610  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2582  010614  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
2583  010622  104003                          ERROR   3
2584  010624                          7$:
2585  010624  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
2586  010632  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
2587  010640  001407                          BEQ     8$              ;YES,CONTINUE
2588  010642  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
2589  010650  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
2590  010656  104003                          ERROR   3
2591  010660                          8$:
2592  010660  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
2593  010666  001406                          BEQ     9$              ;CHECK IF ZERO
2594  010670  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2595  010674  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
2596  010702  104003                          ERROR   3
2597  010704  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
2598  010712  001406                          BEQ     10$             ;CHECK IF ZERO
2599  010714  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2600  010720  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
2601  010726  104003                          ERROR   3
2602  010730                          10$:
2603  010730  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2604  010736  001406                          BEQ     12$             ;CHECK IF ZERO
2605  010740  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2606  010744  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
2607  010752  104003                          ERROR   3
2608  010754                          12$:
2609  010754  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
2610  010762  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
2611  010770  032737  020000  001126          BIT     #ECCW,$BDDAT
2612  010776  001403                          BEQ     13$
2613  011000  052737  020000  001124          BIS     #ECCW,$GDDAT
2614  011006  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
2615  011014  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
2616  011016  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
2617  011024  104003                          ERROR   3
2618  011026                          14$:
2619  011026  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2620  011034  001406                          BEQ     15$             ;CHECK IF ZERO
2621  011036  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2622  011042  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
2623  011050  104003                          ERROR   3
2624  011052  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2625  011060  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
2626  011066  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
```

```
2627  011074  001404                          BEQ     18$                ;YES,INITIALIZE RK611
2628  011076  012737  067702  001322          MOV     #EM1029,EM3N+2     ;LOAD ERROR MESSAGE
2629  011104  104003                          ERROR   3
2630  011106  016237  000004  002014  18$:    MOV     RKBA(R2),PREREG    ;GET PREVIOUS CONTENTS
2631  011114  012762  100000  000000          MOV     #CCLR,RKCS1(R2)    ;CLEAR RK611 CONTROLLER
2632  011122  016237  000004  001126          MOV     RKBA(R2),$BDDAT    ;GET CURRENT VALUE
2633  011130  005037  001124                  CLR     $GDDAT             ;LOAD EXPECTED CONTENTS
2634  011134  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;CHECK IF RKBA CORRECT
2635  011142  001407                          BEQ     19$                ;YES, CHECK IF FINISHED
2636  011144  012737  063403  001330          MOV     #EM3,EM4N          ;LOAD ERROR MESSAGE
2637  011152  012737  066637  001332          MOV     #EM1002,EM4N+2
2638  011160  104004                          ERROR   4
2639  011162  104415                  19$:    SCOP1                      ;CHECK IF LOOP ON ERROR
2640  011164  000261                          SEC                        ;SHIFT IN ONE
2641  011166  006137  002010                  ROL     CONFIG
2642  011172  005301                          DEC     R1                 ;CHECK IF FINISHED
2643  011174  001402                          BEQ     TST13              ;;YES, GO ON TO NEXT TEST
2644  011176  000137  010420                  JMP     1$
2645
2646                                  ;;*****************************************************************
2647                                  ;*TEST 13       REGISTER INTERACTION USING BUS ADDRESS (PART 4)
2648                                  ;*
2649                                  ;*      THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
2650                                  ;*      CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
2651                                  ;*      COUNT REGISTER TO 0.
2652                                  ;*
2653                                  ;*      THE TEST ITSELF WILL CONSIST OF WRITING THE
2654                                  ;*      BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
2655                                  ;*      TEST IF BUS ADDRESS IS CORRECT AND THAT NO
2656                                  ;*      REGISTER INTERACTION TAKES PLACE.
2657                                  ;*
2658                                  ;*      100000  174000  177600  177770  000000
2659                                  ;*      140000  176000  177700  177774
2660                                  ;*      160000  177000  177740  177776
2661                                  ;*      170000  177400  177760  177777
2662                                  ;*
2663                                  ;*****************************************************************
2664  011202  000004          TST13:  SCOPE
2665  011204  012737  000144  001200          MOV     #100.,STIMES       ;;DO 100. ITERATIONS
2666  011212  012701  000021                  MOV     #17.,R             ;LOAD NUMBER OF PATTERNS
2667  011216  005037  002010                  CLR     CONFIG             ;LOAD INITIAL CONFIGURATION
2668  011222  012737  063465  001320          MOV     #EM4,EM3N          ;LOAD ERROR MESSAGE
2669  011230  012762  100000  000000          MOV     #CCLR,RKCS1(R2)    ;CLEAR RK611 WITH CONTROLLER CLEAR
2670  011236  012737  011244  001110          MOV     #1$,$LPERR         ;LOAD LOOP ON ERROR LOCATION FOR
2671                                                                     ;  SUBTEST LOOP
2672
2673  011244                          1$:
2674  011244  005062  000002                  CLR     RKWC(R2)           ;CLEAR WORD COUNT REG.
2675  011250  013762  002010  000004          MOV     CONFIG,RKBA(R2)    ;WRITE RKBA
2676  011256  016237  000004  001126          MOV     RKBA(R2),$BDDAT    ;STORE RKBA
2677  011264  013737  002010  001124          MOV     CONFIG,$GDDAT      ;PREPARE EXPECTED RESULTS
2678  011272  042737  000001  001124          BIC     #1,$GDDAT          ;INITIALIZE READ ONLY BITS
2679  011300  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;CHECK IF RKBA CORRECT
2680  011306  001404                          BEQ     2$                 ;YES,TEST IF ANY OTHER REG MODIFIED
2681  011310  012737  066637  001322          MOV     #EM1002,EM3N+2     ;LOAD ERROR MESSAGE
2682  011316  104003                          ERROR   3
```

```
2683  011320                          2$:
2684  011320  016237  000000  001126        MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
2685  011326  022737  000200  001126        CMP     #RDY,$BDDAT      ;CHECK IF CS1 CORRECT
2686  011334  001407                        BEQ     3$               ;YES, CONTINUE
2687  011336  012737  000100  001124        MOV     #IR,$GDDAT       ;LOAD EXPECTED RESULTS
2688  011344  012737  067314  001322        MOV     #EM1017,EM3N+2   ;LOAD ERROR MESSAGE
2689  011352  104003                        ERROR   3
2690  011354                          3$:
2691  011354  016237  000002  001126        MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
2692  011362  001406                        BEQ     5$               ;CHECK IF ZERO
2693  011364  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2694  011370  012737  067335  001322        MOV     #EM1018,EM3N+2   ;LOAD ERROR MESSAGE
2695  011376  104003                        ERROR   3
2696  011400                          5$:
2697  011400  016237  000006  001126        MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
2698  011406  001406                        BEQ     6$               ;CHECK IF ZERO
2699  011410  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2700  011414  012737  067412  001322        MOV     #EM1020,EM3N+2   ;LOAD ERROR MESSAGE
2701  011422  104003                        ERROR   3
2702  011424                          6$:
2703  011424  016237  000016  001126        MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2704  011432  001406                        BEQ     7$               ;CHECK IF ZERO
2705  011434  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2706  011440  012737  067556  001322        MOV     #EM1024,EM3N+2   ;LOAD ERROR MESSAGE
2707  011446  104003                        ERROR   3
2708  011450                          7$:
2709  011450  016237  000010  001126        MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
2710  011456  022737  000100  001126        CMP     #IR,$BDDAT       ;CHECK IF CS2 CORRECT
2711  011464  001407                        BEQ     8$               ;YES,CONTINUE
2712  011466  012737  000100  001124        MOV     #IR,$GDDAT       ;LOAD EXPECTED CONTENTS
2713  011474  012737  067450  001322        MOV     #EM1021,EM3N+2   ;LOAD ERROR MESSAGE
2714  011502  104003                        ERROR   3
2715  011504                          8$:
2716  011504  016237  000012  001126        MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
2717  011512  001406                        BEQ     9$               ;CHECK IF ZERO
2718  011514  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2719  011520  012737  067471  001322        MOV     #EM1022,EM3N+2   ;LOAD ERROR MESSAGE
2720  011526  104003                        ERROR   3
2721  011530  016237  000014  001126  9$:   MOV     RKER(R2),$BDDAT ;STORE ERROR REG
2722  011536  001406                        BEQ     10$              ;CHECK IF ZERO
2723  011540  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2724  011544  012737  067527  001322        MOV     #EM1023,EM3N+2   ;LOAD ERROR MESSAGE
2725  011552  104003                        ERROR   3
2726  011554                          10$:
2727  011554  016237  000020  001126        MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2728  011562  001406                        BEQ     12$              ;CHECK IF ZERO
2729  011564  005037  001124                CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
2730  011570  012737  067624  001322        MOV     #EM1025,EM3N+2   ;LOAD ERROR MESSAGE
2731  011576  104003                        ERROR   3
2732  011600                          12$:
2733  011600  016237  000026  001126        MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
2734  011606  012737  002000  001124        MOV     #MEWD,$GDDAT     ;LOAD EXPECTED MR1
2735  011614  032737  020000  001126        BIT     #ECCW,$BDDAT
2736  011622  001403                        BEQ     13$
2737  011624  052737  020000  001124        BIS     #ECCW,$GDDAT
2738  011632  023737  001124  001126  13$:  CMP     $GDDAT,$BDDAT    ;CHECK IF MR1 CORRECT
```

```
2739  011640  001404                          BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
2740  011642  012737  067651  001322          MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
2741  011650  104003                          ERROR   3
2742  011652                          14$:
2743  011652  016237  000032  001126          MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
2744  011660  001406                          BEQ     15$                 ;CHECK IF ZERO
2745  011662  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
2746  011666  012737  067727  001322          MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
2747  011674  104003                          ERROR   3
2748  011676  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
2749  011704  012737  004066  001124  16$:    MOV     #4066,$GDDAT        ;USE 4066
2750  011712  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
2751  011720  001404                          BEQ     18$                 ;YES,INITIALIZE RK611
2752  011722  012737  067702  001322          MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
2753  011730  104003                          ERROR   3
2754  011732  016237  000004  002014  18$:    MOV     RKBA(R2),PREREG     ;GET PREVIOUS CONTENTS
2755  011740  012762  100000  000000          MOV     #CCLR,RKCS1(R2)     ;CLEAR RK611 CONTROLLER
2756  011746  016237  000004  001126          MOV     RKBA(R2),$BDDAT     ;GET CURRENT VALUE
2757  011754  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
2758  011760  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;CHECK IF RKBA CORRECT
2759  011766  001407                          BEQ     19$                 ;YES, CHECK IF FINISHED
2760  011770  012737  063403  001330          MOV     #EM3,EM4N           ;LOAD ERROR MESSAGE
2761  011776  012737  066637  001332          MOV     #EM1002,EM4N+2
2762  012004  104004                          ERROR   4
2763  012006  104415                  19$:    SCOP1                       ;CHECK IF LOOP ON ERROR
2764  012010  000261                          SEC                         ;SHIFT IN ONE
2765  012012  006037  002010                  ROR     CONFIG
2766  012016  005301                          DEC     R1                  ;CHECK IF FINISHED
2767  012020  001402                          BEQ     TST14               ;;YES, GO ON TO NEXT TEST
2768  012022  000137  011244                  JMP     1$
2769
2770                                  ;;*****************************************************************
2771                                  ;*TEST 14       REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)
2772                                  ;*
2773                                  ;*      THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
2774                                  ;*      BY WRITING THE WORD COUNT TO ZERO.
2775                                  ;*
2776                                  ;*      THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
2777                                  ;*      AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
2778                                  ;*      SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
2779                                  ;*      THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
2780                                  ;*      REGISTER INTERACTION TAKES PLACE.
2781                                  ;*
2782                                  ;*      000000  000020  000400  010000
2783                                  ;*      000C02  000040  001000  020000
2784                                  ;*      000004  000100  002000  040000
2785                                  ;*      000010  000200  004000
2786                                  ;*
2787                                  ;*****************************************************************
2788  012026  000004          TST14:  SCOPE
2789  012030  012737  000144  001200          MOV     #100.,$TIMES        ;;DO 100. ITERATIONS
2790  012036  012701  000021                  MOV     #17.,R1             ;LOAD NUMBER OF PATTERNS
2791  012042  005037  002010                  CLR     CONFIG              ;LOAD INITIAL CONFIGURATION
2792  012046  012737  000001  002012          MOV     #000001,CONFG1
2793  012054  012737  064321  001320          MOV     #EM16,EM3N          ;LOAD ERROR MESSAGE
2794  012062  012762  100000  000000          MOV     #CCLR,RKCS1(R2)     ;CLEAR RK611 WITH CONTROLLER CLEAR
```

```
2795  012070  012737  012076  00:110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
2796                                                                  ;  SUBTEST LOOP
2797
2798  012076                          1$:
2799  012076  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
2800  012102  013762  002010  000000          MOV     CONFIG,RKCS1(R2)        ;WRITE RKCS1
2801  012110  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE RKCS1
2802  012116  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
2803  012124  042737  064000  001124          BIC     #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
2804  012132  052737  000200  001124          BIS     #RDY,$GDDAT
2805  012140  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS1 CORRECT
2806  012146  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
2807  012150  012737  066574  001322          MOV     #EM1000,EM3N+2  ;LOAD ERROR MESSAGE
2808  012156  104003                          ERROR   3
2809  012160                          2$:
2810  012160  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
2811  012166  001406                          BEQ     4$              ;CHECK IF ZERO
2812  012170  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2813  012174  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
2814  012202  104003                          ERROR   3
2815  012204                          4$:
2816  012204  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
2817  012212  001406                          BEQ     5$              ;CHECK IF ZERO
2818  012214  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2819  012220  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
2820  012226  104003                          ERROR   3
2821  012230                          5$:
2822  012230  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
2823  012236  001406                          BEQ     6$              ;CHECK IF ZERO
2824  012240  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2825  012244  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
2826  012252  104003                          ERROR   3
2827  012254                          6$:
2828  012254  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2829  012262  001406                          BEQ     7$              ;CHECK IF ZERO
2830  012264  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2831  012270  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
2832  012276  104003                          ERROR   3
2833  012300                          7$:
2834  012300  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
2835  012306  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
2836  012314  001407                          BEQ     8$              ;YES,CONTINUE
2837  012316  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
2838  012324  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
2839  012332  104003                          ERROR   3
2840  012334                          8$:
2841  012334  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
2842  012342  001406                          BEQ     9$              ;CHECK IF ZERO
2843  012344  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2844  012350  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
2845  012356  104003                          ERROR   3
2846  012360  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
2847  012366  001406                          BEQ     10$             ;CHECK IF ZERO
2848  012370  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2849  012374  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
2850  012402  104003                          ERROR   3
```

```
2851  012404                                  10$:
2852  012404  016237  000020  001126                MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2853  012412  001406                                BEQ     12$               ;CHECK IF ZERO
2854  012414  005037  001124                        CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
2855  012420  012737  067624  001322                MOV     #EM1025,EM3N+2    ;LOAD ERROR MESSAGE
2856  012426  104003                                ERROR   3
2857  012430                                  12$:
2858  012430  016237  000026  001126                MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
2859  012436  012737  002000  001124                MOV     #MEWD,$GDDAT     ;LOAD EXPECTED MR1
2860  012444  032737  020000  001126                BIT     #ECCW,$BDDAT
2861  012452  001403                                BEQ     13$
2862  012454  052737  020000  001124                BIS     #ECCW,$GDDAT
2863  012462  023737  001124  001126  13$:         CMP     $GDDAT,$BDDAT    ;CHECK IF MR1 CORRECT
2864  012470  001404                                BEQ     14$              ;YES,ISSUE CONTROLLER CLEAR
2865  012472  012737  067651  001322                MOV     #EM1026,EM3N+2   ;LOAD ERROR MESSAGE
2866  012500  104003                                ERROR   3
2867  012502                                  14$:
2868  012502  016237  000032  001126                MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2869  012510  001406                                BEQ     15$               ;CHECK IF ZERO
2870  012512  005037  001124                        CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
2871  012516  012737  067727  001322                MOV     #EM1030,EM3N+2    ;LOAD ERROR MESSAGE
2872  012524  104003                                ERROR   3
2873  012526  016237  000030  001126  15$:         MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2874  012534  032737  010000  002010                BIT     #CFMT,CONFIG      ;CHECK IF IN 18 BIT FORMAT
2875  012542  001404                                BEQ     16$               ;NO, USE 4066
2876  012544  012737  005066  001124                MOV     #5066,$GDDAT      ;USE 5066
2877  012552  000403                                BR      17$     ;CHECK IF POSITION CORRECT
2878
2879  012554  012737  004066  001124  16$:         MOV     #4066,$GDDAT      ;USE 4066
2880  012562  023737  001124  001126  17$:         CMP     $GDDAT,$BDDAT     ;CHECK IF ECC POSITION CORRECT
2881  012570  001404                                BEQ     18$               ;YES,INITIALIZE RK611
2882  012572  012737  067702  001322                MOV     #EM1029,EM3N+2    ;LOAD ERROR MESSAGE
2883  012600  104003                                ERROR   3
2884  012602  016237  000000  002014  18$:         MOV     RKCS1(R2),PREREG          ;GET PREVIOUS CONTENTS
2885  012610  012762  100000  000000                MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
2886  012616  016237  000000  001126                MOV     RKCS1(R2),$BDDAT          ;GET CURRENT VALUE
2887  012624  012737  000200  001124                MOV     #RDY,$GDDAT       ;LOAD EXPECTED CONTENTS
2888  012632  023737  001124  001126                CMP     $GDDAT,$BDDAT     ;CHECK IF RKCS1 CORRECT
2889  012640  001407                                BEQ     19$               ;YES, CHECK IF FINISHED
2890  012642  012737  063403  001330                MOV     #EM3,EM4N         ;LOAD ERROR MESSAGE
2891  012650  012737  066574  001332                MOV     #EM1000,EM4N+2
2892  012656  104004                                ERROR   4
2893  012660  104415                          19$:  SCOP1                     ;CHECK IF LOOP ON ERROR
2894  012662  000241                                CLC                       ;SHIFT IN ZERO
2895  012664  006137  002012                        ROL     CONFG1
2896  012670  013737  002012  002010                MOV     CONFG1,CONFIG   ;MAKE SURE CONTROLLER CLEAR AMP
2897  012676  042737  100001  002010                BIC     #CCLR!GO,CONFIG ;   GO DO NOT SET
2898  012704  005301                                DEC     R1                ;CHECK IF FINISHED
2899  012706  001402                                BEQ     TST15             ;;YES, GO ON TO NEXT TEST
2900  012710  000137  012076                        JMP     1$
2901
2902                                          ;;*****************************************************************
2903                                          ;*TEST 15       REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)
2904                                          ;*
2905                                          ;*       THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
2906                                          ;*       BY WRITING THE WORD COUNT TO ZERO.
```

```
2907                                    ;*
2908                                    ;*      THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
2909                                    ;*      AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
2910                                    ;*      SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
2911                                    ;*      THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
2912                                    ;*      REGISTER INTERACTION TAKES PLACE.
2913                                    ;*
2914                                    ;*      077776  077756  077376  067776
2915                                    ;*      077774  077736  076776  057776
2916                                    ;*      077772  077676  075776  037776
2917                                    ;*      077766  077576  073776
2918                                    ;*
2919                                    ;**************************************************************
2920  012714  000004                    TST15:  SCOPE
2921  012716  012737  000144  001200            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
2922  012724  012701  000021                    MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
2923  012730  012737  077776  002010            MOV     #077776,CONFIG  ;LOAD INITIAL CONFIGURATION
2924  012736  012737  177776  002012            MOV     #177776,CONFG1
2925  012744  012737  064321  001320            MOV     #EM16,EM3N      ;LOAD ERROR MESSAGE
2926  012752  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
2927  012760  012737  012766  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
2928                                                                    ;  SUBTEST LOOP
2929
2930  012766                            1$:
2931  012766  005062  000002                    CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
2932  012772  013762  002010  000000            MOV     CONFIG,RKCS1(R2)        ;WRITE RKCS1
2933  013000  016237  000000  001126            MOV     RKCS1(R2),$BDDAT        ;STORE RKCS1
2934  013006  013737  002010  001124            MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
2935  013014  042737  064000  001124            BIC     #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
2936  013022  052737  000200  001124            BIS     #RDY,$GDDAT
2937  013030  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS1 CORRECT
2938  013036  001404                            BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
2939  013040  012737  066574  001322            MOV     #EM1000,EM3N+2  ;LOAD ERROR MESSAGE
2940  013046  104003                            ERROR   3
2941  013050                            2$:
2942  013050  016237  000004  001126            MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
2943  013056  001406                            BEQ     4$              ;CHECK IF ZERO
2944  013060  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2945  013064  012737  067365  001322            MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
2946  013072  104003                            ERROR   3
2947  013074                            4$:
2949  013074  016237  000002  001126            MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
2949  013102  001406                            BEQ     5$              ;CHECK IF ZERO
2950  013104  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2951  013110  012737  067335    1322           MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
2952  013116  104003                            ERROR   3
2953  013120                            5$:
2954  013120  016237  000006  001126            MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
2955  013126  001406                            BEQ     6$              ;CHECK IF ZERO
2956  013130  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
2957  013134  012737  067412  001322            MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
2958  013142  104003                            ERROR   3
2959  013144                            6$:
2960  013144  016237  000016  001126            MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2961  013152  001406                            BEQ     7$              ;CHECK IF ZERO
2962  013154  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
```

```
 2963  013160  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
 2964  013166  104003                          ERROR   3
 2965  013170                          7$:
 2966  013170  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
 2967  013176  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
 2968  013204  001407                          BEQ     8$              ;YES,CONTINUE
 2969  013206  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
 2970  013214  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
 2971  013222  104003                          ERROR   3
 2972  013224                          8$:
 2973  013224  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
 2974  013232  001406                          BEQ     9$              ;CHECK IF ZERO
 2975  013234  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 2976  013240  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 2977  013246  104003                          ERROR   3
 2978  013250  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
 2979  013256  001406                          BEQ     10$             ;CHECK IF ZERO
 2980  013260  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 2981  013264  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
 2982  013272  104003                          ERROR   3
 2983  013274                          10$:
 2984  013274  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
 2985  013302  001406                          BEQ     12$             ;CHECK IF ZERO
 2986  013304  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 2987  013310  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
 2988  013316  104003                          ERROR   3
 2989  013320                          12$:
 2990  013320  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
 2991  013326  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
 2992  013334  032737  020000  001126          BIT     #ECCW,$BDDAT
 2993  013342  001403                          BEQ     13$
 2994  013344  052737  020000  001124          BIS     #ECCW,$GDDAT
 2995  013352  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
 2996  013360  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
 2997  013362  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
 2998  013370  104003                          ERROR   3
 2999  013372                          14$:
 3000  013372  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
 3001  013400  001406                          BEQ     15$             ;CHECK IF ZERO
 3002  013402  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 3003  013406  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
 3004  013414  104003                          ERROR   3
 3005  013416  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
 3006  013424  032737  010000  002010          BIT     #CFMT,CONFIG    ;CHECK IF IN 18 BIT FORMAT
 3007  013432  001404                          BEQ     16$             ;NO, USE 4066
 3008  013434  012737  005066  001124          MOV     #5066,$GDDAT    ;USE 5066
 3009  013442  000403                          BR      17$     ;CHECK IF POSITION CORRECT
 3010
 3011  013444  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
 3012  013452  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
 3013  013460  001404                          BEQ     18$             ;YES,INITIALIZE RK611
 3014  013462  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
 3015  013470  104003                          ERROR   3
 3016  013472  016237  000000  002014  18$:    MOV     RKCS1(R2),PREREG        ;GET PREVIOUS CONTENTS
 3017  013500  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
 3018  013506  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;GET CURRENT VALUE
```

```
3019  013514  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
3020  013522  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS1 CORRECT
3021  013530  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
3022  013532  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
3023  013540  012737  066574  001332          MOV     #EM1000,EM4N+2
3024  013546  104004                          ERROR   4
3025  013550  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
3026  013552  000261                          SEC                     ;SHIFT IN ONE
3027  013554  006137  002012                  ROL     CONFG1
3028  013560  013737  002012  002010          MOV     CONFG1,CONFIG   ;MAKE SURE CONTROLLER CLEAR AMP
3029  013566  042737  100001  002010          BIC     #CCLR!GO,CONFIG ;   GO DO NOT SET
3030  013574  005301                          DEC     R1              ;CHECK IF FINISHED
3031  013576  001402                          BEQ     TST16           ;;YES, GO ON TO NEXT TEST
3032  013600  000137  012766                  JMP     1$
3033
3034                          ;*****************************************************************
3035                          ;*TEST 16       REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)
3036                          ;*
3037                          ;*      THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
3038                          ;*      BY WRITING THE WORD COUNT TO ZERO.
3039                          ;*
3040                          ;*      THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
3041                          ;*      AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
3042                          ;*      SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
3043                          ;*      THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
3044                          ;*      REGISTER INTERACTION TAKES PLACE.
3045                          ;*
3046                          ;*      000002  000076  001776  037776
3047                          ;*      000006  000176  003776  077776
3048                          ;*      000016  000376  007776  000000
3049                          ;*      000036  000776  017776
3050                          ;*
3051                          ;*****************************************************************
3052  013604  000004          TST16:  SCOPE
3053  013606  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3054  013614  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
3055  013620  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
3056  013624  005037  002012                  CLR     CONFG1
3057  013630  012737  064321  001320          MOV     #EM16,EM3N      ;LOAD ERROR MESSAGE
3058  013636  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
3059  013644  012737  013652  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
3060                                                                  ;   SUBTEST LOOP
3061
3062  013652                          1$:
3063  013652  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
3064  013656  013762  002010  000000          MOV     CONFIG,RKCS1(R2)        ;WRITE RKCS1
3065  013664  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE RKCS1
3066  013672  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
3067  013700  042737  064000  001124          BIC     #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
3068  013706  052737  000200  001124          BIS     #RDY,$GDDAT
3069  013714  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS1 CORRECT
3070  013722  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
3071  013724  012737  066574  001322          MOV     #EM1000,EM3N+2  ;LOAD ERROR MESSAGE
3072  013732  104003                          ERROR   3
3073  013734                          2$:
3074  013734  016237  000004  001126          MOV     RMBA(R2),$BDDAT ;STORE BUS AND REG
```

```
3075  013742  001406                        BEQ     4$                  ;CHECK IF ZERO
3076  013744  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3077  013750  012737  067365  001322        MOV     #EM1019,EM3N+2      ;LOAD ERROR MESSAGE
3078  013756  104003                        ERROR   3
3079  013760                          4$:
3080  013760  016237  000002  001126        MOV     RKWC(R2),$BDDAT     ;STORE WORD COUNT REG
3081  013766  001406                        BEQ     5$                  ;CHECK IF ZERO
3082  013770  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3083  013774  012737  067335  001322        MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
3084  014002  104003                        ERROR   3
3085  014004                          5$:
3086  014004  016237  000006  001126        MOV     RKDA(R2),$BDDAT     ;STORE DISK AVERAGE REG
3087  014012  001406                        BEQ     6$                  ;CHECK IF ZERO
3088  014014  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3089  014020  012737  067412  001322        MOV     #EM1020,EM3N+2      ;LOAD ERROR MESSAGE
3090  014026  104003                        ERROR   3
3091  014030                          6$:
3092  014030  016237  000016  001126        MOV     RKASOF(R2),$BDDAT   ;STORE ATTENTION SUMMARY/OFFSET REG.
3093  014036  001406                        BEQ     7$                  ;CHECK IF ZERO
3094  014040  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3095  014044  012737  067556  001322        MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
3096  014052  104003                        ERROR   3
3097  014054                          7$:
3098  014054  016237  000010  001126        MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
3099  014062  022737  000100  001126        CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
3100  014070  001407                        BEQ     8$                  ;YES,CONTINUE
3101  014072  012737  000100  001124        MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
3102  014100  012737  067450  001322        MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
3103  014106  104003                        ERROR   3
3104  014110                          8$:
3105  014110  016237  000012  001126        MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
3106  014116  001406                        BEQ     9$                  ;CHECK IF ZERO
3107  014120  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3108  014124  012737  067471  001322        MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
3109  014132  104003                        ERROR   3
3110  014134  016237  000014  001126  9$:   MOV     RKER(R2),$BDDAT     ;STORE ERROR REG
3111  014142  001406                        BEQ     10$                 ;CHECK IF ZERO
3112  014144  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3113  014150  012737  067527  001322        MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
3114  014156  104003                        ERROR   3
3115  014160                         10$:
3116  014160  016237  000020  001126        MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG
3117  014166  001406                        BEQ     12$                 ;CHECK IF ZERO
3118  014170  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3119  014174  012737  067624  001322        MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
3120  014202  104003                        ERROR   3
3121  014204                         12$:
3122  014204  016237  000026  001126        MOV     RKMR1(R2),$BDDAT    ;STORE MAINTENANCE REG.1
3123  014212  012737  002000  001124        MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
3124  014220  032737  020000  001126        BIT     #ECCW,$BDDAT
3125  014226  001403                        BEQ     13$
3126  014230  052737  020000  001124        BIS     #ECCW,$GDDAT
3127  014236  023737  001124  001126  13$:  CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
3128  014244  001404                        BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
3129  014246  012737  067651  001322        MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
3130  014254  104003                        ERROR   3
```

```
3131  014256                          14$:
3132  014256  016237  000032  001126        MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3133  014264  001406                        BEQ     15$               ;CHECK IF ZERO
3134  014266  005037  001124              CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3135  014272  012737  067727  001322        MOV     #EM1030,EM3N+2    ;LOAD ERROR MESSAGE
3136  014300  104003                        ERROR   3
3137  014302  016237  000030  001126  15$:  MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3138  014310  032737  010000  002010        BIT     #CFMT,CONFIG      ;CHECK IF IN 18 BIT FORMAT
3139  014316  001404                        BEQ     16$               ;NO, USE 4066
3140  014320  012737  005066  001124        MOV     #5066,$GDDAT      ;USE 5066
3141  014326  000403                        BR      17$     ;CHECK IF POSITION CORRECT
3142
3143  014330  012737  004066  001124  16$:  MOV     #4066,$GDDAT      ;USE 4066
3144  014336  023737  001124  001126  17$:  CMP     $GDDAT,$BDDAT     ;CHECK IF ECC POSITION CORRECT
3145  014344  001404                        BEQ     18$               ;YES,INITIALIZE RK611
3146  014346  012737  067702  001322        MOV     #EM1029,EM3N+2    ;LOAD ERROR MESSAGE
3147  014354  104003                        ERROR   3
3148  014356  016237  000000  002014  18$:  MOV     RKCS1(R2),PREREG          ;GET PREVIOUS CONTENTS
3149  014364  012762  100000  000000        MOV     #CCLR,RKCS1(R2)   ;CLEAR RK611 CONTROLLER
3150  014372  016237  000000  001126        MOV     RKCS1(R2),$BDDAT          ;GET CURRENT VALUE
3151  014400  012737  000200  001124        MOV     #RDY,$GDDAT       ;LOAD EXPECTED CONTENTS
3152  014406  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;CHECK IF RKCS1 CORRECT
3153  014414  001407                        BEQ     19$               ;YES, CHECK IF FINISHED
3154  014416  012737  063403  001330        MOV     #EM3,EM4N         ;LOAD ERROR MESSAGE
3155  014424  012737  066574  001332        MOV     #EM1000,EM4N+2
3156  014432  104004                        ERROR   4
3157  014434  104415                  19$:  SCOPE                     ;CHECK IF LOOP ON ERROR
3158  014436  000261                        SEC                       ;SHIFT IN ONE
3159  014440  006137  002012              ROL     CONFG1
3160  014444  013737  002012  002010        MOV     CONFG1,CONFIG     ;MAKE SURE CONTROLLER CLEAR AMP
3161  014452  042737  100001  002010        BIC     #CCLR!GO,CONFIG   ;  GO DO NOT SET
3162  014460  005301                        DEC     R1                ;CHECK IF FINISHED
3163  014462  001402                        BEQ     TST17             ;;YES, GO ON TO NEXT TEST
3164  014464  000137  013652                JMP     1$
3165
3166                                  ;*****************************************************************
3167                                  ;*TEST 17       REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)
3168                                  ;*
3169                                  ;*      THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
3170                                  ;*      BY WRITING THE WORD COUNT TO ZERO.
3171                                  ;*
3172                                  ;*      THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
3173                                  ;*      AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
3174                                  ;*      SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
3175                                  ;*      THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
3176                                  ;*      REGISTER INTERACTION TAKES PLACE.
3177                                  ;*
3178                                  ;*      000000  074000  077600  077770
3179                                  ;*      040000  076000  077700  077774
3180                                  ;*      060000  077000  077740  077776
3181                                  ;*      070000  077400  077760
3182                                  ;*
3183                                  ;*****************************************************************
3184  014470  000004                  TST17:  SCOPE
3185  014472  012737  000144  001200        MOV     #100.,$TIMES      ;;DO 100. ITERATIONS
3186  014500  012701  000021              MOV     #17.,R1           ;LOAD NUMBER OF PATTERNS
```

```
  3187  014504  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
  3188  014510  005037  002012                  CLR     CONFG1
  3189  014514  012737  064321  001320          MOV     #EM16,EM3N      ;LOAD ERROR MESSAGE
  3190  014522  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
  3191  014530  012737  014536  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
  3192                                                                  ;  SUBTEST LOOP
  3193
  3194  014536                          1$:
  3195  014536  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
  3196  014542  013762  002010  000000          MOV     CONFIG,RKCS1(R2)        ;WRITE RKCS1
  3197  014550  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE RKCS1
  3198  014556  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
  3199  014564  042737  064000  001124          BIC     #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
  3200  014572  052737  000200  001124          BIS     #RDY,$GDDAT
  3201  014600  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS1 CORRECT
  3202  014606  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
  3203  014610  012737  066574  001322          MOV     #EM1000,EM3N+2  ;LOAD ERROR MESSAGE
  3204  014616  104003                          ERROR   3
  3205  014620                          2$:
  3206  014620  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
  3207  014626  001406                          BEQ     4$              ;CHECK IF ZERO
  3208  014630  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
  3209  014634  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
  3210  014642  104003                          ERROR   3
  3211  014644                          4$:
  3212  014644  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
  3213  014652  001406                          BEQ     5$              ;CHECK IF ZERO
  3214  014654  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
  3215  014660  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
  3216  014666  104003                          ERROR   3
  3217  014670                          5$:
  3218  014670  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
  3219  014676  001406                          BEQ     6$              ;CHECK IF ZERO
  3220  014700  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
  3221  014704  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
  3222  014712  104003                          ERROR   3
  3223  014714                          6$:
  3224  014714  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
  3225  014722  001406                          BEQ     7$              ;CHECK IF ZERO
  3226  014724  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
  3227  014730  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
  3228  014736  104003                          ERROR   3
  3229  014740                          7$:
  3230  014740  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
  3231  014746  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
  3232  014754  001407                          BEQ     8$              ;YES,CONTINUE
  3233  014756  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
  3234  014764  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
  3235  014772  104003                          ERROR   3
  3236  014774                          8$:
  3237  014774  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
  3238  015002  001406                          BEQ     9$              ;CHECK IF ZERO
  3239  015004  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
  3240  015010  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
  3241  015016  104003                          ERROR   3
  3242  015020  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
```

```
3243  015026  001406                          BEQ     10$                 ;CHECK IF ZERO
3244  015030  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3245  015034  012737  067527  001322          MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
3246  015042  104003                          ERROR   3
3247  015044                          10$:
3248  015044  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
3249  015052  001406                          BEQ     12$                 ;CHECK IF ZERO
3250  015054  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3251  015060  012737  067624  001322          MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
3252  015066  104003                          ERROR   3
3253  015070                          12$:
3254  015070  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
3255  015076  012737  002000  001124          MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
3256  015104  032737  020000  001126          BIT     #ECCW,$BDDAT
3257  015112  001403                          BEQ     13$
3258  015114  052737  020000  001124          BIS     #ECCW,$GDDAT
3259  015122  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
3260  015130  001404                          BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
3261  015132  012737  067651  001322          MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
3262  015140  104003                          ERROR   3
3263  015142                          14$:
3264  015142  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3265  015150  001406                          BEQ     15$                 ;CHECK IF ZERO
3266  015152  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3267  015156  012737  067727  001322          MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
3268  015164  104003                          ERROR   3
3269  015166  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3270  015174  032737  010000  002010          BIT     #CFMT,CONFIG        ;CHECK IF IN 18 BIT FORMAT
3271  015202  001404                          BEQ     16$                 ;NO, USE 4066
3272  015204  012737  005066  001124          MOV     #5066,$GDDAT        ;USE 5066
3273  015212  000403                          BR      17$     ;CHECK IF POSITION CORRECT
3274
3275  015214  012737  004066  001124  16$:    MOV     #4066,$GDDAT        ;USE 4066
3276  015222  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
3277  015230  001404                          BEQ     18$                 ;YES,INITIALIZE RK611
3278  015232  012737  067702  001322          MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
3279  015240  104003                          ERROR   3
3280  015242  016237  000000  002014  18$:    MOV     RKCS1(R2),PREREG            ;GET PREVIOUS CONTENTS
3281  015250  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3282  015256  016237  000000  001126          MOV     RKCS1(R2),$BDDAT            ;GET CURRENT VALUE
3283  015264  012737  000200  001124          MOV     #RDY,$GDDAT         ;LOAD EXPECTED CONTENTS
3284  015272  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;CHECK IF RKCS1 CORRECT
3285  015300  001407                          BEQ     19$                 ;YES, CHECK IF FINISHED
3286  015302  012737  063403  001330          MOV     #EM3,EM4N           ;LOAD ERROR MESSAGE
3287  015310  012737  066574  001332          MOV     #EM1000,EM4N+2
3288  015316  104004                          ERROR   4
3289  015320  104415                  19$:    SCOP1                       ;CHECK IF LOOP ON ERROR
3290  015322  000261                          SEC                         ;SHIFT IN ONE
3291  015324  006037  002012                  ROR     CONFG1
3292  015330  013737  002012  002010          MOV     CONFG1,CONFIG   ;MAKE SURE CONTROLLER CLEAR AMP
3293  015336  042737  100001  002010          BIC     #CCLR!GO,CONFIG ;   GO DO NOT SET
3294  015344  005301                          DEC     R1                  ;CHECK IF FINISHED
3295  015346  001402                          BEQ     TST20               ;;YES, GO ON TO NEXT TEST
3296  015350  000137  014536                  JMP     1$
3297
3298                          ;;*****************************************************************
```

```
                                        ;*TEST 20       REGISTER INTERACTION USING SPARE REG
                                        ;*
                                        ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
                                        ;*      WRITE THE WORD COUNT REGISTER WITH 0.
                                        ;*
                                        ;*      WRITE THE SPARE REGISTER WITH 177777 AND MAKE SURE
                                        ;*      NO INTERACTION TAKES PLACE.
                                        ;*
                                        ;**********************************************************************
015354  000004                          TST20:  SCOPE
015356  012737  000764  001200                  MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
015364  013702  001270                          MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
015370  012737  177777  002010                  MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
015376  012737  063757  001320                  MOV     #EM9,EM3N       ;LOAD ERROR MESSAGE
015404  012762  100000  000000                  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
015412  005062  000002                          CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
015416  012762  177777  000022                  MOV     #177777,RKSPAR(R2) ;WRITE RKSPAR WITH 177777
015424  016237  000000  001126                  MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
015432  022737  000200  001126                  CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
015440  001407                                  BEQ     1$              ;YES,CONTINUE
015442  012737  000200  001124                  MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
015450  012737  067314  001322                  MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
015456  104003                                  ERROR   3
015460  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
015466  001406                                  BEQ     2$              ;CHECK IF ZERO
015470  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
015474  012737  067365  001322                  MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
015502  104003                                  ERROR   3
015504  016237  000002  001126  2$:     MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
015512  001406                                  BEQ     3$              ;CHECK IF ZERO
015514  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
015520  012737  067335  001322                  MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
015526  104003                                  ERROR   3
015530  016237  000006  001126  3$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG
015536  001406                                  BEQ     4$              ;CHECK IF ZERO
015540  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
015544  012737  067412  001322                  MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
015552  104003                                  ERROR   3
015554  016237  000016  001126  4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
015562  001406                                  BEQ     5$              ;CHECK IF ZERO
015564  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
015570  012737  067556  001322                  MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
015576  104003                                  ERROR   3
015600  016237  000010  001126  5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
015606  022737  000100  001126                  CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
015614  001407                                  BEQ     6$              ;YES,CONTINUE
015616  012737  000100  001124                  MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
015624  012737  067450  001322                  MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
015632  104003                                  ERROR   3
015634  016237  000012  001126  6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
015642  001406                                  BEQ     7$              ;CHECK IF ZERO
015644  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
015650  012737  067471  001322                  MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
015656  104003                                  ERROR   3
015660  016237  000014  001126  7$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
015666  001406                                  BEQ     8$              ;CHECK IF ZERO
```

```
3355  015670  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3356  015674  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
3357  015702  104003                          ERROR   3
3358  015704  016237  000020  001126  8$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
3359  015712  001406                          BEQ     10$             ;CHECK IF ZERO
3360  015714  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
3361  015720  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
3362  015726  104003                          ERROR   3
3363  015730  016237  000026  001126  10$:    MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
3364  015736  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
3365  015744  032737  020000  001126          BIT     #ECCW,$BDDAT
3366  015752  001403                          BEQ     11$
3367  015754  052737  020000  001124          BIS     #ECCW,$GDDAT
3368  015762  023737  001124  001126  11$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
3369  015770  001404                          BEQ     12$             ;YES,CONTINUE TEST
3370  015772  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
3371  016000  104003                          ERROR   3
3372  016002  016237  000032  001126  12$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3373  016010  001406                          BEQ     13$             ;CHECK IF ZERO
3374  016012  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3375  016016  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
3376  016024  104003                          ERROR   3
3377  016026  016237  000030  001126  13$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
3378  016034  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
3379  016042  001407                          BEQ     TST21           ;;YES,GO TO NEXT TEST
3380  016044  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
3381  016052  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESAGE
3382  016060  104003                          ERROR   3
3383
3384                                  ;;*****************************************************************
3385                                  ;*TEST 21       REGISTER INTERACTION USING WORD COUNT (PART 1)
3386                                  ;*
3387                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
3388                                  ;*      WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
3389                                  ;*      AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
3390                                  ;*      PLACE.
3391                                  ;*
3392                                  ;*      000000  000010  000200  004000  100000
3393                                  ;*      000001  000020  000400  010000
3394                                  ;*      000002  000040  001000  020000
3395                                  ;*      000004  000100  002000  040000
3396                                  ;*
3397                                  ;*****************************************************************
3398  016062  000004                  TST21:  SCOPE
3399  016064  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
3400  016072  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
3401  016076  012737  000001  002010          MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
3402  016104  012737  063663  001320          MOV     #EM7,EM3N       ;LOAD ERROR MESSAGE
3403  016112  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
3404  016120  012737  016126  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
3405                                                                  ;  SUBTEST LOOP
3406
3407  016126                          1$:
3408  016126  013762  002010  000002          MOV     CONFIG,RKWC(R2) ;WRITE RKWC
3409  016134  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE RKWC
3410  016142  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
```

```
3411  016150  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKWC CORRECT
3412  016156  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
3413  016160  012737  066612  001322          MOV     #EM1001,EM3N+2  ;LOAD ERROR MESSAGE
3414  016166  104003                          ERROR   3
3415  016170                          2$:
3416  016170  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
3417  016176  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
3418  016204  001407                          BEQ     3$              ;YES, CONTINUE
3419  016206  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
3420  016214  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
3421  016222  104003                          ERROR   3
3422  016224                          3$:
3423  016224  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
3424  016232  001406                          BEQ     4$              ;CHECK IF ZERO
3425  016234  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3426  016240  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
3427  016246  104003                          ERROR   3
3428  016250                          4$:
3429  016250  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
3430  016256  001406                          BEQ     6$              ;CHECK IF ZERO
3431  016260  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3432  016264  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
3433  016272  104003                          ERROR   3
3434  016274                          6$:
3435  016274  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
3436  016302  001406                          BEQ     7$              ;CHECK IF ZERO
3437  016304  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3438  016310  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
3439  016316  104003                          ERROR   3
3440  016320                          7$:
3441  016320  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
3442  016326  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
3443  016334  001407                          BEQ     8$              ;YES,CONTINUE
3444  016336  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
3445  016344  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
3446  016352  104003                          ERROR   3.
3447  016354                          8$:
3448  016354  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
3449  016362  001406                          BEQ     9$              ;CHECK IF ZERO
3450  016364  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3451  016370  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
3452  016376  104003                          ERROR   3
3453  016400  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
3454  016406  001406                          BEQ     10$             ;CHECK IF ZERO
3455  016410  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS.
3456  016414  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
3457  016422  104003                          ERROR   3
3458  016424                          10$:
3459  016424  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
3460  016432  001406                          BEQ     12$             ;CHECK IF ZERO
3461  016434  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3462  016440  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
3463  016446  104003                          ERROR   3
3464  016450                          12$:
3465  016450  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
3466  016456  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
```

```
3467   016464  032737  020000  001126          BIT     #ECCW,$BDDAT
3468   016472  001403                          BEQ     13$
3469   016474  052737  020000  001124          BIS     #ECCW,$GDDAT
3470   016502  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
3471   016510  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
3472   016512  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
3473   016520  104003                          ERROR   3
3474   016522                          14$:
3475   016522  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3476   016530  001406                          BEQ     15$             ;CHECK IF ZERO
3477   016532  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3478   016536  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
3479   016544  104003                          ERROR   3
3480   016546  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3481   016554  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
3482   016562  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
3483   016570  001404                          BEQ     18$             ;YES,INITIALIZE RK611
3484   016572  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
3485   016600  104003                          ERROR   3
3486   016602  016237  000002  002014  18$:    MOV     RKWC(R2),PREREG ;GET PREVIOUS CONTENTS
3487   016610  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3488   016616  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;GET CURRENT VALUE
3489   016624  023737  002010  001126          CMP     CONFIG,$BDDAT   ;CHECK IF WORD COUNT NOT CHANGE
3490                                                                  ;  BY CONTROLLER CLEAR
3491   016632  001412                          BEQ     19$             ;YES, CHECK IF FINISHED
3492   016634  013737  002010  001124          MOV     CONFIG,$GDDAT   ;LOAD EXPECTED DATA
3493   016642  012737  063403  001310          MOV     #EM3,EM2N       ;LOAD ERROR MESSAGE
3494   016650  012737  067335  001312          MOV     #EM1018,EM2N+2
3495   016656  104002                          ERROR   2
3496   016660  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
3497   016662  000241                          CLC                     ;SHIFT IN ZERO
3498   016664  006137  002010                  ROL     CONFIG
3499   016670  005301                          DEC     R1              ;CHECK IF FINISHED
3500   016672  001402                          BEQ     TST22           ;;YES, GO ON TO NEXT TEST
3501   016674  000137  016126                  JMP     1$
3502
3503                                  ;;*******************  ***********************************************
3504                                  ;*TEST 22       REGISTER INTERACTION USING WORD COUNT (PART 2)
3505                                  ;*
3506                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
3507                                  ;*      WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
3508                                  ;*      AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
3509                                  ;*      PLACE.
3510                                  ;*
3511                                  ;*      177777  177767  177577  173777  077777
3512                                  ;*      177776  177757  177377  167777
3513                                  ;*      177775  177737  176777  157777
3514                                  ;*      177773  177677  175777  137777
3515                                  ;*
3516                                  ;;*****************************************************************
3517   016700  000004                  TST22:  SCOPE
3518   016702  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
3519   016710  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
3520   016714  012737  177776  002010          MOV     #177776,CONFIG  ;LOAD INITIAL CONFIGURATION
3521   016722  012737  063663  001320          MOV     #EM7,EM3N       ;LOAD ERROR MESSAGE
3522   016730  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
```

```
3523  016736  012737  016744  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
3524                                                                  ;  SUBTEST LOOP
3525
3526  016744                          1$:
3527  016744  013762  002010  000002          MOV     CONFIG,RKWC(R2) ;WRITE RKWC
3528  016752  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE RKWC
3529  016760  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
3530  016766  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKWC CORRECT
3531  016774  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
3532  016776  012737  066612  001322          MOV     #EM1001,EM3N+2  ;LOAD ERROR MESSAGE
3533  017004  104003                          ERROR   3
3534  017006                          2$:
3535  017006  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
3536  017014  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
3537  017022  001407                          BEQ     3$              ;YES, CONTINUE
3538  017024  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
3539  017032  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
3540  017040  104003                          ERROR   3
3541  017042                          3$:
3542  017042  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
3543  017050  001406                          BEQ     4$              ;CHECK IF ZERO
3544  017052  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3545  017056  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
3546  017064  104003                          ERROR   3
3547  017066                          4$:
3548  017066  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
3549  017074  001406                          BEQ     6$              ;CHECK IF ZERO
3550  017076  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3551  017102  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
3552  017110  104003                          ERROR   3
3553  017112                          6$:
3554  017112  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
3555  017120  001406                          BEQ     7$              ;CHECK IF ZERO
3556  017122  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3557  017126  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
3558  017134  104003                          ERROR   3
3559  017136                          7$:
3560  017136  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
3561  017144  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
3562  017152  001407                          BEQ     8$              ;YES,CONTINUE
3563  017154  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
3564  017162  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
3565  017170  104003                          ERROR   3
3566  017172                          8$:
3567  017172  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
3568  017200  001406                          BEQ     9$              ;CHECK IF ZERO
3569  017202  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3570  017206  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
3571  017214  104003                          ERROR   3
3572  017216  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
3573  017224  001406                          BEQ     10$             ;CHECK IF ZERO
3574  017226  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3575  017232  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
3576  017240  104003                          ERROR   3
3577  017242                          10$:
3578  017242  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
```

```
3579  017250  001406                        BEQ     12$                 ;CHECK IF ZERO
3580  017252  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3581  017256  012737  067624  001322        MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
3582  017264  104003                        ERROR   3
3583  017266                          12$:
3584  017266  016237  000026  001126        MOV     RKMR1(R2),$BDDAT    ;STORE MAINTENANCE REG.1
3585  017274  012737  002000  001124        MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
3586  017302  032737  020000  001126        BIT     #ECCW,$BDDAT
3587  017310  001403                        BEQ     13$
3588  017312  052737  020000  001124        BIS     #ECCW,$GDDAT
3589  017320  023737  001124  001126  13$:  CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
3590  017326  001404                        BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
3591  017330  012737  067651  001322        MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
3592  017336  104003                        ERROR   3
3593  017340                          14$:
3594  017340  016237  000032  001126        MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
3595  017346  001406                        BEQ     15$                 ;CHECK IF ZERO
3596  017350  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
3597  017354  012737  067727  001322        MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
3598  017362  104003                        ERROR   3
3599  017364  016237  000030  001126  15$:  MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
3600  017372  012737  004066  001124  16$:  MOV     #4066,$GDDAT        ;USE 4066
3601  017400  023737  001124  001126  17$:  CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
3602  017406  001404                        BEQ     18$                 ;YES,INITIALIZE RK611
3603  017410  012737  067702  001322        MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
3604  017416  104003                        ERROR   3
3605  017420  016237  000002  002014  18$:  MOV     RKWC(R2),PREREG     ;GET PREVIOUS CONTENTS
3606  017426  012762  100000  000000        MOV     #CCLR,RKCS1(R2)     ;CLEAR RK611 CONTROLLER
3607  017434  016237  000002  001126        MOV     RKWC(R2),$BDDAT     ;GET CURRENT VALUE
3608  017442  023737  002010  001126        CMP     CONFIG,$BDDAT       ;CHECK IF WORD COUNT NOT CHANGE
3609                                                                    ;  BY CONTROLLER CLEAR
3610  017450  001412                        BEQ     19$                 ;YES, CHECK IF FINISHED
3611  017452  013737  002010  001124        MOV     CONFIG,$GDDAT       ;LOAD EXPECTED DATA
3612  017460  012737  063403  001310        MOV     #EM3,EM2N           ;LOAD ERROR MESSAGE
3613  017466  012737  067335  001312        MOV     #EM1018,EM2N+2
3614  017474  104002                        ERROR   2
3615  017476  104415                  19$:  SCOP1                       ;CHECK IF LOOP ON ERROR
3616  017500  000261                        SEC                         ;SHIFT IN ONE
3617  017502  006137  002010                ROL     CONFIG
3618  017506  005301                        DEC     R1                  ;CHECK IF FINISHED
3619  017510  001402                        BEQ     TST23               ;;YES, GO ON TO NEXT TEST
3620  017512  000137  016744                JMP     1$
3621
3622                                  ;;*****************************************************************
3623                                  ;*TEST 23       REGISTER INTERACTION USING WORD COUNT (PART 3)
3624                                  ;*
3625                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
3626                                  ;*      WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
3627                                  ;*      AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
3628                                  ;*      PLACE.
3629                                  ;*
3630                                  ;*      000001  000037  000777  017777  000000
3631                                  ;*      000003  000077  001777  037777
3632                                  ;*      000007  000177  003777  077777
3633                                  ;*      000017  000377  007777  177777
3634                                  ;*
```

```
                        ;;*****************************************************************
                        ;ST23:  SCOPE
017516  000004                
017520  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
017526  012701  000021                    MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
017532  005037  002010                    CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
017536  012737  063663  001320            MOV     #EM7,EM3N       ;LOAD ERROR MESSAGE
017544  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
017552  012737  017560  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;   SUBTEST LOOP
                        
017560                  1$:
017560  013762  002010  000002            MOV     CONFIG,RKWC(R2) ;WRITE RKWC
017566  016237  000002  001126            MOV     RKWC(R2),$BDDAT ;STORE RKWC
017574  013737  002010  001124            MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
017602  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKWC CORRECT
017610  001404                            BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
017612  012737  066612  001322            MOV     #EM1001,EM3N+2  ;LOAD ERROR MESSAGE
017620  104003                            ERROR   3
017622                  2$:
017622  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
017630  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
017636  001407                            BEQ     3$              ;YES, CONTINUE
017640  012737  000100  001124            MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
017646  012737  067314  001322            MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
017654  104003                            ERROR   3
017656                  3$:
017656  016237  000004  001126            MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
017664  001406                            BEQ     4$              ;CHECK IF ZERO
017666  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
017672  012737  067365  001322            MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
017700  104003                            ERROR   3
017702                  4$:
017702  016237  000006  001126            MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
017710  001406                            BEQ     6$              ;CHECK IF ZERO
017712  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
017716  012737  067412  001322            MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
017724  104003                            ERROR   3
017726                  6$:
017726  016237  000016  001126            MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
017734  001406                            BEQ     7$              ;CHECK IF ZERO
017736  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
017742  012737  067556  001322            MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
017750  104003                            ERROR   3
017752                  7$:
017752  016237  000010  001126            MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
017760  022737  000010  001126            CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
017766  001407                            BEQ     8$              ;YES,CONTINUE
017770  012737  000100  001124            MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
017776  012737  067450  001322            MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
020004  104003                            ERROR   3
020006                  8$:
020006  016237  000012  001126            MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
020014  001406                            BEQ     9$              ;CHECK IF ZERO
020016  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
020022  012737  067471  001322            MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
020030  104003                            ERROR   3
```

```
3691  020032  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
3692  020040  001406                          BEQ     10$             ;CHECK IF ZERO
3693  020042  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3694  020046  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
3695  020054  104003                          ERROR   3
3696  020056                          10$:
3697  020056  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
3698  020064  001406                          BEQ     12$             ;CHECK IF ZERO
3699  020066  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3700  020072  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
3701  020100  104003                          ERROR   3
3702  020102                          12$:
3703  020102  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
3704  020110  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
3705  020116  032737  020000  001126          BIT     #ECCW,$BDDAT
3706  020124  001403                          BEQ     13$
3707  020126  052737  020000  001124          BIS     #ECCW,$GDDAT
3708  020134  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
3709  020142  001404                          BEQ     14$             ;YES, ISSUE CONTROLLER CLEAR
3710  020144  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
3711  020152  104003                          ERROR   3
3712  020154                          14$:
3713  020154  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3714  020162  001406                          BEQ     15$             ;CHECK IF ZERO
3715  020164  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3716  020170  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
3717  020176  104003                          ERROR   3
3718  020200  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3719  020206  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
3720  020214  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
3721  020222  001404                          BEQ     18$             ;YES, INITIALIZE RK611
3722  020224  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
3723  020232  104003                          ERROR   3
3724  020234  016237  000002  002014  18$:    MOV     RKWC(R2),PREREG ;GET PREVIOUS CONTENTS
3725  020242  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3726  020250  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;GET CURRENT VALUE
3727  020256  023737  002010  001126          CMP     CONFIG,$BDDAT   ;CHECK IF WORD COUNT NOT CHANGE
3728                                                                  ;  BY CONTROLLER CLEAR
3729  020264  001412                          BEQ     19$             ;YES, CHECK IF FINISHED
3730  020266  013737  002010  001124          MOV     CONFIG,$GDDAT   ;LOAD EXPECTED DATA
3731  020274  012737  063403  001310          MOV     #EM3,EM2N       ;LOAD ERROR MESSAGE
3732  020302  012737  067335  001312          MOV     #EM1018,EM2N+2
3733  020310  104002                          ERROR   2
3734  020312  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
3735  020314  000261                          SEC                     ;SHIFT IN ONE
3736  020316  006137  002010                  ROL     CONFIG
3737  020322  005301                          DEC     R1              ;CHECK IF FINISHED
3738  020324  001402                          BEQ     TST24           ;;YES, GO ON TO NEXT TEST
3739  020326  000137  017560                  JMP     1$
3740
3741                                  ;;*****************************************************************
3742                                  ;*TEST 24        REGISTER INTERACTION USING WORD COUNT (PART 4)
3743                                  ;*
3744                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
3745                                  ;*      WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGJRATIONS
3746                                  ;*      AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
```

```
                                        ;*      PLACE.
                                        ;*
                                        ;*      100000  174000  177600  177770  000000
                                        ;*      140000  176000  177700  177774
                                        ;*      160000  177000  177740  177776
                                        ;*      170000  177400  177760  177777
                                        ;*
                                        ;*****************************************************************
020332  000004                        TST24:  SCOPE
020334  012737  000764  001200                MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
020342  012701  000021                        MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
020346  005037  002010                        CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
020352  012737  063663  001320                MOV     #EM7,EM3N       ;LOAD ERROR MESSAGE
020360  012762  100000  000000                MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
020366  012737  020374  001110                MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                      ;  SUBTEST LOOP
020374                                  1$:
020374  013762  002010  000002                MOV     CONFIG,RKWC(R2) ;WRITE RKWC
020402  016237  000002  001126                MOV     RKWC(R2),$BDDAT ;STORE RKWC
020410  013737  002010  001124                MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
020416  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;CHECK IF RKWC CORRECT
020424  001404                              BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
020426  012737  066612  001322                MOV     #EM1001,EM3N+2  ;LOAD ERROR MESSAGE
020434  104003                              ERROR   3
020436                                  2$:
020436  016237  000000  001126                MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
020444  022737  000200  001126                CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
020452  001407                              BEQ     3$              ;YES, CONTINUE
020454  012737  000100  001124                MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
020462  012737  067314  001322                MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
020470  104003                              ERROR   3
020472                                  3$:
020472  016237  000004  001126                MOV     RkBA(R2),$BDDAT ;STORE BUS AND REG
020500  001406                              BEQ     4$              ;CHECK IF ZERO
020502  005037  001124                        CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
020506  012737  067365  001322                MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
020514  104003                              ERROR   3
020516                                  4$:
020516  016237  000006  001126                MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
020524  001406                              BEQ     6$              ;CHECK IF ZERO
020526  005037  001124                        CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
020532  012737  067412  001322                MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
020540  104003                              ERROR   3
020542                                  6$:
020542  016237  000016  001126                MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY OFFSET REG.
020550  001406                              BEQ     7$              ;CHECK IF ZERO
020552  005037  001124                        CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
020556  012737  067556  001322                MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
020564  104003                              ERROR   3
020566                                  7$:
020566  016237  000010  001126                MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
020574  022737  000100  001126                CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
020602  001407                              BEQ     8$              ;YES,CONTINUE
020604  012737  000100  001124                MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
020612  012737  067450  001322                MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
```

```
3803  020620  104003                          ERROR   3
3804  020622                          8$:
3805  020622  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
3806  020630  001406                          BEQ     9$              ;CHECK IF ZERO
3807  020632  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3808  020636  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
3809  020644  104003                          ERROR   3
3810  020646  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
3811  020654  001406                          BEQ     10$             ;CHECK IF ZERO
3812  020656  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3813  020662  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
3814  020670  104003                          ERROR   3
3815  020672                          10$:
3816  020672  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
3817  020700  001406                          BEQ     12$             ;CHECK IF ZERO
3818  020702  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3819  020706  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
3820  020714  104003                          ERROR   3
3821  020716                          12$:
3822  020716  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
3823  020724  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
3824  020732  032737  020000  001126          BIT     #ECCW,$BDDAT
3825  020740  001403                          BEQ     13$
3826  020742  052737  020000  001124          BIS     #ECCW,$GDDAT
3827  020750  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
3828  020756  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
3829  020760  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
3830  020766  104003                          ERROR   3
3831  020770                          14$:
3832  020770  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3833  020776  001406                          BEQ     15$             ;CHECK IF ZERO
3834  021000  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
3835  021004  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
3836  021012  104003                          ERROR   3
3837  021014  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3838  021022  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
3839  021030  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
3840  021036  001404                          BEQ     18$             ;YES,INITIALIZE RK611
3841  021040  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
3842  021046  104003                          ERROR   3
3843  021050  016237  000002  002014  18$:    MOV     RKWC(R2),PREREG ;GET PREVIOUS CONTENTS
3844  021056  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR Rk611 CONTROLLER
3845  021064  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;GET CURRENT VALUE
3846  021072  023737  002010  001126          CMP     CONFIG,$BDDAT   ;CHECK IF WORD COUNT NOT CHANGE
3847                                                                  ;  BY CONTROLLER CLEAR
3848  021100  001412                          BEQ     19$             ;YES, CHECK IF FINISHED
3849  021102  013737  002010  001124          MOV     CONFIG,$GDDAT   ;LOAD EXPECTED DATA
3850  021110  012737  063403  001310          MOV     #EM3,EM2N       ;LOAD ERROR MESSAGE
3851  021116  012737  067335  001312          MOV     #EM1018,EM2N+2
3852  021124  104002                          ERROR   2
3853  021126  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
3854  021130  000261                          SEC                     ;SHIFT IN ONE
3855  021132  006037  002010                  ROR     CONFIG
3856  021136  005301                          DEC     R1              ;CHECK IF FINISHED
3857  021140  001402                          BEQ     TST25           ;;YES, GO ON TO NEXT TEST
3858  021142  000137  020374                  JMP     1$
```

```
                                        ;;*****************************************************************
                                        ;*TEST 25       REGISTER INTERACTION USING DISK ADDRESS (PART 1)
                                        ;*
                                        ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
                                        ;*      WRITE THE WORD COUNT REGISTER WITH 0.
                                        ;*
                                        ;*      WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
                                        ;*      THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
                                        ;*      SURE NO INTERACTION TAKES PLACE.
                                        ;*
                                        ;*      000000  000010  000200  004000  100000
                                        ;*      000001  000020  000400  010000
                                        ;*      000002  000040  001000  020000
                                        ;*      000004  000100  002000  040000
                                        ;*
                                        ;*****************************************************************
021146  000004                          TST25:  SCOPE
021150  012737  000764  001200                  MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
021156  012701  000021                          MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
021162  012737  000001  002010                  MOV     #^O00001,CONFIG ;LOAD INITIAL CONFIGURATION
021170  012737  063722  001320                  MOV     #EM8,EM3N       ;LOAD ERROR MESSAGE
021176  012762  100000  000000                  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
021204  012737  021212  001110                  MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                        ;  SUBTEST LOOP
021212                                  1$:
021212  005062  000002                          CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
021216  013762  002010  000006                  MOV     CONFIG,RKDA(R2) ;WRITE RKDA
021224  016237  000006  001126                  MOV     RKDA(R2),$BDDAT ;STORE RKDA
021232  013737  002010  001124                  MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
021240  042737  174340  001124                  BIC     #174340,$GDDAT  ;INITIALIZE READ ONLY BITS
021246  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;CHECK IF RKDA CORRECT
021254  001404                                  BEQ     2$              ;YES, TEST IF ANY OTHER REG MODIFIED
021256  012737  066661  001322                  MOV     #EM1003,EM3N+2  ;LOAD ERROR MESSAGE
021264  104003                                  ERROR   3
021266                                  2$:
021266  016237  000000  001126                  MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
021274  022737  000200  001126                  CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
021302  001407                                  BEQ     3$              ;YES, CONTINUE
021304  012737  000100  001124                  MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
021312  012737  067314  001322                  MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
021320  104003                                  ERROR   3
021322                                  3$:
021322  016237  000004  001126                  MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
021330  001406                                  BEQ     4$              ;CHECK IF ZERO
021332  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
021336  012737  067365  001322                  MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
021344  104003                                  ERROR   3
021346                                  4$:
021346  016237  000002  001126                  MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
021354  001406                                  BEQ     5$              ;CHECK IF ZERO
021356  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
021362  012737  067335  001322                  MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
021370  104003                                  ERROR   3
021372                                  5$:
```

```
3915  021372  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
3916  021400  001406                          BEQ     7$                ;CHECK IF ZERO
3917  021402  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3918  021406  012737  067556  001322          MOV     #EM1024,EM3N+2    ;LOAD ERROR MESSAGE
3919  021414  104003                          ERROR   3
3920  021416                          7$:
3921  021416  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
3922  021424  022737  000100  001126          CMP     #IR,$BDDAT        ;CHECK IF CS2 CORRECT
3923  021432  001407                          BEQ     8$                ;YES,CONTINUE
3924  021434  012737  000100  001124          MOV     #IR,$GDDAT        ;LOAD EXPECTED CONTENTS
3925  021442  012737  067450  001322          MOV     #EM1021,EM3N+2    ;LOAD ERROR MESSAGE
3926  021450  104003                          ERROR   3
3927  021452                          8$:
3928  021452  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
3929  021460  001406                          BEQ     9$                ;CHECK IF ZERO
3930  021462  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3931  021466  012737  067471  001322          MOV     #EM1022,EM3N+2    ;LOAD ERROR MESSAGE
3932  021474  104003                          ERROR   3
3933  021476  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
3934  021504  001406                          BEQ     10$               ;CHECK IF ZERO
3935  021506  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3936  021512  012737  067527  001322          MOV     #EM1023,EM3N+2    ;LOAD ERROR MESSAGE
3937  021520  104003                          ERROR   3
3938  021522                          10$:
3939  021522  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
3940  021530  001406                          BEQ     12$               ;CHECK IF ZERO
3941  021532  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3942  021536  012737  067624  001322          MOV     #EM1025,EM3N+2    ;LOAD ERROR MESSAGE
3943  021544  104003                          ERROR   3
3944  021546                          12$:
3945  021546  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
3946  021554  012737  002000  001124          MOV     #MEWD,$GDDAT      ;LOAD EXPECTED MR1
3947  021562  032737  020000  001126          BIT     #ECCW,$BDDAT
3948  021570  001403                          BEQ     13$
3949  021572  052737  020000  001124          BIS     #ECCW,$GDDAT
3950  021600  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT     ;CHECK IF MR1 CORRECT
3951  021606  001404                          BEQ     14$               ;YES,ISSUE CONTROLLER CLEAR
3952  021610  012737  067651  001322          MOV     #EM1026,EM3N+2    ;LOAD ERROR MESSAGE
3953  021616  104003                          ERROR   3
3954  021620                          14$:
3955  021620  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3956  021626  001406                          BEQ     15$               ;CHECK IF ZERO
3957  021630  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3958  021634  012737  067727  001322          MOV     #EM1030,EM3N+2    ;LOAD ERROR MESSAGE
3959  021642  104003                          ERROR   3
3960  021644  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3961  021652  012737  004066  001124  16$:    MOV     #4066,$GDDAT      ;USE 4066
3962  021660  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT     ;CHECK IF ECC POSITION CORRECT
3963  021666  001404                          BEQ     18$               ;YES,INITIALIZE RK611
3964  021670  012737  067702  001322          MOV     #EM1029,EM3N+2    ;LOAD ERROR MESSAGE
3965  021676  104003                          ERROR   3
3966  021700  016237  000006  002014  18$:    MOV     RKDA(R2),PREREG ;GET PREVIOUS CONTENTS
3967  021706  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3968  021714  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;GET CURRENT VALUE
3969  021722  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
3970  021726  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;CHECK IF RKDA CORRECT
```

```
 3971  021734  001407                          BEQ     19$                 ;YES, CHECK IF FINISHED
 3972  021736  012737  063403  001330          MOV     #EM3,EM4N           ;LOAD ERROR MESSAGE
 3973  021744  012737  066661  001332          MOV     #EM1003,EM4N+2
 3974  021752  104004                          ERROR   4
 3975  021754  104415                  19$:    SCOP1                       ;CHECK IF LOOP ON ERROR
 3976  021756  000241                          CLC                         ;SHIFT IN ZERO
 3977  021760  006137  002010                  ROL     CONFIG
 3978  021764  005301                          DEC     R1                  ;CHECK IF FINISHED
 3979  021766  001402                          BEQ     TST26               ;;YES, GO ON TO NEXT TEST
 3980  021770  000137  021212                  JMP     1$
 3981
 3982                                  ;*****************************************************************
 3983                                  ;TEST 26        REGISTER INTERACTION USING DISK ADDRESS (PART 2)
 3984                                  ;*
 3985                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
 3986                                  ;*      WRITE THE WORD COUNT REGISTER WITH 0.
 3987                                  ;*
 3988                                  ;*      WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
 3989                                  ;*      THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
 3990                                  ;*      SURE NO INTERACTION TAKES PLACE.
 3991                                  ;*
 3992                                  ;*      177777  177767  177577  173777  077777
 3993                                  ;*      177776  177757  177377  167777
 3994                                  ;*      177775  177737  176777  157777
 3995                                  ;*      177773  177677  175777  137777
 3996                                  ;*
 3997                                  ;*****************************************************************
 3998  021774  000004                  TST26:  SCOPE
 3999  021776  012737  000764  001200          MOV     #500.,STIMES        ;;DO 500. ITERATIONS
 4000  022004  012701  000021                  MOV     #17.,R1             ;LOAD NUMBER OF PATTERNS
 4001  022010  012737  177776  002010          MOV     #177776,CONFIG      ;LOAD INITIAL CONFIGURATION
 4002  022016  012737  063722  001320          MOV     #EM8,EM3N           ;LOAD ERROR MESSAGE
 4003  022024  012762  100000  000000          MOV     #CCLR,RKCS1(R2)     ;CLEAR RK611 WITH CONTROLLER CLEAR
 4004  022032  012737  022040  001110          MOV     #1$,$LPERR          ;LOAD LOOP ON ERROR LOCATION FOR
 4005                                                                      ;  SUBTEST LOOP
 4006
 4007  022040                          1$:
 4008  022040  005062  000002                  CLR     RKWC(R2)            ;CLEHR WORD COUNT REG.
 4009  022044  013762  002010  000006          MOV     CONFIG,RKDA(R2)     ;WRITE RKDA
 4010  022052  016237  000006  001126          MOV     RKDA(R2),$BDDAT     ;STORE RKDA
 4011  022060  013737  002010  001124          MOV     CONFIG,$GDDAT       ;PREPARE EXPECTED RESULTS
 4012  022066  042737  174340  001124          BIC     #174340,$GDDAT      ;INITIALIZE READ ONLY BITS
 4013  022074  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;CHECK IF RKDA CORRECT
 4014  022102  001404                          BEQ     2$                  ;YES,TEST IF ANY OTHER REG MODIFIED
 4015  022104  012737  066661  001322          MOV     #EM1003,EM3N+2      ;LOAD ERROR MESSAGE
 4016  022112  104003                          ERROR   3
 4017  022114                          2$:
 4018  022114  016237  000000  001126          MOV     RKCS1(R2),$BDDAT    ;STORE COMMAND AND STATUS REG. 1
 4019  022122  022737  000200  001126          CMP     #RDY,$BDDAT         ;CHECK IF CS1 CORRECT
 4020  022130  001407                          BEQ     3$                  ;YES, CONTINUE
 4021  022132  012737  000100  001124          MOV     #IR,$GDDAT          ;LOAD EXPECTED RESULTS
 4022  022140  012737  067314  001322          MOV     #EM1017,EM3N+2      ;LOAD ERROR MESSAGE
 4023  022146  104003                          ERROR   3
 4024  022150                          3$:
 4025  022150  016237  000004  001126          MOV     RKBA(R2),$BDDAT     ;STORE BUS AND REG
 4026  022156  001406                          BEQ     4$                  ;CHECK IF ZERO
```

```
4027  022160  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4028  022164  012737  067365  001322      MOV     #EM1019,EM3N+2      ;LOAD ERROR MESSAGE
4029  022172  104003                      ERROR   3
4030  022174                        4$:
4031  022174  016237  000002  001126      MOV     RKWC(R2),$BDDAT     ;STORE WORD COUNT REG
4032  022202  001406                      BEQ     5$                  ;CHECK IF ZERO
4033  022204  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4034  022210  012737  067335  001322      MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
4035  022216  104003                      ERROR   3
4036  022220                        5$:
4037  022220  016237  000016  001126      MOV     RKASOF(R2),$BDDAT   ;STORE ATTENTION SUMMARY/OFFSET REG.
4038  022226  001406                      BEQ     7$                  ;CHECK IF ZERO
4039  022230  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4040  022234  012737  067556  001322      MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
4041  022242  104003                      ERROR   3
4042  022244                        7$:
4043  022244  016237  000010  001126      MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
4044  022252  022737  000100  001126      CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
4045  022260  001407                      BEQ     8$                  ;YES,CONTINUE
4046  022262  012737  000100  001124      MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
4047  022270  012737  067450  001322      MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
4048  022276  104003                      ERROR   3
4049  022300                        8$:
4050  022300  016237  000012  001126      MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
4051  022306  001406                      BEQ     9$                  ;CHECK IF ZERO
4052  022310  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4053  022314  012737  067471  001322      MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
4054  022322  104003                      ERROR   3
4055  022324  016237  000014  001126  9$: MOV     RKER(R2),$BDDAT     ;STORE ERROR REG
4056  022332  001406                      BEQ     10$                 ;CHECK IF ZERO
4057  022334  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4058  022340  012737  067527  001322      MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
4059  022346  104003                      ERROR   3
4060  022350                        10$:
4061  022350  016237  000020  001126      MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG
4062  022356  001406                      BEQ     12$                 ;CHECK IF ZERO
4063  022360  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4064  022364  012737  067624  001322      MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
4065  022372  104003                      ERROR   3
4066  022374                        12$:
4067  022374  016237  000026  001126      MOV     RKMR1(R2),$BDDAT    ;STORE MAINTENANCE REG.1
4068  022402  012737  002000  001124      MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
4069  022410  032737  020000  001126      BIT     #ECCW,$BDDAT
4070  022416  001403                      BEQ     13$
4071  022420  052737  020000  001124      BIS     #ECCW,$GDDAT
4072  022426  023737  001124  001126  13$: CMP    $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
4073  022434  001404                      BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
4074  022436  012737  067651  001322      MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
4075  022444  104003                      ERROR   3
4076  022446                        14$:
4077  022446  016237  000032  001126      MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
4078  022454  001406                      BEQ     15$                 ;CHECK IF ZERO
4079  022456  005037  001124              CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4080  022462  012737  067727  001322      MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
4081  022470  104003                      ERROR   3
4082  022472  016237  000030  001126  15$: MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
```

```
4083  022500  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
4084  022506  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
4085  022514  001404                          BEQ     18$             ;YES,INITIALIZE RK611
4086  022516  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
4087  022524  104003                          ERROR   3
4088  022526  016237  000006  002014  18$:    MOV     RKDA(R2),PREREG ;GET PREVIOUS CONTENTS
4089  022534  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4090  022542  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;GET CURRENT VALUE
4091  022550  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4092  022554  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDA CORRECT
4093  022562  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4094  022564  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4095  022572  012737  066661  001332          MOV     #EM1003,EM4N+2
4096  022600  104004                          ERROR   4
4097  022602  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
4098  022604  000261                          SEC                     ;SHIFT IN ONE
4099  022606  006137  002010                  ROL     CONFIG
4100  022612  005301                          DEC     R1              ;CHECK IF FINISHED
4101  022614  001402                          BEQ     TST27           ;;YES, GO ON TO NEXT TEST
4102  022616  000137  022040                  JMP     1$
4103
4104                                  ;;*****************************************************************
4105                                  ;*TEST 27       REGISTER INTERACTION USING DISK ADDRESS (PART 3)
4106                                  ;*
4107                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4108                                  ;*      WRITE THE WORD COUNT REGISTER WITH 0.
4109                                  ;*
4110                                  ;*      WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
4111                                  ;*      THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
4112                                  ;*      SURE NO INTERACTION TAKES PLACE.
4113                                  ;*
4114                                  ;*      000001  000037  000777  017777  000000
4115                                  ;*      000003  000077  001777  037777
4116                                  ;*      000007  000177  003777  077777
4117                                  ;*      000017  000377  007777  177777
4118                                  ;*
4119                                  ;*****************************************************************
4120  022622  007004          TST27:  SCOPE
4121  022624  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
4122  022632  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
4123  022636  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
4124  022642  012737  063722  001320          MOV     #EM8,EM3N       ;LOAD ERROR MESSAGE
4125  022650  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4126  022656  012737  022664  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
4127                                                                  ;  SUBTEST LOOP
4128
4129  022664                          1$:
4130  022664  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
4131  022670  013762  002010  000006          MOV     CONFIG,RKDA(R2) ;WRITE RKDA
4132  022676  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE RKDA
4133  022704  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
4134  022712  042737  174340  001124          BIC     #174340,$GDDAT  ;INITIALIZE READ ONLY BITS
4135  022720  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDA CORRECT
4136  022726  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
4137  022730  012737  066661  001322          MOV     #EM1003,EM3N+2  ;LOAD ERROR MESSAGE
4138  022736  104003                          ERROR   3
```

```
4139  022740                                  2$:
4140  022740  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
4141  022746  022737  000200  001126            CMP     #RDY,$BDDAT      ;CHECK IF CS1 CORRECT
4142  022754  001407                            BEQ     3$               ;YES, CONTINUE
4143  022756  012737  000100  001124            MOV     #IR,$GDDAT       ;LOAD EXPECTED RESULTS
4144  022764  012737  067314  001322            MOV     #EM1017,EM3N+2   ;LOAD ERROR MESSAGE
4145  022772  104003                            ERROR   3
4146  022774                                  3$:
4147  022774  016237  000004  001126            MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
4148  023002  001406                            BEQ     4$               ;CHECK IF ZERO
4149  023004  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
4150  023010  012737  067365  001322            MOV     #EM1019,EM3N+2   ;LOAD ERROR MESSAGE
4151  023016  104003                            ERROR   3
4152  023020                                  4$:
4153  023020  016237  000002  001126            MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
4154  023026  001406                            BEQ     5$               ;CHECK IF ZERO
4155  023030  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
4156  023034  012737  067335  001322            MOV     #EM1018,EM3N+2   ;LOAD ERROR MESSAGE
4157  023042  104003                            ERROR   3
4158  023044                                  5$:
4159  023044  016237  000016  001126            MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
4160  023052  001406                            BEQ     7$               ;CHECK IF ZERO
4161  023054  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
4162  023060  012737  067556  001322            MOV     #EM1024,EM3N+2   ;LOAD ERROR MESSAGE
4163  023066  104003                            ERROR   3
4164  023070                                .7$:
4165  023070  016237  000010  001126            MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
4166  023076  022737  000100  001126            CMP     #IR,$BDDAT       ;CHECK IF CS2 CORRECT
4167  023104  001407                            BEQ     8$               ;YES,CONTINUE
4168  023106  012737  000100  001124            MOV     #IR,$GDDAT       ;LOAD EXPECTED CONTENTS
4169  023114  012737  067450  001322            MOV     #EM1021,EM3N+2   ;LOAD ERROR MESSAGE
4170  023122  104003                            ERROR   3
4171  023124                                  8$:
4172  023124  016237  000012  001126            MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
4173  023132  001406                            BEQ     9$               ;CHECK IF ZERO
4174  023134  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
4175  023140  012737  067471  001322            MOV     #EM1022,EM3N+2   ;LOAD ERROR MESSAGE
4176  023146  104003                            ERROR   3
4177  023150  016237  000014  001126  9$:       MOV     RKER(R2),$BDDAT ;STORE ERROR REG
4178  023156  001406                            BEQ     10$              ;CHECK IF ZERO
4179  023160  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
4180  023164  012737  067527  001322            MOV     #EM1023,EM3N+2   ;LOAD ERROR MESSAGE
4181  023172  104003                            ERROR   3
4182  023174                                 10$:
4183  023174  016237  000020  001126            MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER WD REG
4184  023202  001406                            BEQ     12$              ;CHECK IF ZERO
4185  023204  005037  001124                    CLR     $GDDAT           ;LOAD EXPECTED CONTENTS
4186  023210  012737  067624  001322            MOV     #EM1025,EM3N+2   ;LOAD ERROR MESSAGE
4187  023216  104003                            ERROR   3
4188  023220                                 12$:
4189  023220  016237  000026  001126            MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
4190  023226  012737  002000  001124            MOV     #MEWD,$GDDAT     ;LOAD EXPECTED MR1
4191  023234  032737  020000  001126            BIT     #ECCW,$BDDAT
4192  023242  001403                            BEQ     13$
4193  023244  052737  020000  001124            BIS     #ECCW,$GDDAT
4194  023252  023737  001124  001126  13$:      CMP     $GDDAT,$BDDAT    ;CHECK IF MR1 CORRECT
```

```
4195  023260  001404                          BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
4196  023262  012737  067651  001322          MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
4197  023270  104003                          ERROR   3
4198  023272                          14$:
4199  023272  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4200  023300  001406                          BEQ     15$                 ;CHECK IF ZERO
4201  023302  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4202  023306  012737  067727  001322          MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
4203  023314  104003                          ERROR   3
4204  023316  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4205  023324  012737  004066  001124  16$:    MOV     #4066,$GDDAT        ;USE 4066
4206  023332  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
4207  023340  001404                          BEQ     18$                 ;YES,INITIALIZE RK611
4208  023342  012737  067702  001322          MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
4209  023350  104003                          ERROR   3
4210  023352  016237  000006  002014  18$:    MOV     RKDA(R2),PREREG ;GET PREVIOUS CONTENTS
4211  023360  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4212  023366  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;GET CURRENT VALUE
4213  023374  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4214  023400  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;CHECK IF RKDA CORRECT
4215  023406  001407                          BEQ     19$                 ;YES, CHECK IF FINISHED
4216  023410  012737  063403  001330          MOV     #EM3,EM4N           ;LOAD ERROR MESSAGE
4217  023416  012737  066661  001332          MOV     #EM1003,EM4N+2
4218  023424  104004                          ERROR   4
4219  023426  104415                  19$:    SCOPE                       ;CHECK IF LOOP ON ERROR
4220  023430  000261                          SEC                         ;SHIFT IN ONE
4221  023432  006137  002010                  ROL     CONFIG
4222  023436  005301                          DEC     R1                  ;CHECK IF FINISHED
4223  023440  001402                          BEQ     TST30               ;;YES, GO ON TO NEXT TEST
4224  023442  000137  022664                  JMP     1$
4225
4226                          ;;*****************************************************************
4227                          ;*TEST 30       REGISTER INTERACTION USING DISK ADDRESS (PART 4)
4228                          ;*
4229                          ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4230                          ;*      WRITE THE WORD COUNT REGISTER WITH 0.
4231                          ;*
4232                          ;*      WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
4233                          ;*      THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
4234                          ;*      SURE NO INTERACTION TAKES PLACE.
4235                          ;*
4236                          ;*      100000  174000  177600  177770  000000
4237                          ;*      140000  176000  177700  177774
4238                          ;*      160000  177000  177740  177776
4239                          ;*      170000  177400  177760  177777
4240                          ;*
4241                          ;*****************************************************************
4242  023446  000004          TST30:  SCOPE
4243  023450  012737  000764  001200          MOV     #500.,$TIMES        ;;DO 500. ITERATIONS
4244  023456  012701  000021                  MOV     #17.,R1             ;LOAD NUMBER OF PATTERNS
4245  023462  005037  002010                  CLR     CONFIG              ;LOAD INITIAL CONFIGURATION
4246  023466  012737  063722  001320          MOV     #EM8,EM3N           ;LOAD ERROR MESSAGE
4247  023474  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4248  023502  012737  023510  001110          MOV     #1$,$LPERR          ;LOAD LOOP ON ERROR LOCATION FOR
4249                                                                      ;  SUBTEST LOOP
4250
```

```
4251  023510                                1$:
4252  023510  005062  000002                      CLR     RKWC(R2)            ;CLEAR WORD COUNT REG.
4253  023514  013762  002010  000006              MOV     CONFIG,RKDA(R2)     ;WRITE RKDA
4254  023522  016237  000006  001126              MOV     RKDA(R2),$BDDAT     ;STORE RKDA
4255  023530  013737  002010  001124              MOV     CONFIG,$GDDAT       ;PREPARE EXPECTED RESULTS
4256  023536  042737  174340  001124              BIC     #174340,$GDDAT      ;INITIALIZE READ ONLY BITS
4257  023544  023737  001124  001126              CMP     $GDDAT,$BDDAT       ;CHECK IF RKDA CORRECT
4258  023552  001404                              BEQ     2$                  ;YES,TEST IF ANY OTHER REG MODIFIED
4259  023554  012737  066661  001322              MOV     #EM1003,EM3N+2      ;LOAD ERROR MESSAGE
4260  023562  104003                              ERROR   3
4261  023564                                2$:
4262  023564  016237  000000  001126              MOV     RKCS1(R2),$BDDAT    ;STORE COMMAND AND STATUS REG. 1
4263  023572  022737  000200  001126              CMP     #RDY,$BDDAT         ;CHECK IF CS1 CORRECT
4264  023600  001407                              BEQ     3$                  ;YES, CONTINUE
4265  023602  012737  000100  001124              MOV     #IR,$GDDAT          ;LOAD EXPECTED RESULTS
4266  023610  012737  067314  001322              MOV     #EM1017,EM3N+2      ;LOAD ERROR MESSAGE
4267  023616  104003                              ERROR   3
4268  023620                                3$:
4269  023620  016237  000004  001126              MOV     RKBA(R2),$BDDAT     ;STORE BUS AND REG
4270  023626  001406                              BEQ     4$                  ;CHECK IF ZERO
4271  023630  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4272  023634  012737  067365  001322              MOV     #EM1019,EM3N+2      ;LOAD ERROR MESSAGE
4273  023642  104003                              ERROR   3
4274  023644                                4$:
4275  023644  016237  000002  001126              MOV     RKWC(R2),$BDDAT     ;STORE WORD COUNT REG
4276  023652  001406                              BEQ     5$                  ;CHECK IF ZERO
4277  023654  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4278  023660  012737  067335  001322              MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
4279  023666  104003                              ERROR   3
4280  023670                                5$:
4281  023670  016237  000016  001126              MOV     RKASOF(R2),$BDDAT   ;STORE ATTENTION SUMMARY/OFFSET REG.
4282  023676  001406                              BEQ     7$                  ;CHECK IF ZERO
4283  023700  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4284  023704  012737  067556  001322              MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
4285  023712  104003                              ERROR   3
4286  023714                                7$:
4287  023714  016237  000010  001126              MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
4288  023722  022737  000100  001126              CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
4289  023730  001407                              BEQ     8$                  ;YES,CONTINUE
4290  023732  012737  000100  001124              MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
4291  023740  012737  067450  001322              MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
4292  023746  104003                              ERROR   3
4293  023750                                8$:
4294  023750  016237  000012  001126              MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
4295  023756  001406                              BEQ     9$                  ;CHECK IF ZERO
4296  023760  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4297  023764  012737  067471  001322              MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
4298  023772  104003                              ERROR   3
4299  023774  016237  000014  001126  9$:         MOV     RKER(R2),$BDDAT     ;STORE ERROR REG
4300  024002  001406                              BEQ     10$                 ;CHECK IF ZERO
4301  024004  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
4302  024010  012737  067527  001322              MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
4303  024016  104003                              ERROR   3
4304  024020                                10$:
4305  024020  016237  000020  001126              MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG
4306  024026  001406                              BEQ     12$                 ;CHECK IF ZERO
```

```
4307  024030  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4308  024034  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
4309  024042  104003                          ERROR   3
4310  024044                          12$:
4311  024044  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
4312  024052  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
4313  024060  032737  020000  001126          BIT     #ECCW,$BDDAT
4314  024066  001403                          BEQ     13$
4315  024070  052737  020000  001124          BIS     #ECCW,$GDDAT
4316  024076  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
4317  024104  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
4318  024106  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
4319  024114  104003                          ERROR   3
4320  024116                          14$:
4321  024116  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4322  024124  001406                          BEQ     15$             ;CHECK IF ZERO
4323  024126  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4324  024132  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
4325  024140  104003                          ERROR   3
4326  024142  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4327  024150  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
4328  024156  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
4329  024164  001404                          BEQ     18$             ;YES,INITIALIZE RK611
4230  024166  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
4331  024174  104003                          ERROR   3
4332  024176  016237  000006  002014  18$:    MOV     RKDA(R2),PREREG ;GET PREVIOUS CONTENTS
4333  024204  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4334  024212  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;GET CURRENT VALUE
4335  024220  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4336  024224  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDA CORRECT
4337  024232  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4338  024234  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4339  024242  012737  066661  001332          MOV     #EM1003,EM4N+2
4340  024250  104004                          ERROR   4
4341  024252  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
4342  024254  000261                          SEC                     ;SHIFT IN ONE
4343  024256  006137  002010                  ROL     CONFIG
4344  024262  005301                          DEC     R1              ;CHECK IF FINISHED
4345  024264  001402                          BEQ     TST31           ;;YES, GO ON TO NEXT TEST
4346  024266  000137  023510                  JMP     1$
4347
4348                          ;;*****************************************************************
4349                          ;*TEST 31        REGISTER INTERACTION USING ATTN/OFFSET (PART 1)
4350                          ;*
4351                          ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4352                          ;*      WRITE THE WORD COUNT REGISTER WITH 0.
4353                          ;*
4354                          ;*      WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
4355                          ;*      WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
4356                          ;*      MAKE SURE NO INTERACTION TAKES PLACE.
4357                          ;*
4358                          ;*      000000  000010  000200  004000  100000
4359                          ;*      000001  000020  000400  010000
4360                          ;*      000002  000040  001000  020000
4361                          ;*      000004  000100  002000  040000
4362                          ;*
```

```
                                ;;*****************************************************************
                        TST31:  SCOPE
024272  000004                  
024274  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
024302  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
024306  012737  000001  002010          MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
024314  012737  064351  001320          MOV     #EM17,EM3N      ;LOAD ERROR MESSAGE
024322  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
024330  012737  024336  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
                        
024336                          1$:
024336  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
024342  013762  002010  000016          MOV     CONFIG,RKASOF(R2)       ;WRITE RKASOF
024350  016237  000016  001126          MOV     RKASOF(R2),$BDDAT       ;STORE RKASOF
024356  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
024364  042737  177400  001124          BIC     #177400,$GDDAT  ;INITIALIZE READ ONLY BITS
024372  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
024400  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
024402  012737  067001  001322          MOV     #EM1007,EM3N+2  ;LOAD ERROR MESSAGE
024410  104003                          ERROR   3
024412                          2$:
024412  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
024420  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
024426  001407                          BEQ     3$              ;YES, CONTINUE
024430  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
024436  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
024444  104003                          ERROR   3
024446                          3$:
024446  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
024454  001406                          BEQ     4$              ;CHECK IF ZERO
024456  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
024462  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
024470  104003                          ERROR   3
024472                          4$:
024472  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
024500  001406                          BEQ     5$              ;CHECK IF ZERO
024502  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
024506  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
024514  104003                          ERROR   3
024516                          5$:
024516  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
024524  001406                          BEQ     6$              ;CHECK IF ZERO
024526  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
024532  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
024540  104003                          ERROR   3
024542                          6$:
024542  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
024550  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
024556  001407                          BEQ     8$              ;YES,CONTINUE
024560  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
024566  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
024574  104003                          ERROR   3
024576                          8$:
024576  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
024604  001406                          BEQ     9$              ;CHECK IF ZERO
024606  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
```

```
4419  024612  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
4420  024620  104003                          ERROR   3
4421  024622  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
4422  024630  001406                          BEQ     10$             ;CHECK IF ZERO
4423  024632  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4424  024636  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
4425  024644  104003                          ERROR   3
4426  024646                          10$:
4427  024646  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
4428  024654  001406                          BEQ     12$             ;CHECK IF ZERO
4429  024656  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4430  024662  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
4431  024670  104003                          ERROR   3
4432  024672                          12$:
4433  024672  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
4434  024700  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
4435  024706  032737  020000  001126          BIT     #ECCW,$BDDAT
4436  024714  001403                          BEQ     13$
4437  024716  052737  020000  001124          BIS     #ECCW,$GDDAT
4438  024724  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
4439  024732  001404                          BEQ     14$             ;YES, ISSUE CONTROLLER CLEAR
4440  024734  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
4441  024742  104003                          ERROR   3
4442  024744                          14$:
4443  024744  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4444  024752  001406                          BEQ     15$             ;CHECK IF ZERO
4445  024754  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4446  024760  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
4447  024766  104003                          ERROR   3
4448  024770  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4449  024776  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
4450  025004  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
4451  025012  001404                          BEQ     18$             ;YES,INITIALIZE RK611
4452  025014  012737  067702  001322          MOV     #EM1029,FM3N+2  ;LOAD ERROR MESSAGE
4453  025022  104003                          ERROR   3
4454  025024  016237  000016  002014  18$:    MOV     RKASOF(R2),PREREG       ;GET PREVIOUS CONTENTS
4455  025032  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4456  025040  016237  000016  001126          MOV     RKASOF(R2),$BDDAT       ;GET CURRENT VALUE
4457  025046  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4458  025052  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
4459  025060  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4460  025062  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4461  025070  012737  067001  001332          MOV     #EM1007,EM4N+2
4462  025076  104004                          ERROR   4
4463  025100  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
4464  025102  000241                          CLC                     ;SHIFT IN ZERO
4465  025104  006137  002010                  ROL     CONFIG
4466  025110  005301                          DEC     R1              ;CHECK IF FINISHED
4467  025112  001402                          BEQ     TST32           ;;YES, GO ON TO NEXT TEST
4468  025114  000137  024336                  JMP     1$
4469
4470                            ;;*****************************************************************
4471                            ;*TEST 32       REGISTER INTERACTION USING ATTN/OFFSET (PART 2)
4472                            ;*
4473                            ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4474                            ;*      WRITE THE WORD COUNT REGISTER WITH 0.
```

```
                                        ;*
                                        ;*      WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
                                        ;*      WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
                                        ;*      MAKE SURE NO INTERACTION TAKES PLACE.
                                        ;*
                                        ;*      177777  177767  177577  173777  077777
                                        ;*      177776  177757  177377  167777
                                        ;*      177775  177737  176777  157777
                                        ;*      177773  177677  175777  137777
                                        ;*
                                        ;**********************************************************************
025120  000004                    T$T32:  SCOPE
025122  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
025130  012701  000021                    MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
025134  012737  177776  002010            MOV     #177776,CONFIG  ;LOAD INITIAL CONFIGURATION
025142  012737  064351  001320            MOV     #EM17,EM3N      ;LOAD ERROR MESSAGE
025150  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
025156  012737  025164  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
025164                              1$:
025164  005062  000002                    CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
025170  013762  002010  000016            MOV     CONFIG,RKASOF(R2)       ;WRITE RKASOF
025176  016237  000016  001126            MOV     RKASOF(R2),$BDDAT       ;STORE RKASOF
025204  013737  002010  001124            MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
025212  042737  177400  001124            BIC     #177400,$GDDAT  ;INITIALIZE READ ONLY BITS
025220  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
025226  001404                            BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
025230  012737  067001  001322            MOV     #EM1007,EM3N+2  ;LOAD ERROR MESSAGE
025236  104003                            ERROR   3
025240                              2$:
025240  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
025246  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
025254  001407                            BEQ     3$              ;YES, CONTINUE
025256  012737  000100  001124            MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
025264  012737  067314  001322            MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
025272  104003                            ERROR   3
025274                              3$:
025274  016237  000004  001126            MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
025302  001406                            BEQ     4$              ;CHECK IF ZERO
025304  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
025310  012737  067365  001322            MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
025316  104003                            ERROR   3
025320                              4$:
025320  016237  000002  001126            MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
025326  001406                            BEQ     5$              ;CHECK IF ZERO
025330  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
025334  012737  067335  001322            MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
025342  104003                            ERROR   3
025344                              5$:
025344  016237  000006  001126            MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
025352  001406                            BEQ     6$              ;CHECK IF ZERO
025354  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
025360  012737  067412  001322            MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
025366  104003                            ERROR   3
025370                              6$:
```

```
4531  025370  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
4532  025376  022737  000100  001126          CMP     #IR,$BDDAT       ;CHECK IF CS2 CORRECT
4533  025404  001407                          BEQ     8$               ;YES,CONTINUE
4534  025406  012737  000100  001124          MOV     #IR,$GDDAT       ;LOAD EXPECTED CONTENTS
4535  025414  012737  067450  001322          MOV     #EM1021,EM3N+2   ;LOAD ERROR MESSAGE
4536  025422  104003                          ERROR   3
4537  025424                          8$:
4538  025424  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
4539  025432  001406                          BEQ     9$              ;CHECK IF ZERO
4540  025434  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4541  025440  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
4542  025446  104003                          ERROR   3
4543  025450  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
4544  025456  001406                          BEQ     10$             ;CHECK IF ZERO
4545  025460  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4546  025464  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
4547  025472  104003                          ERROR   3
4548  025474                          10$:
4549  025474  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
4550  025502  001406                          BEQ     12$             ;CHECK IF ZERO
4551  025504  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4552  025510  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
4553  025516  104003                          ERROR   3
4554  025520                          12$:
4555  025520  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
4556  025526  012737  002000  001124          MOV     #MEWD,$GDDAT     ;LOAD EXPECTED MR1
4557  025534  032737  020000  001126          BIT     #ECCW,$BDDAT
4558  025542  001403                          BEQ     13$
4559  025544  052737  020000  001124          BIS     #ECCW,$GDDAT
4560  025552  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
4561  025560  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
4562  025562  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
4563  025570  104003                          ERROR   3
4564  025572                          14$:
4565  025572  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4566  025600  001406                          BEQ     15$             ;CHECK IF ZERO
4567  025602  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4568  025606  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
4569  025614  104003                          ERROR   3
4570  025616  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4571  025624  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
4572  025632  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
4573  025640  001404                          BEQ     18$             ;YES,INITIALIZE RK611
4574  025642  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
4575  025650  104003                          ERROR   3
4576  025652  016237  000016  002014  18$:    MOV     RKASOF(R2),PREREG       ;GET PREVIOUS CONTENTS
4577  025660  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4578  025666  016237  000016  001126          MOV     RKASOF(R2),$BDDAT       ;GET CURRENT VALUE
4579  025674  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4580  025700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
4581  025706  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4582  025710  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4583  025716  012737  067001  001332          MOV     #EM1007,EM4N+2
4584  025724  104004                          ERROR   4
4585  025726  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
4586  025730  000261                          SEC                     ;SHIFT IN ONE
```

```
4587  025732  006137  002010          ROL     CONFIG
4588  025736  005301                  DEC     R1              ;CHECK IF FINISHED
4589  025740  001402                  BEQ     TST33           ;;YES, GO ON TO NEXT TEST
4590  025742  000137  025164          JMP     1$
4591
4592                          ;;*****************************************************************
4593                          ;*TEST 33       REGISTER INTERACTION USING ATTN/OFFSET (PART 3)
4594                          ;*
4595                          ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4596                          ;*      WRITE THE WORD COUNT REGISTER WITH 0.
4597                          ;*
4598                          ;*      WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
4599                          ;*      WITH THE FOLLOWING CONFIGURATIONS. CHE^' REGISTER CONTENTS AND
4600                          ;*      MAKE SURE NO INTERACTION TAKES PLACE.
4601                          ;*
4602                          ;*      000001  000037  000777  017777  ⌴00000
4603                          ;*      000003  000077  001777  037777
4604                          ;*      000007  000177  003777  077777
4605                          ;*      000017  000377  007777  177777
4606                          ;*
4607                          ;*****************************************************************
4608  025746  000004          TST33:  SCOPE
4609  025750  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
4610  025756  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
4611  025762  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
4612  025766  012737  064351  001320          MOV     #EM17,EM3N      ;LOAD ERROR MESSAGE
4613  025774  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4614  026002  012737  026010  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
4615                                                                  ;  SUBTEST LOOP
4616
4617  026010                  1$:
4618  026010  005062  000002          CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
4619  026014  013762  002010  000016  MOV     CONFIG,RKASOF(R2)       ;WRITE RKASOF
4620  026022  016237  000016  001126  MOV     RKASOF(R2),$BDDAT       ;STORE RKASOF
4621  026030  013737  002010  001124  MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
4622  026036  042737  177400  001124  BIC     #177400,$GDDAT  ;INITIALIZE READ ONLY BITS
4623  026044  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
4624  026052  001404                  BEQ     2$              ;YES, TEST IF ANY OTHER REG MODIFIED
4625  026054  012737  067001  001322  MOV     #EM1007,EM3N+2  ;LOAD ERROR MESSAGE
4626  026062  104003                  ERROR   3
4627  026064                  2$:
4628  026064  016237  000000  001126  MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
4629  026072  022737  000200  001126  CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
4630  026100  001407                  BEQ     3$              ;YES, CONTINUE
4631  026102  012737  000100  001124  MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
4632  026110  012737  067314  001322  MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
4633  026116  104003                  ERROR   3
4634  026120                  3$:
4635  026120  016237  000004  001126  MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
4636  026126  001406                  BEQ     4$              ;CHECK IF ZERO
4637  026130  005037  001124          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4638  026134  012737  067365  001322  MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
4639  026142  104003                  ERROR   3
4640  026144                  4$:
4641  026144  016237  000002  001126  MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
4642  026152  001406                  BEQ     5$              ;CHECK IF ZERO
```

```
 4643  026154  005037  001124                    CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 4644  026160  012737  067335  001322            MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
 4645  026166  104003                            ERROR   3
 4646  026170                            5$:
 4647  026170  016237  000006  001126            MOV     RKDA(R2),$BDDAT     ;STORE DISK AVERAGE REG
 4648  026176  001406                            BEQ     6$                  ;CHECK IF ZERO
 4649  026200  005037  001124                    CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 4650  026204  012737  067412  001322            MOV     #EM1020,EM3N+2      ;LOAD ERROR MESSAGE
 4651  026212  104003                            ERROR   3
 4652  026214                            6$:
 4653  026214  016237  000010  001126            MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
 4654  026222  022737  000100  001126            CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
 4655  026230  001407                            BEQ     8$                  ;YES,CONTINUE
 4656  026232  012737  000100  001124            MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
 4657  026240  012737  067450  001322            MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
 4658  026246  104003                            ERROR   3
 4659  026250                            8$:
 4660  026250  016237  000012  001126            MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
 4661  026256  001406                            BEQ     9$                  ;CHECK IF ZERO
 4662  026260  005037  001124                    CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 4663  026264  012737  067471  001322            MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
 4664  026272  104003                            ERROR   3
 4665  026274  016237  000014  001126  9$:       MOV     RKER(R2),$BDDAT     ;STORE ERROR REG
 4666  026302  001406                            BEQ     10$                 ;CHECK IF ZERO
 4667  026304  005037  001124                    CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 4668  026310  012737  067527  001322            MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
 4669  026316  104003                            ERROR   3
 4670  026320                            10$:
 4671  026320  016237  000020  001126            MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG
 4672  026326  001406                            BEQ     12$                 ;CHECK IF ZERO
 4673  026330  005037  001124                    CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 4674  026334  012737  067624  001322            MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
 4675  026342  104003                            ERROR   3
 4676  026344                            12$:
 4677  026344  016237  000026  001126            MOV     RKMR1(R2),$BDDAT    ;STORE MAINTENANCE REG.1
 4678  026352  012737  002000  001124            MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
 4679  026360  032737  020000  001126            BIT     #ECCW,$BDDAT
 4680  026366  001403                            BEQ     13$
 4681  026370  052737  020000  001124            BIS     #ECCW,$GDDAT
 4682  026376  023737  001124  001126  13$:      CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
 4683  026404  001404                            BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
 4684  026406  012737  067651  001322            MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
 4685  026414  104003                            ERROR   3
 4686  026416                            14$:
 4687  026416  016237  000032  001126            MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
 4688  026424  001406                            BEQ     15$                 ;CHECK IF ZERO
 4689  026426  005037  001124                    CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 4690  026432  012737  067727  001322            MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
 4691  026440  104003                            ERROR   3
 4692  026442  016237  000030  001126  15$:      MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
 4693  026450  012737  004066  001124  16$:      MOV     #4066,$GDDAT        ;USE 4066
 4694  026456  023737  001124  001126  17$:      CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
 4695  026464  001404                            BEQ     18$                 ;YES,INITIALIZE RK611
 4696  026466  012737  067702  001322            MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
 4697  026474  104003                            ERROR   3
 4698  026476  016237  000016  002014  18$:      MOV     RKASOF(R2),PREREG       ;GET PREVIOUS CONTENTS
```

```
4699  026504  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4700  026512  016237  000016  001126          MOV     RKASOF(R2),$BDDAT       ;GET CURRENT VALUE
4701  026520  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4702  026524  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
4703  026532  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4704  026534  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4705  026542  012737  067001  001332          MOV     #EM1007,EM4N+2
4706  026550  104004                          ERROR   4
4707  026552  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
4708  026554  000261                          SEC                     ;SHIFT IN ONE
4709  026556  006137  002010                  ROL     CONFIG
4710  026562  005301                          DEC     R1              ;CHECK IF FINISHED
4711  026564  001402                          BEQ     TST34           ;;YES, GO ON TO NEXT TEST
4712  026566  000137  026010                  JMP     1$
4713
4714                                  ;;*****************************************************************
4715                                  ;*TEST 34       REGISTER INTERACTION USING ATTN/OFFSET (PART 4)
4716                                  ;*
4717                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4718                                  ;*      WRITE THE WORD COUNT REGISTER WITH 0.
4719                                  ;*
4720                                  ;*      WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
4721                                  ;*      WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
4722                                  ;*      MAKE SURE NO INTERACTION TAKES PLACE.
4723                                  ;*
4724                                  ;*      100000  174000  177600  177770  000000
4725                                  ;*      140000  176000  177700  177774
4726                                  ;*      160000  177000  177740  177776
4727                                  ;*      170000  177400  177760  177777
4728                                  ;*
4729                                  ;;*****************************************************************
4730  026572  000004          TST34:  SCOPE
4731  026574  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
4732  026602  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
4733  026606  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
4734  026612  012737  064351  001320          MOV     #EM17,EM3N      ;LOAD ERROR MESSAGE
4735  026620  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4736  026626  012737  026634  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
4737                                                                  ;  SUBTEST LOOP
4738
4739  026634                          1$:
4740  026634  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
4741  026640  013762  002010  000016          MOV     CONFIG,RKASOF(R2)       ;WRITE RKASOF
4742  026646  016237  000016  001126          MOV     RKASOF(R2),$BDDAT       ;STORE RKASOF
4743  026654  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
4744  026662  042737  177400  001124          BIC     #177400,$GDDAT  ;INITIALIZE READ ONLY BITS
4745  026670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
4746  026676  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
4747  026700  012737  067001  001322          MOV     #EM1007,EM3N+2  ;LOAD ERROR MESSAGE
4748  026706  104003                          ERROR   3
4749  026710                          2$:
4750  026710  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
4751  026716  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
4752  026724  001407                          BEQ     3$              ;YES, CONTINUE
4753  026726  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
4754  026734  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
```

```
4755  026742  104003                          ERROR   3
4756  026744                          3$:
4757  026744  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
4758  026752  001406                          BEQ     4$              ;CHECK IF ZERO
4759  026754  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4760  026760  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
4761  026766  104003                          ERROR   3
4762  026770                          4$:
4763  026770  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
4764  026776  001406                          BEQ     5$              ;CHECK IF ZERO
4765  027000  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4766  027004  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
4767  027012  104003                          ERROR   3
4768  027014                          5$:
4769  027014  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
4770  027022  001406                          BEQ     6$              ;CHECK IF ZERO
4771  027024  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4772  027030  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
4773  027036  104003                          ERROR   3
4774  027040                          6$:
4775  027040  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
4776  027046  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
4777  027054  001407                          BEQ     8$              ;YES,CONTINUE
4778  027056  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
4779  027064  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
4780  027072  104003                          ERROR   3
4781  027074                          8$:
4782  027074  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
4783  027102  001406                          BEQ     9$              ;CHECK IF ZERO
4784  027104  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4785  027110  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
4786  027116  104003                          ERROR   3
4787  027120  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
4788  027126  001406                          BEQ     10$             ;CHECK IF ZERO
4789  027130  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4790  027134  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
4791  027142  104003                          ERROR   3
4792  027144                          10$:
4793  027144  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
4794  027152  001406                          BEQ     12$             ;CHECK IF ZERO
4795  027154  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4796  027160  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
4797  027166  104003                          ERROR   3
4798  027170                          12$:
4799  027170  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
4800  027176  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
4801  027204  032737  020000  001126          BIT     #ECCW,$BDDAT
4802  027212  001403                          BEQ     13$
4803  027214  052737  020000  001124          BIS     #ECCW,$GDDAT
4804  027222  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
4805  027230  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
4806  027232  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
4807  027240  104003                          ERROR   3
4808  027242                          14$:
4809  027242  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4810  027250  001406                          BEQ     15$             ;CHECK IF ZERO
```

```
4811  027252  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4812  027256  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
4813  027264  104003                          ERROR   3
4814  027266  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4815  027274  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
4816  027302  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
4817  027310  001404                          BEQ     18$             ;YES,INITIALIZE RK611
4818  027312  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
4819  027320  104003                          ERROR   3
4820  027322  016237  000016  002014  18$:    MOV     RKASOF(R2),PREREG       ;GET PREVIOUS CONTENTS
4821  027330  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4822  027336  016237  000016  001126          MOV     RKASOF(R2),$BDDAT       ;GET CURRENT VALUE
4823  027344  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4824  027350  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
4825  027356  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4826  027360  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4827  027366  012737  067001  001332          MOV     #EM1007,EM4N+2
4828  027374  104004                          ERROR   4
4829  027376  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
4830  027400  000261                          SEC                     ;SHIFT IN ONE
4831  027402  006037  002010                  ROR     CONFIG
4832  027406  005301                          DEC     R1              ;CHECK IF FINISHED
4833  027410  001402                          BEQ     TST35           ;;YES, GO ON TO NEXT TEST
4834  027412  000137  026634                  JMP     1$
4835
4836                                  ;;*****************************************************************
4837                                  ;*TEST 35       REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1)
4838                                  ;*
4839                                  ;*      RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
4840                                  ;*      WRITE THE WORD COUNT REGISTER WITH ZERO.
4841                                  ;*
4842                                  ;*      WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
4843                                  ;*      SUCH THAT SUBSYSTEM CLEAR IS RESET, CHECK COMMAND AND STATUS
4844                                  ;*      REG. 2 AND CHECK FOR REGISTER INTERACTION.
4845                                  ;*
4846                                  ;*      000000  000010  000400  010000
4847                                  ;*      000001  000020  001000  020000
4848                                  ;*      000002  000100  002000  040000
4849                                  ;*      000004  000200  004000  100000
4850                                  ;*
4851                                  ;*****************************************************************
4852  027416  000004                  TST35:  SCOPE
4853  027420  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
4854  027426  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
4855  027432  012737  000001  002010          MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
4856  027440  012737  000001  002012          MOV     #000001,CONFG1
4857  027446  012737  064426  001320          MOV     #EM18,EM3N      ;LOAD ERROR MESSAGE
4858  027454  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4859  027462  012737  027470  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
4860                                                                  ;  SUBTEST LOOP
4861
4862  027470                          1$:
4863  027470  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
4864  027474  013762  002010  000010          MOV     CONFIG,RKCS2(R2)        ;WRITE RKCS2
4865  027502  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;STORE RKCS2
4866  027510  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
```

```
4867  027516  042737  177600  001124          BIC     #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,$GDDAT
4868  027524  052737  000100  001124          BIS     #IR,$GDDAT      ;INITIALIZE READ ONLY BITS
4869  027532  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS2 CORRECT
4870  027540  001404                          BEQ     2$              ;YES, TEST IF ANY OTHER REG MODIFIED
4871  027542  012737  066704  001322          MOV     #EM1004,EM3N+2  ;LOAD ERROR MESSAGE
4872  027550  104003                          ERROR   3
4873  027552                          2$:
4874  027552  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
4875  027560  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
4876  027566  001407                          BEQ     3$              ;YES, CONTINUE
4877  027570  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
4878  027576  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
4879  027604  104003                          ERROR   3
4880  027606                          3$:
4881  027606  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
4882  027614  001406                          BEQ     4$              ;CHECK IF ZERO
4883  027616  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4884  027622  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
4885  027630  104003                          ERROR   3
4886  027632                          4$:
4887  027632  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
4888  027640  001406                          BEQ     5$              ;CHECK IF ZERO
4889  027642  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4890  027646  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
4891  027654  104003                          ERROR   3
4892  027656                          5$:
4893  027656  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
4894  027664  001406                          BEQ     6$              ;CHECK IF ZERO
4895  027666  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4896  027672  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
4897  027700  104003                          ERROR   3
4898  027702                          6$:
4899  027702  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
4900  027710  001406                          BEQ     7$              ;CHECK IF ZERO
4901  027712  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4902  027716  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
4903  027724  104003                          ERROR   3
4904  027726                          7$:
4905  027726  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
4906  027734  001406                          BEQ     9$              ;CHECK IF ZERO
4907  027736  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4908  027742  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
4909  027750  104003                          ERROR   3
4910  027752  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
4911  027760  001406                          BEQ     10$             ;CHECK IF ZERO
4912  027762  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4913  027766  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
4914  027774  104003                          ERROR   3
4915  027776                          10$:
4916  027776  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
4917  030004  001406                          BEQ     12$             ;CHECK IF ZERO
4918  030006  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4919  030012  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
4920  030020  104003                          ERROR   3
4921  030022                          12$:
4922  030022  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
```

```
4923  030030  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
4924  030036  032737  020000  001126          BIT     #ECCW,$BDDAT
4925  030044  001403                          BEQ     13$
4926  030046  052737  020000  001124          BIS     #ECCW,$GDDAT
4927  030054  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
4928  030062  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
4929  030064  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
4930  030072  104003                          ERROR   3
4931  030074                          14$:
4932  030074  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4933  030102  001406                          BEQ     15$             ;CHECK IF ZERO
4934  030104  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
4935  030110  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
4936  030116  104003                          ERROR   3
4937  030120  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4938  030126  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
4939  030134  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
4940  030142  001404                          BEQ     18$             ;YES,INITIALIZE RK611
4941  030144  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
4942  030152  104003                          ERROR   3
4943  030154  016237  000010  002014  18$:    MOV     RKCS2(R2),PREREG        ;GET PREVIOUS CONTENTS
4944  030162  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4945  030170  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;GET CURRENT VALUE
4946  030176  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
4947  030204  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS2 CORRECT
4948  030212  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
4949  030214  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
4950  030222  012737  066704  001332          MOV     #EM1004,EM4N+2
4951  030230  104004                          ERROR   4
4952  030232  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
4953  030234  000241                          CLC                     ;SHIFT IN ZERO
4954  030236  006137  002012                  ROL     CONFG1
4955  030242  013737  002012  002010          MOV     CONFG1,CONFIG   ;MAKE SURE SUBSYSTEM CLEAR
4956  030250  042737  000040  002010          BIC     #SCLR,CONFIG    ;  DOES NOT SET
4957  030256  005301                          DEC     R1              ;CHECK IF FINISHED
4958  030260  001402                          BEQ     TST36           ;;YES, GO ON TO NEXT TEST
4959  030262  000137  027470                  JMP     1$
4960
4961                          ;;*****************************************************************
4962                          ;*TEST 36       REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 2)
4963                          ;*
4964                          ;*      RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
4965                          ;*      WRITE THE WORD COUNT REGISTER WITH ZERO.
4966                          ;*
4967                          ;*      WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
4968                          ;*      SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
4969                          ;*      REG. 2 AND CHECK FOR REGISTER INTERACTION.
4970                          ;*
4971                          ;*      177737  177727  177337  167737
4972                          ;*      177736  177717  176737  157737
4973                          ;*      177735  177637  175737  137737
4974                          ;*      177733  177537  173737  077737
4975                          ;*
4976                          ;*****************************************************************
4977  030266  000004          TST36:  SCOPE
4978  030270  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
```

```
 4979  030276  012701  000021            MOV     #17.,R1             ;LOAD NUMBER OF PATTERNS
 4980  030302  012737  177736  002010    MOV     #177736,CONFIG      ;LOAD INITIAL CONFIGURATION
 4981  030310  012737  177776  002012    MOV     #177776,CONFG1
 4982  030316  012737  064426  001320    MOV     #EM18,EM3N          ;LOAD ERROR MESSAGE
 4983  030324  012762  100000  000000    MOV     #CCLR,RKCS1(R2)     ;CLEAR RK611 WITH CONTROLLER CLEAR
 4984  030332  012737  030340  001110    MOV     #1$,$LPERR          ;LOAD LOOP ON ERROR LOCATION FOR
 4985                                                                ;  SUBTEST LOOP
 4986
 4987  030340                          1$:
 4988  030340  005062  000002            CLR     RKWC(R2)            ;CLEAR WORD COUNT REG.
 4989  030344  013762  002010  000010    MOV     CONFIG,RKCS2(R2)            ;WRITE RKCS2
 4990  030352  016237  000010  001126    MOV     RKCS2(R2),$BDDAT            ;STORE RKCS2
 4991  030360  013737  002010  001124    MOV     CONFIG,$GDDAT       ;PREPARE EXPECTED RESULTS
 4992  030366  042737  177600  001124    BIC     #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,$GDDAT
 4993  030374  052737  000100  001124    BIS     #IR,$GDDAT          ;INITIALIZE READ ONLY BITS
 4994  030402  023737  001124  001126    CMP     $GDDAT,$BDDAT       ;CHECK IF RKCS2 CORRECT
 4995  030410  001404                    BEQ     2$                  ;YES, TEST IF ANY OTHER REG MODIFIED
 4996  030412  012737  066704  001322    MOV     #EM1004,EM3N+2      ;LOAD ERROR MESSAGE
 4997  030420  104003                    ERROR   3
 4998  030422                          2$:
 4999  030422  016237  000000  001126    MOV     RKCS1(R2),$BDDAT    ;STORE COMMAND AND STATUS REG. 1
 5000  030430  022737  000200  001126    CMP     #RDY,$BDDAT         ;CHECK IF CS1 CORRECT
 5001  030436  001407                    BEQ     3$                  ;YES, CONTINUE
 5002  030440  012737  000100  001124    MOV     #IR,$GDDAT          ;LOAD EXPECTED RESULTS
 5003  030446  012737  067314  001322    MOV     #EM1017,EM3N+2      ;LOAD ERROR MESSAGE
 5004  030454  104003                    ERROR   3
 5005  030456                          3$:
 5006  030456  016237  000004  001126    MOV     RKBA(R2),$BDDAT     ;STORE BUS AND REG
 5007  030464  001406                    BEQ     4$                  ;CHECK IF ZERO
 5008  030466  005037  001124            CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 5009  030472  012737  067365  001322    MOV     #EM1019,EM3N+2      ;LOAD ERROR MESSAGE
 5010  030500  104003                    ERROR   3
 5011  030502                          4$:
 5012  030502  016237  000002  001126    MOV     RKWC(R2),$BDDAT     ;STORE WORD COUNT REG
 5013  030510  001406                    BEQ     5$                  ;CHECK IF ZERO
 5014  030512  005037  001124            CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 5015  030516  012737  067335  001322    MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
 5016  030524  104003                    ERROR   3
 5017  030526                          5$:
 5018  030526  016237  000006  001126    MOV     RKDA(R2),$BDDAT     ;STORE DISK AVERAGE REG
 5019  030534  001406                    BEQ     6$                  ;CHECK IF ZERO
 5020  030536  005037  001124            CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 5021  030542  012737  067412  001322    MOV     #EM1020,EM3N+2      ;LOAD ERROR MESSAGE
 5022  030550  104003                    ERROR   3
 5023  030552                          6$:
 5024  030552  016237  000016  001126    MOV     RKASOF(R2),$BDDAT   ;STORE ATTENTION SUMMARY/OFFSET REG.
 5025  030560  001406                    BEQ     7$                  ;CHECK IF ZERO
 5026  030562  005037  001124            CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 5027  030566  012737  067556  001322    MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
 5028  030574  104003                    ERROR   3
 5029  030576                          7$:
 5030  030576  016237  000012  001126    MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
 5031  030604  001406                    BEQ     9$                  ;CHECK IF ZERO
 5032  030606  005037  001124            CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 5033  030612  012737  067471  001322    MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
 5034  030620  104003                    ERROR   3
```

```
5035  030622  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STOPE ERROR REG
5036  030630  001406                          BEQ     10$             ;CHECK IF ZERO
5037  030632  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5038  030636  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
5039  030644  104003                          ERROR   3
5040  030646                          10$:
5041  030646  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
5042  030654  001406                          BEQ     12$             ;CHECK IF ZERO
5043  030656  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5044  030662  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
5045  030670  104003                          ERROR   3
5046  030672                          12$:
5047  030672  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
5048  030700  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
5049  030706  032737  020000  001126          BIT     #ECCW,$BDDAT
5050  030714  001403                          BEQ     13$
5051  030716  052737  020000  001124          BIS     #ECCW,$GDDAT
5052  030724  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
5053  030732  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
5054  030734  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
5055  030742  104003                          ERROR   3
5056  030744                          14$:
5057  030744  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5058  030752  001406                          BEQ     15$             ;CHECK IF ZERO
5059  030754  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5060  030760  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
5061  030766  104003                          ERROR   3
5062  030770  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
5063  030776  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
5064  031004  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
5065  031012  001404                          BEQ     18$             ;YES,INITIALIZE RK611
5066  031014  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
5067  031022  104003                          ERROR   3
5068  031024  016237  000010  002014  18$:    MOV     RKCS2(R2),PREREG        ;GET PREVIOUS CONTENTS
5069  031032  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5070  031040  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;GET CURRENT VALUE
5071  031046  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
5072  031054  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS2 CORRECT
5073  031062  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
5074  031064  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
5075  031072  012737  066704  001332          MOV     #EM1004,EM4N+2
5076  031100  104004                          ERROR   4
5077  031102  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
5078  031104  000261                          SEC                     ;SHIFT IN ONE
5079  031106  006137  002012                  ROL     CONFG1
5080  031112  013737  002012  002010          MOV     CONFG1,CONFIG   ;MAKE SURE SUBSYSTEM CLEAR
5081  031120  042737  000040  002010          BIC     #SCLR,CONFIG    ;  DOES NOT SET
5082  031126  005301                          DEC     R1              ;CHECK IF FINISHED
5083  031130  001402                          BEQ     TST37           ;;YES, GO ON TO NEXT TEST
5084  031132  000137  030340                  JMP     1$
5085
5086                          ;;*****************************************************************
5087                          ;*TEST 37        REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 3)
5088                          ;*
5089                          ;*      RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
5090                          ;*      WRITE THE WORD COUNT REGISTER WITH ZERO.
```

```
                                ;*
                                ;*      WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
                                ;*      SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
                                ;*      REG. 2 AND CHECK FOR REGISTER INTERACTION.
                                ;*
                                ;*      000001  000037  001737  037737
                                ;*      000003  000137  003737  077737
                                ;*      000007  000337  007737  177737
                                ;*      000017  000737  017737  000000
                                ;*
                                ;*****************************************************************
031136  000004                  TST37:  SCOPE
031140  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
031146  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
031152  005037  002010                  CLR     CONFIG          ;LOAD INITIAL OCNFIGURATION
031156  005037  002012                  CLR     CONFG1
031162  012737  064426  001320          MOV     #EM18,EM3N      ;LOAD ERROR MESSAGE
031170  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
031176  012737  031204  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
031204                          1$:
031204  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
031210  013762  002010  000010          MOV     CONFIG,RKCS2(R2)        ;WRITE RKCS2
031216  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;STORE RKCS2
031224  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
031232  042737  177600  001124          BIC     #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,$GDDAT
031240  052737  000100  001124          BIS     #IR,$GDDAT      ;INITIALIZE READ ONLY BITS
031246  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS2 CORRECT
031254  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
031256  012737  066704  001322          MOV     #EM1004,EM3N+2  ;LOAD ERROR MESSAGE
031264  104003                          ERROR   3
031266                          2$:
031266  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
031274  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
031302  001407                          BEQ     3$              ;YES, CONTINUE
031304  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
031312  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
031320  104003                          ERROR   3
031322                          3$:
031322  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
031330  001406                          BEQ     4$              ;CHECK IF ZERO
031332  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
031336  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
031344  104003                          ERROR   3
031346                          4$:
031346  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
031354  001406                          BEQ     5$              ;CHECK IF ZERO
031356  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
031362  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
031370  104003                          ERROR   3
031372                          5$:
031372  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
031400  001406                          BEQ     6$              ;CHECK IF ZERO
031402  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
031406  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
```

```
5147  031414  104003                          ERROR   3
5148  031416                          6$:
5149  031416  016237  000016  001126          MOV     RKASOF(P2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
5150  031424  001406                          BEQ     7$                ;CHECK IF ZERO
5151  031426  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
5152  031432  012737  067556  001322          MOV     #EM1024,EM3N+2    ;LOAD ERROR MESSAGE
5153  031440  104003                          ERROR   3
5154  031442                          7$:
5155  031442  016237  000012  001126          MOV     RKDS(R2),$BDDAT   ;STORE DRIVE STATUS REG
5156  031450  001406                          BEQ     9$                ;CHECK IF ZERO
5157  031452  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
5158  031456  012737  067471  001322          MOV     #EM1022,EM3N+2    ;LOAD ERROR MESSAGE
5159  031464  104003                          ERROR   3
5160  031466  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT   ;STORE ERROR REG
5161  031474  001406                          BEQ     10$               ;CHECK IF ZERO
5162  031476  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
5163  031502  012737  067527  001322          MOV     #EM1023,EM3N+2    ;LOAD ERROR MESSAGE
5164  031510  104003                          ERROR   3
5165  031512                          10$:
5166  031512  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
5167  031520  001406                          BEQ     12$               ;CHECK IF ZERO
5168  031522  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
5169  031526  012737  067624  001322          MOV     #EM1025,EM3N+2    ;LOAD ERROR MESSAGE
5170  031534  104003                          ERROR   3
5171  031536                          12$:
5172  031536  016237  000026  001126          MOV     RKMR1(R2),$BDDAT  ;STORE MAINTENANCE REG.1
5173  031544  012737  002000  001124          MOV     #MEWD,$GDDAT      ;LOAD EXPECTED MR1
5174  031552  032737  020000  001126          BIT     #ECCW,$BDDAT
5175  031560  001403                          BEQ     13$
5176  031562  052737  020000  001124          BIS     #ECCW,$GDDAT
5177  031570  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT     ;CHECK IF MR1 CORRECT
5178  031576  001404                          BEQ     14$               ;YES,ISSUE CONTROLLER CLEAR
5179  031600  012737  067651  001322          MOV     #EM1026,EM3N+2    ;LOAD ERROR MESSAGE
5180  031606  104003                          ERROR   3
5181  031610                          14$:
5182  031610  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5183  031616  001406                          BEQ     15$               ;CHECK IF ZERO
5184  031620  005037  001124                  CLR     $GDDAT            ;LOAD EXPECTED CONTENTS
5185  031624  012737  067727  001322          MOV     #EM1030,EM3N+2    ;LOAD ERROR MESSAGE
5186  031632  104003                          ERROR   3
5187  031634  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
5188  031642  012737  004066  001124  16$:    MOV     #4066,$GDDAT      ;USE 4066
5189  031650  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT     ;CHECK IF ECC POSITION CORRECT
5190  031656  001404                          BEQ     18$               ;YES,INITIALIZE RK611
5191  031660  012737  067702  001322          MOV     #EM1029,EM3N+2    ;LOAD ERROR MESSAGE
5192  031666  104003                          ERROR   3
5193  031670  016237  000010  002014  18$:    MOV     RKCS2(R2),PREREG          ;GET PREVIOUS CONTENTS
5194  031676  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5195  031704  016237  000010  001126          MOV     RKCS2(R2),$BDDAT          ;GET CURRENT VALUE
5196  031712  012737  000100  001124          MOV     #IR,$GDDAT        ;LOAD EXPECTED CONTENTS
5197  031720  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;CHECK IF RKCS2 CORRECT
5198  031726  001407                          BEQ     19$               ;YES, CHECK IF FINISHED
5199  031730  012737  063403  001330          MOV     #EM3,EM4N         ;LOAD ERROR MESSAGE
5200  031736  012737  066704  001332          MOV     #EM1004,EM4N+2
5201  031744  104004                          ERROR   4
5202  031746  104415                  19$:    SCOPE                     ;CHECK IF LOOP ON ERROR
```

```
                        SEC                     ;SHIFT IN ONE
                        ROL     CONFG1
                        MOV     CONFG1,CONFIG   ;MAKE SURE SUBSYSTEM CLEAR
                        BIC     #SCLR,CONFIG    ;  DOES NOT SET
                        DEC     R1              ;CHECK IF FINISHED
                        BEQ     TST40           ;;YES, GO ON TO NEXT TEST
                        JMP     1$

                ;;*****************************************************************
                ;*TEST 40        REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 4)
                ;*
                ;*      RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
                ;*      WRITE THE WORD COUNT REGISTER WITH ZERO.
                ;*
                ;*      WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
                ;*      SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
                ;*      REG. 2 AND CHECK FOR REGISTER INTERACTION.
                ;*
                ;*      100000  174000  177600  177734
                ;*      140000  176000  177700  177736
                ;*      160000  177000  177720  177737
                ;*      170000  177400  177730  000000
                ;*
                ;*****************************************************************
TST40:  SCOPE
                        MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
                        MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
                        CLR     CONFIG          ;LOAD INITIAL OCNFIGURATION
                        CLR     CONFG1
                        MOV     #EM18,EM3N      ;LOAD ERROR MESSAGE
                        MOV     #CCLR,RKCS1(P2) ;CLEAR RK611 WITH CONTROLLER CLEAR
                        MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                ;  SUBTEST LOOP

1$:
                        CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
                        MOV     CONFIG,RKCS2(R2)        ;WRITE RKCS2
                        MOV     RKCS2(R2),$BDDAT        ;STORE RKCS2
                        MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
                        BIC     #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,$GDDAT
                        BIS     #IR,$GDDAT      ;INITIALIZE READ ONLY BITS
                        CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS2 CORRECT
                        BEQ     2$              ;YES, TEST IF ANY OTHER REG MODIFIED
                        MOV     #EM1004,EM3N+2  ;LOAD ERROR MESSAGE
                        ERROR   3
2$:
                        MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
                        CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
                        BEQ     3$              ;YES, CONTINUE
                        MOV     #EM1017,EM3N+2  ;LOAD EXPECTED RESULTS
                        ERROR   3               ;LOAD ERROR MESSAGE
3$:
                        MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
                        BEQ     4$              ;CHECK IF ZERO
                        CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
```

```
5259  032202  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
5260  032210  104003                          ERROR   3
5261  032212                          4$:
5262  032212  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
5263  032220  001406                          BEQ     5$              ;CHECK IF ZERO
5264  032222  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5265  032226  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
5266  032234  104003                          ERROR   3
5267  032236                          5$:
5268  032236  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
5269  032244  001406                          BEQ     6$              ;CHECK IF ZERO
5270  032246  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5271  032252  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
5272  032260  104003                          ERROR   3
5273  032262                          6$:
5274  032262  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
5275  032270  001406                          BEQ     7$              ;CHECK IF ZERO
5276  032272  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5277  032276  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
5278  032304  104003                          ERROR   3
5279  032306                          7$:
5280  032306  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
5281  032314  001406                          BEQ     9$              ;CHECK IF ZERO
5282  032316  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5283  032322  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
5284  032330  104003                          ERROR   3
5285  032332  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
5286  032340  001406                          BEQ     10$             ;CHECK IF ZERO
5287  032342  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5288  032346  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
5289  032354  104003                          ERROR   3
5290  032356                          10$:
5291  032356  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
5292  032364  001406                          BEQ     12$             ;CHECK IF ZERO
5293  032366  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5294  032372  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
5295  032400  104003                          ERROR   3
5296  032402                          12$:
5297  032402  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
5298  032410  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
5299  032416  032737  020000  001126          BIT     #ECCW,$BDDAT
5300  032424  001403                          BEQ     13$
5301  032426  052737  020000  001124          BIS     #ECCW,$GDDAT
5302  032434  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
5303  032442  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
5304  032444  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
5305  032452  104003                          ERROR   3
5306  032454                          14$:
5307  032454  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5308  032462  001406                          BEQ     15$             ;CHECK IF ZERO
5309  032464  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5310  032470  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
5311  032476  104003                          ERROR   3
5312  032500  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
5313  032506  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
5314  032514  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
```

```
5315  032522  001404                          BEQ     18$                     ;YES,INITIALIZE RK611
5316  032524  012737  067702  001322          MOV     #EM1029,EM3N+2          ;LOAD ERROR MESSAGE
5317  032532  104003                          ERROR   3
5318  032534  016237  000010  002014  18$:    MOV     RKCS2(R2),PREREG                ;GET PREVIOUS CONTENTS
5319  032542  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5320  032550  016237  000010  001126          MOV     RKCS2(R2),$BDDAT                ;GET CURRENT VALUE
5321  032556  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
5322  032564  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKCS2 CORRECT
5323  032572  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
5324  032574  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
5325  032602  012737  066704  001332          MOV     #EM1004,EM4N+2
5326  032610  104004                          ERROR   4
5327  032612  104415                  19$:    SCOPE                   ;CHECK IF LOOP ON ERROR
5328  032614  000261                          SEC                     ;SHIFT IN ONE
5329  032616  006037  002012                  ROR     CONFG1
5330  032622  013737  002012  002010          MOV     CONFG1,CONFIG   ;MAKE SURE SUBSYSTEM CLEAR
5331  032630  042737  000040  002010          BIC     #SCLR,CONFIG    ;  DOES NOT SET
5332  032636  005301                          DEC     R1              ;CHECK IF FINISHED
5333  032640  001402                          BEQ     TST41           ;;YES, GO ON TO NEXT TEST
5334  032642  000137  032050                  JMP     1$
5335
5336                                  ;;*****************************************************************
5337                                  ;*TEST 41       CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS
5338                                  ;*
5339                                  ;*      THIS TEST WILL TEST THE ABILITY OF THE SUBSYSTEM CLEAR TO
5340                                  ;*      INITIALIZE THE BUS ADDRESS REGISTER AND COMMAND
5341                                  ;*      AND STATUS REGISTER 1.  IT WILL ALSO VERIFY THAT ALL
5342                                  ;*      OTHER REGISTERS REMAIN IN THE INITIALIZED STATE.
5343                                  ;*
5344                                  ;;*****************************************************************
5345  032646  000004                  TST41:  SCOPE
5346  032650  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
5347  032656  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
5348  032662  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5349  032670  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG
5350  032674  012762  177776  000004          MOV     #177776,RKBA(R2) ;WRITE WORD COUNT REG.
5351  032702  012762  013776  000000          MOV     #013776,RKCS1(R2) ;WRITE COMMAND AND STATUS REG 1
5352  032710  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;ISSUE A SUBSYSTEM CLEAR
5353  032716  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS ADDRESS REG.
5354  032724  001414                          BEQ     1$              ;CHECK IF ZERO
5355  032726  012737  177776  002014          MOV     #177776,PREREG  ;LOAD PREVIOUS CONTENTS
5356  032734  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5357  032740  012737  064557  001330          MOV     #EM21,EM4N      ;LOAD ERROR MESSAGE
5358  032746  012737  066637  001330          MOV     #EM1002,EM4N
5359  032754  104004                          ERROR   4
5360  032756  016237  000000  001126  1$:     MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
5361  032764  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
5362  032772  001415                          BEQ     2$              ;YES, CHECK IF ALL OTHER REGISTER
5363                                                                  ;  IN INITIALIZE STATE
5364  032774  012737  013776  002014          MOV     #013776,PREREG  ;LOAD PREVIOUS CONTENTS
5365  033002  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
5366  033010  012737  064634  001330          MOV     #EM22,EM4N;     LOAD ERROR MESSAGE
5367  033016  012737  066637  001332          MOV     #EM1002,EM4N+2
5368  033024  104004                          ERROR   4
5369  033026  012737  064705  001310  2$:     MOV     #EM23,EM2N      ;LOAD ERROR MESSAGE
5370  033034  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
```

```
5371  033042  001406                        BEQ     3$                  ;CHECK IF ZERO
5372  033044  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5373  033050  012737  066612  001312        MOV     #EM1001,EM2N+2      ;LOAD ERROR MESSAGE
5374  033056  104002                        ERROR   2
5375  033060  016237  000006  001126  3$:   MOV     RKDA(R2),$BDDAT     ;STORE DISK ADDRESS REG.
5376  033066  001406                        BEQ     4$                  ;CHECK IF ZERO
5377  033070  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5378  033074  012737  066661  001312        MOV     #EM1003,EM2N+2      ;LOAD ERROR MESSAGE
5379  033102  104002                        ERROR   2
5380  033104  016237  000016  001126  4$:   MOV     RKASOF(R2),$BDDAT   ;STORE ATTENTION SUMMARY AND OFFSET REG.
5381  033112  001406                        BEQ     5$                  ;CHECK IF ZERO
5382  033114  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5383  033120  012737  067001  001312        MOV     #EM1007,EM2N+2      ;LOAD ERROR MESSAGE
5384  033126  104002                        ERROR   2
5385  033130  016237  000010  001126  5$:   MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
5386  033136  022737  000100  001126        CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
5387  033144  001407                        BEQ     6$                  ;YES, CONTINUE
5388  033146  012737  000100  001124        MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
5389  033154  012737  066704  001312        MOV     #EM1004,EM2N+2      ;LOAL ERROR MESSAGE
5390  033162  104002                        ERROR   2
5391  033164  016237  000012  001126  6$:   MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG.
5392  033172  001406                        BEQ     7$                  ;CHECK IF ZERO
5393  033174  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5394  033200  012737  066722  001312        MOV     #EM1005,EM2N+2      ;LOAD ERROR MESSAGE
5395  033206  104002                        ERROR   2
5396  033210  016237  000014  001126  7$:   MOV     RKER(R2),$BDDAT     ;STORE ERROR REGISTER
5397  033216  001406                        BEQ     8$                  ;CHECK IF ZERO
5398  033220  005037  001126                CLR     $BDDAT              ;LOAD EXPECTED CONTENTS
5399  033224  012737  066755  001312        MOV     #EM1006,EM2N+2      ;LOAD ERROR MESSAGE
5400  033232  104002                        ERROR   2
5401  033234  016237  000020  001126  8$:   MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG.
5402  033242  001406                        BEQ     10$                 ;CHECK IF ZERO
5403  033244  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5404  033250  012737  067272  001312        MOV     #EM1016,EM2N+2      ;LOAD ERROR MESSAGE
5405  033256  104002                        ERROR   2
5406  033260  016237  000026  001126  10$:  MOV     RKMR1(R2),$BDDAT    ;STORE CYLINDER ADD REG.
5407  033266  012737  002000  001124        MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
5408  033274  032737  020000  001126        BIT     #ECCW,$BDDAT
5409  033302  001403                        BEQ     11$
5410  033304  052737  020000  001124        BIS     #ECCW,$GDDAT
5411  033312  023737  001124  001126  11$:  CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
5412  033320  001404                        BEQ     12$                 ;YES, CONTINUE TEST
5413  033322  012737  067072  001312        MOV     #EM1009,EM2N+2      ;LOAD ERROR MESSAGE
5414  033330  104002                        ERROR   2
5415  033332  016237  000032  001126  12$:  MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
5416  033340  001406                        BEQ     13$                 ;CHECK IF ZERO
5417  033342  005037  001124                CLR     $GDDAT              ;LOAD EXPECTED RESULTS
5418  033346  012737  067142  001312        MOV     #EM1013,EM2N+2      ;LOAD ERROR MESSAGE
5419  033354  104002                        ERROR   2
5420  033356  016237  000030  001126  13$:  MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
5421  033364  022737  004066  001126        CMP     #4066,$BDDAT        ;CHECK IF ECC POSITION CORRECT
5422  033372  001407                        BEQ     TST42               ;;YES,GO TO NEXT TEST
5423  033374  012737  004066  001124        MOV     #4066,$GDDAT        ;LOAD EXPECTED RESULTS
5424  033402  012737  067120  001312        MOV     #EM1012,EM2N+2      ;LOAD ERROR MESSAGE
5425  033410  104002                        ERROR   2
5426                        ;;**************************************************************
```

```
                                        ;*TEST 42       REGISTER INTERACTION USING DRIVE STATUS
                                        ;*
                                        ;*      ISSUE A CONTROLLER CLEAR TO ININITALIZE RK611 REGISTERS.
                                        ;*      WRITE WORD COUNT TO 0, WRITE DRIVE STATUS REGISTER
                                        ;*      WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
                                        ;*      INTERACTION TAKES PLACE.
                                        ;*
                                        ;*****************************************************************
033412  000004                          $ST42:  SCOPE
033414  012737  000764  001200                  MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
033422  013702  001270                          MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
033426  012737  177777  002010                  MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
033434  012737  064015  001320                  MOV     #EM10,EM3N      ;LOAD ERROR MESSAGE
033442  012762  100000  000000                  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
033450  005062  000002                          CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
033454  012762  177777  000012                  MOV     #177777,RKDS(R2) ;WRITE RKDS WITH 177777
033462  016237  000000  001126                  MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
033470  022737  000200  001126                  CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
033476  001407                                  BEQ     1$              ;YES,CONTINUE
033500  012737  000200  001124                  MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
033506  012737  067314  001322                  MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
033514  104003                                  ERROR   3
033516  016237  000004  001126          1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
033524  001406                                  BEQ     2$              ;CHECK IF ZERO
033526  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
033532  012737  067365  001322                  MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
033540  104003                                  ERROR   3
033542  016237  000002  001126          2$:     MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
033550  001406                                  BEQ     3$              ;CHECK IF ZERO
033552  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
033556  012737  067335  001322                  MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
033564  104003                                  ERROR   3
033566  016237  000006  001126          3$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG
033574  001406                                  BEQ     4$              ;CHECK IF ZERO
033576  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
033602  012737  067412  001322                  MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
033610  104003                                  ERROR   3
033612  016237  000016  001126          4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
033620  001406                                  BEQ     5$              ;CHECK IF ZERO
033622  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
033626  012737  067556  001322                  MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
033634  104003                                  ERROR   3
033636  016237  000010  001126          5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
033644  022737  000100  001126                  CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
033652  001407                                  BEQ     6$              ;YES,CONTINUE
033654  012737  000100  001124                  MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
033662  012737  067450  001322                  MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
033670  104003                                  ERROR   3
033672  016237  000012  001126          6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
033700  001406                                  BEQ     7$              ;CHECK IF ZERO
033702  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
033706  012737  067471  001322                  MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
033714  104003                                  ERROR   3
033716  016237  000014  001126          7$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
033724  001406                                  BEQ     8$              ;CHECK IF ZERO
033726  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
```

```
5483  033732  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
5484  033740  104003                          ERROR   3
5485  033742  016237  000020  001126  8$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
5486  033750  001406                          BEQ     10$             ;CHECK IF ZERO
5487  033752  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
5488  033756  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
5489  033764  104003                          ERROR   3
5490  033766  016237  000026  001126  10$:    MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
5491  033774  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
5492  034002  032737  020000  001126          BIT     #ECCW,$BDDAT
5493  034010  001403                          BEQ     11$
5494  034012  052737  020000  001124          BIS     #ECCW,$GDDAT
5495  034020  023737  001124  001126  11$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
5496  034026  001404                          BEQ     12$             ;YES,CONTINUE TEST
5497  034030  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
5498  034036  104003                          ERROR   3
5499  034040  016237  000032  001126  12$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5500  034046  001406                          BEQ     13$             ;CHECK IF ZERO
5501  034050  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5502  034054  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
5503  034062  104003                          ERROR   3
5504  034064  016237  000030  001126  13$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
5505  034072  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
5506  034100  001407                          BEQ     TST43           ;;YES,GO TO NEXT TEST
5507  034102  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
5508  034110  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESAGE
5509  034116  104003                          ERROR   3
5510
5511                                  ;;*****************************************************************
5512                                  ;*TEST 43       REGISTER INTERACTION USING ERROR REGISTER
5513                                  ;*
5514                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5515                                  ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE ERROR REGISTER
5516                                  ;*      WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
5517                                  ;*      INTERACTION TAKES PLACE.
5518                                  ;*
5519                                  ;;*****************************************************************
5520  034120  000004                  TST43:  SCOPE
5521  034122  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
5522  034130  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
5523  034134  012737  177777  002010          MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
5524  034142  012737  064062  001320          MOV     #EM11,EM3N      ;LOAD ERROR MESSAGE
5525  034150  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5526  034156  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
5527  034162  012762  177777  000014          MOV     #177777,RKER(R2) ;WRITE RKER WITH 177777
5528  034170  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
5529  034176  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
5530  034204  001407                          BEQ     1$              ;YES,CONTINUE
5531  034206  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
5532  034214  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
5533  034222  104003                          ERROR   3
5534  034224  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
5535  034232  001406                          BEQ     2$              ;CHECK IF ZERO
5536  034234  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5537  034240  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
5538  034246  104003                          ERROR   3
```

```
 5539  034250  016237  000002  001126  2$:     MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
 5540  034256  001406                          BEQ     3$              ;CHECK IF ZERO
 5541  034260  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5542  034264  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
 5543  034272  104003                          ERROR   3
 5544  034274  016237  000006  001126  3$:     MOV     RKDA(R2) $BDDAT ;STORE DISK ADD REG
 5545  034302  001406                          BEQ     4$              ;CHECK IF ZERO
 5546  034304  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5547  034310  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
 5548  034316  104003                          ERROR   3
 5549  034320  016237  000016  001126  4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
 5550  034326  001406                          BEQ     5$              ;CHECK IF ZERO
 5551  034330  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5552  034334  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
 5553  034342  104003                          ERROR   3
 5554  034344  016237  000010  001126  5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
 5555  034352  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
 5556  034360  001407                          BEQ     6$              ;YES,CONTINUE
 5557  034362  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
 5558  034370  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
 5559  034376  104003                          ERROR   3
 5560  034400  016237  000012  001126  6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
 5561  034406  001406                          BEQ     7$              ;CHECK IF ZERO
 5562  034410  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5563  034414  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 5564  034422  104003                          ERROR   3
 5565  034424  016237  000014  001126  7$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
 5566  034432  001406                          BEQ     8$              ;CHECK IF ZERO
 5567  034434  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5568  034440  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
 5569  034446  104003                          ERROR   3
 5570  034450  016237  000020  001126  8$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
 5571  034456  001406                          BEQ     10$             ;CHECK IF ZERO
 5572  034460  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
 5573  034464  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
 5574  034472  104003                          ERROR   3
 5575  034474  016237  000026  001126  10$:    MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
 5576  034502  012737  002000  001124          MOV     #MEW0,$GDDAT    ;LOAD EXPECTED CONTENTS
 5577  034510  032737  020000  001126          BIT     #ECCW,$BDDAT
 5578  034516  001403                          BEQ     11$
 5579  034520  052737  020000  001124          BIS     #ECCW,$GDDAT
 5580  034526  023737  001124  001126  11$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
 5581  034534  001404                          BEQ     12$             ;YES,CONTINUE TEST
 5582  034536  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
 5583  034544  104003                          ERROR   3
 5584  034546  016237  000032  001126  12$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
 5585  034554  001406                          BEQ     13$             ;CHECK IF ZERO
 5586  034556  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5587  034562  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
 5588  034570  104003                          ERROR   3
 5589  034572  016237  000030  001126  13$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
 5590  034600  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
 5591  034606  001407                          BEQ     TST44           ;;YES,GO TO NEXT TEST
 5592  034610  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
 5593  034616  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESAGE
 5594  034624  104003                          ERROR   3
```

```
                                        ;;*****************************************************************
                                        ;*TEST 44       REGISTER INTERACTION USING MAINT REG 2
                                        ;*
                                        ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
                                        ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
                                        ;*      REGISTER 2 TO 177777 AND NO INTERACTION TAKES PLACE.
                                        ;*
                                        ;*****************************************************************
034626  000004                  TST44:  SCOPE
034630  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
034636  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
034642  012737  177777  002010          MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
034650  012737  064120  001320          MOV     #EM12,EM3N      ;LOAD ERROR MESSAGE
034656  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
034664  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
034670  012762  177777  000034          MOV     #177777,RKMR2(R2) ;WRITE RKMR2 WITH 177777
034676  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
034704  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
034712  001407                          BEQ     1$              ;YES,CONTINUE
034714  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
034722  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
034730  104003                          ERROR   3
034732  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
034740  001406                          BEQ     2$              ;CHECK IF ZERO
034742  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
034746  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
034754  104003                          ERROR   3
034756  016237  000002  001126  2$:     MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
034764  001406                          BEQ     3$              ;CHECK IF ZERO
034766  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
034772  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
035000  104003                          ERROR   3
035002  016237  000006  001126  3$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG
035010  001406                          BEQ     4$              ;CHECK IF ZERO
035012  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
035016  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
035024  104003                          ERROR   3
035026  016237  000016  001126  4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
035034  001406                          BEQ     5$              ;CHECK IF ZERO
035036  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
035042  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
035050  104003                          ERROR   3
035052  016237  000010  001126  5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
035060  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
035066  001407                          BEQ     6$              ;YES,CONTINUE
035070  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
035076  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
035104  104003                          ERROR   3
035106  016237  000012  001126  6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
035114  001406                          BEQ     7$              ;CHECK IF ZERO
035116  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
035122  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
035130  104003                          ERROR   3
035132  016237  000014  001126  7$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
035140  001406                          BEQ     8$              ;CHECK IF ZERO
```

```
5651  035142  005037  001124                      CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5652  035146  012737  067527  001322              MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
5653  035154  104003                              ERROR   3
5654  035156  016237  000020  001126  8$:         MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
5655  035164  001406                              BEQ     10$             ;CHECK IF ZERO
5656  035166  005037  001124                      CLR     $GDDAT          ;LOAD EXPECTED RESULTS
5657  035172  012737  067624  001322              MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
5658  035200  104003                              ERROR   3
5659  035202  016237  000026  001126  10$:        MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
5660  035210  012737  002000  001124              MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
5661  035216  032737  020000  001126              BIT     #ECCW,$BDDAT
5662  035224  001403                              BEQ     11$
5663  035226  052737  020000  001124              BIS     #ECCW,$GDDAT
5664  035234  023737  001124  001126  11$:        CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
5665  035242  001404                              BEQ     12$             ;YES,CONTINUE TEST
5666  035244  012737  067651  001322              MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
5667  035252  104003                              ERROR   3
5668  035254  016237  000032  001126  12$:        MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5669  035262  001406                              BEQ     13$             ;CHECK IF ZERO
5670  035264  005037  001124                      CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5671  035270  012737  067727  001322              MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
5672  035276  104003                              ERROR   3
5673  035300  016237  000030  001126  13$:        MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
5674  035306  022737  004066  001126              CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
5675  035314  001407                              BEQ     TST45           ;;YES,GO TO NEXT TEST
5676  035316  012737  004066  001124              MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
5677  035324  012737  067702  001322              MOV     #EM1029,EM3N+2  ;LOAD ERROR MESAGE
5678  035332  104003                              ERROR   3
5679
5680                                  ;;*****************************************************************
5681                                  ;*TEST 45       REGISTER INTERACTION USING MAINT REG 3
5682                                  ;*
5683                                  ;*       ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5684                                  ;*       WRITE THE WORD COUNT REGISTER TO ZERO, WRITE MAINTENANCE
5685                                  ;*       REGISTER 3 TO 177777 AND NO INTERACTION TAKES PLACE.
5686                                  ;*
5687                                  ;;*****************************************************************
5688  035334  000004                  TST45:  SCOPE
5689  035336  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
5690  035344  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
5691  035350  012737  177777  002010          MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
5692  035356  012737  064160  001320          MOV     #EM13,EM3N      ;LOAD ERROR MESSAGE
5693  035364  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5694  035372  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
5695  035376  012762  177777  000036          MOV     #177777,RKMR3(R2) ;WRITE RKMR3 WITH 177777
5696  035404  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
5697  035412  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
5698  035420  001407                          BEQ     1$              ;YES,CONTINUE
5699  035422  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
5700  035430  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
5701  035436  104003                          ERROR   3
5702  035440  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
5703  035446  001406                          BEQ     2$              ;CHECK IF ZERO
5704  035450  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5705  035454  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
5706  035462  104003                          ERROR   3
```

```
5707  035464  016237  000002  001126  2$:     MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
5708  035472  001406                          BEQ     3$              ;CHECK IF ZERO
5709  035474  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5710  035500  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
5711  035506  104003                          ERROR   3
5712  035510  016237  000006  001126  3$:     MOV     RK)A(R2),$BDDAT ;STORE DISK ADD REG
5713  035516  001406                          BEQ     4$              ;CHECK IF ZERO
5714  035520  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5715  035524  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
5716  035532  104003                          ERROR   3
5717  035534  016237  000016  001126  4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
5718  035542  001406                          BEQ     5$              ;CHECK IF ZERO
5719  035544  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5720  035550  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
5721  035556  104003                          ERROR   3
5722  035560  016237  000010  001126  5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
5723  035566  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
5724  035574  001407                          BEQ     6$              ;YES,CONTINUE
5725  035576  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
5726  035604  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
5727  035612  104003                          ERROR   3
5728  035614  016237  000012  001126  6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
5729  035622  001406                          BEQ     7$              ;CHECK IF ZERO
5730  035624  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5731  035630  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
5732  035636  104003                          ERROR   3
5733  035640  016237  000014  001126  7$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
5734  035646  001406                          BEQ     8$              ;CHECK IF ZERO
5735  035650  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5736  035654  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
5737  035662  104003                          ERROR   3
5738  035664  016237  000020  001126  8$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
5739  035672  001406                          BEQ     10$             ;CHECK IF ZERO
5740  035674  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
5741  035700  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
5742  035706  104003                          ERROR   3
5743  035710  016237  000026  001126  10$:    MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
5744  035716  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
5745  035724  032737  020000  001126          BIT     #ECCW,$BDDAT
5746  035732  001403                          BEQ     11$
5747  035734  052737  020000  001124          BIS     #ECCW,$GDDAT
5748  035742  023737  001124  001126  11$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
5749  035750  001404                          BEQ     12$             ;YES,CONTINUE TEST
5750  035752  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
5751  035760  104003                          ERROR   3
5752  035762  016237  000032  001126  12$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5753  035770  001406                          BEQ     13$             ;CHECK IF ZERO
5754  035772  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5755  035776  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
5756  036004  104003                          ERROR   3
5757  036006  016237  000030  001126  13$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
5758  036014  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
5759  036022  001407                          BEQ     TST46           ;;YES,GO TO NEXT TEST
5760  036024  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
5761  036032  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
5762  036040  104003                          ERROR   3
```

```
5763
5764                                    ;*****************************************************************
5765                                    ;*TEST 46       REGISTER INTERACTION WITH DISK CYLINDER (PART 1)
5766                                    ;*
5767                                    ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5768                                    ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
5769                                    ;*      CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
5770                                    ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
5771                                    ;*
5772                                    ;*      000000  000010  000200  004000  100000
5773                                    ;*      000001  000020  000400  010000
5774                                    ;*      000002  000040  001000  020000
5775                                    ;*      000004  000100  002000  040000
5776                                    ;*
5777                                    ;*****************************************************************
5778  036042  000004                    TST46:  SCOPE
5779  036044  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
5780  036052  012701  000021                    MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
5781  036056  012737  000001  002010            MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
5782  036064  012737  064456  001320            MOV     #EM19,EM3N      ;LOAD ERROR MESSAGE
5783  036072  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5784  036100  012737  036106  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
5785                                                                    ;  SUBTEST LOOP
5786
5787  036106                            1$:
5788  036106  005062  000002                    CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
5789  036112  013762  002010  000020            MOV     CONFIG,RKDCYL(R2)       ;WRITE RKDCYL
5790  036120  016237  000020  001126            MOV     RKDCYL(R2),$BDDAT       ;STORE RKDCYL
5791  036126  013737  002010  001124            MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
5792  036134  042737  176000  001124            BIC     #176000,$GDDAT  ;INITIALIZE READ ONLY BITS
5793  036142  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
5794  036150  001404                            BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
5795  036152  012737  067272  001322            MOV     #EM1016,EM3N+2  ;LOAD ERROR MESSAGE
5796  036160  104003                            ERROR   3
5797  036162                            2$:
5798  036162  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
5799  036170  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
5800  036176  001407                            BEQ     3$              ;YES, CONTINUE
5801  036200  012737  000100  001124            MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
5802  036206  012737  067314  001322            MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
5803  036214  104003                            ERROR   3
5804  036216                            3$:
5805  036216  016237  000004  001126            MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
5806  036224  001406                            BEQ     4$              ;CHECK IF ZERO
5807  036226  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5808  036232  012737  067365  001322            MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
5809  036240  104003                            ERROR   3
5810  036242                            4$:
5811  036242  016237  000002  001126            MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
5812  036250  001406                            BEQ     5$              ;CHECK IF ZERO
5813  036252  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5814  036256  012737  067335  001322            MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
5815  036264  104003                            ERROR   3
5816  036266                            5$:
5817  036266  016237  000006  001126            MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
5818  036274  001406                            BEQ     6$              ;CHECK IF ZERO
```

```
5819  036276  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5820  036302  012737  067412  001322              MOV     #EM1020,EM3N+2      ;LOAD ERROR MESSAGE
5821  036310  104003                              ERROR   3
5822  036312                            6$:
5823  036312  016237  000016  001126              MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
5824  036320  001406                              BEQ     7$                  ;CHECK IF ZERO
5825  036322  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5826  036326  012737  067556  001322              MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
5827  036334  104003                              ERROR   3
5828  036336                            7$:
5829  036336  016237  000010  001128              MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
5830  036344  022737  000100  001126              CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
5831  036352  001407                              BEQ     8$                  ;YES,CONTINUE
5832  036354  012737  000100  001124              MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
5833  036362  012737  067450  001322              MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
5834  036370  104003                              ERROR   3
5835  036372                            8$:
5836  036372  016237  000012  001126              MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
5837  036400  001406                              BEQ     9$                  ;CHECK IF ZERO
5838  036402  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5839  036406  012737  067471  001322              MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
5840  036414  104003                              ERROR   3
5841  036416  016237  000014  001126  9$:         MOV     RKER(R2),$BDDAT ;STORE ERROR REG
5842  036424  001406                              BEQ     10$                 ;CHECK IF ZERO
5843  036426  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5844  036432  012737  067527  001322              MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
5845  036440  104003                              ERROR   3
5846  036442                            10$:
5847  036442  016237  000026  001126              MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
5848  036450  012737  002000  001124              MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
5849  036456  032737  020000  001126              BIT     #ECCW,$BDDAT
5850  036464  001403                              BEQ     13$
5851  036466  052737  020000  001124              BIS     #ECCW,$GDDAT
5852  036474  023737  001124  001126  13$:        CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
5853  036502  001404                              BEQ     14$                 ;YES,ISSUE CONTROLLER CLEAR
5854  036504  012737  067651  001322              MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
5855  036512  104003                              ERROR   3
5856  036514                            14$:
5857  036514  016237  000032  001126              MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5858  036522  001406                              BEQ     15$                 ;CHECK IF ZERO
5859  036524  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5860  036530  012737  067727  001322              MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
5861  036536  104003                              ERROR   3
5862  036540  016237  000030  001126  15$:        MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
5863  036546  012737  004066  001124  16$:        MOV     #4066,$GDDAT        ;USE 4066
5864  036554  023737  001124  001126  17$:        CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
5865  036562  001404                              BEQ     18$                 ;YES,INITIALIZE RK611
5866  036564  012737  067702  001322              MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
5867  036572  104003                              ERROR   3
5868  036574  016237  000020  002014  18$:        MOV     RKDCYL(R2),PREREG       ;GET PREVIOUS CONTENTS
5869  036602  012762  100000  000000              MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5870  036610  016237  000020  001126              MOV     RKDCYL(R2),$BDDAT       ;GET CURRENT VALUE
5871  036616  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
5872  036622  023737  001124  001126              CMP     $GDDAT,$BDDAT       ;CHECK IF RKDCYL CORRECT
5873  036630  001407                              BEQ     19$                 ;YES, CHECK IF FINISHED
5874  036632  012737  063403  001330              MOV     #EM3,EM4N           ;LOAD ERROR MESSAGE
```

```
 5875  036640  012737  067272  001332          MOV     #EM1016,EM4N+2
 5876  036646  104004                          ERROR   4
 5877  036650  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
 5878  036652  000241                          CLC                     ;SHIFT IN ZERO
 5879  036654  006137  002010                  ROL     CONFIG
 5880  036660  005301                          DEC     R1              ;CHECK IF FINISHED
 5881  036662  001402                          BEQ     TST47           ;;YES, GO ON TO NEXT TEST
 5882  036664  000137  036106                  JMP     1$
 5883
 5884                                  ;;*****************************************************************
 5885                                  ;*TEST 47       REGISTER INTERACTION WITH DISK CYLINDER (PART 2)
 5886                                  ;*
 5887                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
 5888                                  ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
 5889                                  ;*      CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
 5890                                  ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
 5891                                  ;*
 5892                                  ;*      177777  177767  177577  173777  077777
 5893                                  ;*      177776  177757  177377  167777
 5894                                  ;*      177775  177737  176777  157777
 5895                                  ;*      177773  177677  175777  137777
 5896                                  ;*
 5897                                  ;*****************************************************************
 5898  036670  000004                  TST47:  SCOPE
 5899  036672  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
 5900  036700  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
 5901  036704  012737  177776  002010          MOV     #177776,CONFIG  ;LOAD INITIAL CONFIGURATION
 5902  036712  012737  064456  001320          MOV     #EM19,EM3N      ;LOAD ERROR MESSAGE
 5903  036720  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
 5904  036726  012737  036734  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
 5905                                                                  ;  SUBTEST LOOP
 5906
 5907  036734                          1$:
 5908  036734  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
 5909  036740  013762  002010  000020          MOV     CONFIG,RKDCYL(R2)       ;WRITE RKDCYL
 5910  036746  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT       ;STORE RKDCYL
 5911  036754  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
 5912  036762  042737  176000  001124          BIC     #176000,$GDDAT  ;INITIALIZE READ ONLY BITS
 5913  036770  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
 5914  036776  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
 5915  037000  012737  067272  001322          MOV     #EM1016,EM3N+2  ;LOAD ERROR MESSAGE
 5916  037006  104003                          ERROR   3
 5917  037010                          2$:
 5918  037010  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
 5919  037016  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
 5920  037024  001407                          BEQ     3$              ;YES, CONTINUE
 5921  037026  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
 5922  037034  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
 5923  037042  104003                          ERROR   3
 5924  037044                          3$:
 5925  037044  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
 5926  037052  001406                          BEQ     4$              ;CHECK IF ZERO
 5927  037054  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 5928  037060  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
 5929  037066  104003                          ERROR   3
 5930  037070                          4$:
```

```
5931  037070  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
5932  037076  001406                          BEQ     5$              ;CHECK IF ZERO
5933  037100  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5934  037104  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
5935  037112  104003                          ERROR   3
5936  037114                          5$:
5937  037114  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
5938  037122  001406                          BEQ     6$              ;CHECK IF ZERO
5939  037124  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5940  037130  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
5941  037136  104003                          ERROR   3
5942  037140                          6$:
5943  037140  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
5944  037146  001406                          BEQ     7$              ;CHECK IF ZERO
5945  037150  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5946  037154  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
5947  037162  104003                          ERROR   3
5948  037164                          7$:
5949  037164  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
5950  037172  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
5951  037200  001407                          BEQ     8$              ;YES,CONTINUE
5952  037202  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
5953  037210  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
5954  037216  104003                          ERROR   3
5955  037220                          8$:
5956  037220  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
5957  037226  001406                          BEQ     9$              ;CHECK IF ZERO
5958  037230  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5959  037234  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
5960  037242  104003                          ERROR   3
5961  037244  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
5962  037252  001406                          BEQ     10$             ;CHECK IF ZERO
5963  037254  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5964  037260  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
5965  037266  104003                          ERROR   3
5966  037270                          10$:
5967  037270  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
5968  037276  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
5969  037304  032737  020000  001126          BIT     #ECCW,$BDDAT
5970  037312  001403                          BEQ     13$
5971  037314  052737  020000  001124          BIS     #ECCW,$GDDAT
5972  037322  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
5973  037330  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
5974  037332  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
5975  037340  104003                          ERROR   3
5976  037342                          14$:
5977  037342  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5978  037350  001406                          BEQ     15$             ;CHECK IF ZERO
5979  037352  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5980  037356  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
5981  037364  104003                          ERROR   3
5982  037366  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
5983  037374  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
5984  037402  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
5985  037410  001404                          BEQ     18$             ;YES,INITIALIZE RK611
5986  037412  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
```

```
                                        ERROR   3
5988  037422  016237  000020  002014    18$:    MOV     RKDCYL(R2),PREREG       ;GET PREVIOUS CONTENTS
5989  037430  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5990  037436  016237  000020  001126            MOV     RKDCYL(R2),$BDDAT       ;GET CURRENT VALUE
5991  037444  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
5992  037450  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
5993  037456  001407                            BEQ     19$             ;YES, CHECK IF FINISHED
5994  037460  012737  063403  001330            MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
5995  037466  012737  067272  001332            MOV     #EM1016,EM4N+2
5996  037474  104004                            ERROR   4
5997  037476  104415                    19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
5998  037500  000261                            SEC                     ;SHIFT IN ONE
5999  037502  006137  002010                    ROL     CONFIG
6000  037506  005301                            DEC     R1              ;CHECK IF FINISHED
6001  037510  001402                            BEQ     TST50           ;;YES, GO ON TO NEXT TEST
6002  037512  000137  036734                    JMP     1$
6003
6004                                    ;;*****************************************************************
6005                                    ;*TEST 50       REGISTER INTERACTION WITH DISK CYLINDER (PART 3)
6006                                    ;*
6007                                    ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
6008                                    ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
6009                                    ;*      CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
6010                                    ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
6011                                    ;*
6012                                    ;*      000001  000037  000777  017777  000000
6013                                    ;*      000003  000077  001777  037777
6014                                    ;*      000007  000177  003777  077777
6015                                    ;*      000017  000377  007777  177777
6016                                    ;*
6017                                    ;;*****************************************************************
6018  037516  000004                    TST50:  SCOPE
6019  037520  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
6020  037526  012701  000021                    MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
6021  037532  005037  002010                    CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
6022  037536  012737  064456  001320            MOV     #EM19,EM3N      ;LOAD ERROR MESSAGE
6023  037544  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
6024  037552  012737  037560  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
6025                                                                    ;  SUBTEST LOOP
6026
6027  037560                            1$:
6028  037560  005062  000002                    CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
6029  037564  013762  002010  000020            MOV     CONFIG,RKDCYL(R2)       ;WRITE RKDCYL
6030  037572  016237  000020  001126            MOV     RKDCYL(R2),$BDDAT       ;STORE RKDCYL
6031  037600  013737  002010  001124            MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
6032  037606  042737  176000  001124            BIC     #176000,$GDDAT  ;INITIALIZE READ ONLY BITS
6033  037614  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
6034  037622  001404                            BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
6035  037624  012737  067272  001322            MOV     #EM1016,EM3N+2  ;LOAD ERROR MESSAGE
6036  037632  104003                            ERROR   3
6037  037634                            2$:
6038  037634  016237  000000  001126            MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
6039  037642  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
6040  037650  001407                            BEQ     3$              ;YES, CONTINUE
6041  037652  012737  000100  001124            MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
6042  037660  012737  067314  001322            MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
```

```
6043  037666  104003                          ERROR   3
6044  037670                          3$:
6045  037670  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
6046  037676  001406                          BEQ     4$              ;CHECK IF ZERO
6047  037700  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6048  037704  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
6049  037712  104003                          ERROR   3
6050  037714                          4$:
6051  037714  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
6052  037722  001406                          BEQ     5$              ;CHECK IF ZERO
6053  037724  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6054  037730  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
6055  037736  104003                          ERROR   3
6056  037740                          5$:
6057  037740  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
6058  037746  001406                          BEQ     6$              ;CHECK IF ZERO
6059  037750  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6060  037754  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
6061  037762  104003                          ERROR   3
6062  037764                          6$:
6063  037764  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
6064  037772  001406                          BEQ     7$              ;CHECK IF ZERO
6065  037774  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6066  040000  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
6067  040006  104003                          ERROR   3
6068  040010                          7$:
6069  040010  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
6070  040016  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
6071  040024  001407                          BEQ     8$              ;YES,CONTINUE
6072  040026  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
6073  040034  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
6074  040042  104003                          ERROR   3
6075  040044                          8$:
6076  040044  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
6077  040052  001406                          BEQ     9$              ;CHECK IF ZERO
6078  040054  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6079  040060  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
6080  040066  104003                          ERROR   3
6081  040070  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
6082  040076  001406                          BEQ     10$             ;CHECK IF ZERO
6083  040100  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6084  040104  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
6085  040112  104003                          ERROR   3
6086  040114                          10$:
6087  040114  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
6088  040122  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
6089  040130  032737  020000  001126          BIT     #ECCW,$BDDAT
6090  040136  001403                          BEQ     13$
6091  040140  052737  020000  001124          BIS     #ECCW,$GDDAT
6092  040146  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
6093  040154  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
6094  040156  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
6095  040164  104003                          ERROR   3
6096  040166                          14$:
6097  040166  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
6098  040174  001406                          BEQ     15$             ;CHECK IF ZERO
```

```
                                        CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
                                        MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
                                        ERROR   3
040212  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
040220  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
040226  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
040234  001404                          BEQ     18$             ;YES,INITIALIZE RK611
040236  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
040244  104003                          ERROR   3
040246  016237  000020  002014  18$.    MOV     RKDCYL(R2),PREREG       ;GET PREVIOUS CONTENTS
040254  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
040262  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT       ;GET CURRENT VALUE
040270  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
040274  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
040302  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
040304  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
040312  012737  067272  001332          MOV     #EM1016,EM4N+2
040320  104004                          ERROR   4
040322  104415                  19$:    SCOPE                   ;CHECK IF LOOP ON ERROR
040324  000261                          SEC                     ;SHIFT IN ONE
040326  006137  002010                  ROL     CONFIG
040332  005301                          DEC     R1              ;CHECK IF FINISHED
040334  001402                          BEQ     TST51           ;;YES, GO ON TO NEXT TEST
040336  000137  037560                  JMP     1$

;;*****************************************************************
;*TEST 51        REGISTER INTERACTION WITH DISK CYLINDER (PART 4)
;*
;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
;*      CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
;*
;*      100000  174000  177600  177770  000000
;*      140000  176000  177700  177774
;*      160000  177000  177740  177776
;*      170000  177400  177760  177777
;*
;;*****************************************************************
040342  000004          TST51:  SCOPE
040344  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
040352  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
040356  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
040362  012737  064456  001320          MOV     #EM19,EM3N      ;LOAD ERROR MESSAGE
040370  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
040376  012737  040404  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
040404                          1$:
040404  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
040410  013762  002010  000020          MOV     CONFIG,RKDCYL(R2)       ;WRITE RKDCYL
040416  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT       ;STORE RKDCYL
040424  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
040432  042737  176000  001124          BIC     #176000,$GDDAT  ;INITIALIZE READ ONLY BITS
040440  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
040446  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
```

```
 6155  040450  012737  067272  001322          MOV     #EM1016,EM3N+2  ;LOAD ERROR MESSAGE
 6156  040456  104003                          ERROR   3
 6157  040460                          2$:
 6158  040460  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
 6159  040466  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
 6160  040474  001407                          BEQ     3$              ;YES, CONTINUE
 6161  040476  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
 6162  040504  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
 6163  040512  104003                          ERROR   3
 6164  040514                          3$:
 6165  040514  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
 6166  040522  001406                          BEQ     4$              ;CHECK IF ZERO
 6167  040524  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6168  040530  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
 6169  040536  104003                          ERROR   3
 6170  040540                          4$:
 6171  040540  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
 6172  040546  001406                          BEQ     5$              ;CHECK IF ZERO
 6173  040550  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6174  040554  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
 6175  040562  104003                          ERROR   3
 6176  040564                          5$:
 6177  040564  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
 6178  040572  001406                          BEQ     6$              ;CHECK IF ZERO
 6179  040574  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6180  040600  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
 6181  040606  104003                          ERROR   3
 6182  040610                          6$:
 6183  040610  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
 6184  040616  001406                          BEQ     7$              ;CHECK IF ZERO
 6185  040620  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6186  040624  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
 6187  040632  104003                          ERROR   3
 6188  040634                          7$:
 6189  040634  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
 6190  040642  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
 6191  040650  001407                          BEQ     8$              ;YES,CONTINUE
 6192  040652  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
 6193  040660  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
 6194  040666  104003                          ERROR   3
 6195  040670                          8$:
 6196  040670  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
 6197  040676  001406                          BEQ     9$              ;CHECK IF ZERO
 6198  040700  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6199  040704  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 6200  040712  104003                          ERROR   3
 6201  040714  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
 6202  040722  001406                          BEQ     10$             ;CHECK IF ZERO
 6203  040724  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6204  040730  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
 6205  040736  104003                          ERROR   3
 6206  040740                          10$:
 6207  040740  016237  000026  001126          MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
 6208  040746  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
 6209  040754  032737  020000  001126          BIT     #ECCW,$BDDAT
 6210  040762  001403                          BEQ     13$
```

```
6211  040764  052737  020000  001124          BIS     #ECCW,$GDDAT
6212  040772  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
6213  041000  001404                          BEQ     14$             ;YES,ISSUE CONTROLLER CLEAR
6214  041002  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
6215  041010  104003                          ERROR   3
6216  041012                          14$:
6217  041012  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
6218  041020  001406                          BEQ     15$             ;CHECK IF ZERO
6219  041022  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6220  041026  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
6221  041034  104003                          ERROR   3
6222  041036  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
6223  041044  012737  004066  001124  16$:    MOV     #4066,$GDDAT    ;USE 4066
6224  041052  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
6225  041060  001404                          BEQ     18$             ;YES,INITIALIZE RK611
6226  041062  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
6227  041070  104003                          ERROR   3
6228  041072  016237  000020  002014  18$:    MOV     RKDCYL(R2),PREREG       ;GET PREVIOUS CONTENTS
6229  041100  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
6230  041106  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT       ;GET CURRENT VALUE
6231  041114  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6232  041120  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKDCYL CORRECT
6233  041126  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
6234  041130  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
6235  041136  012737  067272  001332          MOV     #EM1016,EM4N+2
6236  041144  104004                          ERROR   4
6237  041146  104415                  19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
6238  041150  000261                          SEC                     ;SHIFT IN ONE
6239  041152  006037  002010                  ROR     CONFIG
6240  041156  005301                          DEC     R1              ;CHECK IF FINISHED
6241  041160  001402                          BEQ     TST52           ;;YES, GO ON TO NEXT TEST
6242  041162  000137  040404                  JMP     1$
6243
6244                                  ;;*****************************************************************
6245                                  ;*TEST 52       REGISTER INTERACTION USING MAINT REG 1 (PART 1)
6246                                  ;*
6247                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
6248                                  ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
6249                                  ;*      REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
6250                                  ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
6251                                  ;*
6252                                  ;*      000000  000010  000200  004000  100000
6253                                  ;*      000001  000020  000400  010000
6254                                  ;*      000002  000040  001000  020000
6255                                  ;*      000004  000100  002000  040000
6256                                  ;*
6257                                  ;*****************************************************************
6258  041166  000004                  TST52:  SCOPE
6259  041170  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
6260  041176  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
6261  041202  012737  000001  002010          MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
6262  041210  012737  064517  001320          MOV     #EM20,EM3N      ;LOAD ERROR MESSAGE
6263  041216  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
6264  041224  012737  041232  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
6265                                                                  ;  SUBTEST LOOP
6266
```

```
6267  041232                          1$:
6268  041232  005062  000002                  CLR     RKWC(R2)            ;CLEAR WORD COUNT REG.
6269  041236  013762  002010  000026          MOV     CONFIG,RKMR1(R2)            ;WRITE RKMR1
6270  041244  016237  000026  001126          MOV     RKMR1(R2),$BDDAT            ;STORE RKMR1
6271  041252  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
6272  041260  042737  174000  001124          BIC     #PCA!PCD!ECCW!WRTGAT!RDGATE,$GDDAT ;INITIALIZE READ ONLY BITS
6273  041266  052737  002000  001124          BIS     #MEWD,$GDDAT
6274  041274  032737  020000  001126          BIT     #ECCW,$BDDAT
6275  041302  001403                          BEQ     30$
6276  041304  052737  020000  001124          BIS     #ECCW,$GDDAT
6277  041312                          30$:
6278  041312  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
6279  041320  001404                          BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
6280  041322  012737  067072  001322          MOV     #EM1009,EM3N+2  ;LOAD ERROR MESSAGE
6281  041330  104003                          ERROR   3
6282  041332                          2$:
6283  041332  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
6284  041340  022737  000200  001126          CMP     #RDY,$BDDAT             ;CHECK IF CS1 CORRECT
6285  041346  001407                          BEQ     3$                      ;YES, CONTINUE
6286  041350  012737  000100  001124          MOV     #IR,$GDDAT              ;LOAD EXPECTED RESULTS
6287  041356  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
6288  041364  104003                          ERROR   3
6289  041366                          3$:
6290  041366  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
6291  041374  001406                          BEQ     4$              ;CHECK IF ZERO
6292  041376  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6293  041402  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
6294  041410  104003                          ERROR   3
6295  041412                          4$:
6296  041412  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
6297  041420  001406                          BEQ     5$              ;CHECK IF ZERO
6298  041422  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6299  041426  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
6300  041434  104003                          ERROR   3
6301  041436                          5$:
6302  041436  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
6303  041444  001406                          BEQ     6$              ;CHECK IF ZERO
6304  041446  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6305  041452  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
6306  041460  104003                          ERROR   3
6307  041462                          6$:
6308  041462  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
6309  041470  001406                          BEQ     7$              ;CHECK IF ZERO
6310  041472  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6311  041476  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
6312  041504  104003                          ERROR   3
6313  041506                          7$:
6314  041506  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
6315  041514  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
6316  041522  001407                          BEQ     8$              ;YES,CONTINUE
6317  041524  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
6318  041532  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
6319  041540  104003                          ERROR   3
6320  041542                          8$:
6321  041542  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
6322  041550  001406                          BEQ     9$              ;CHECK IF ZERO
```

```
 6323  041552  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6324  041556  012737  067471  001322            MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 6325  041564  104003                            ERROR   3
 6326  041566  016237  000014  001126  9$:       MOV     RKER(R2),$BDDAT ;STORE ERROR REG
 6327  041574  001406                            BEQ     10$             ;CHECK IF ZERO
 6328  041576  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6329  041602  012737  067527  001322            MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
 6330  041610  104003                            ERROR   3
 6331  041612                          10$:
 6332  041612  016237  000020  001126            MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
 6333  041620  001406                            BEQ     12$             ;CHECK IF ZERO
 6334  041622  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6335  041626  012737  067624  001322            MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
 6336  041634  104003                            ERROR   3
 6337  041636                          12$:
 6338  041636  016237  000032  001126            MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
 6339  041644  001406                            BEQ     15$             ;CHECK IF ZERO
 6340  041646  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6341  041652  012737  067727  001322            MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
 6342  041660  104003                            ERROR   3
 6343  041662  016237  000030  001126  15$:      MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
 6344  041670  012737  004066  001124  16$:      MOV     #4066,$GDDAT    ;USE 4066
 6345  041676  023737  001124  001126  17$:      CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
 6346  041704  001404                            BEQ     18$             ;YES,INITIALIZE RK611
 6347  041706  012737  067702  001322            MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
 6348  041714  104003                            ERROR   3
 6349  041716  016237  000026  002014  18$:      MOV     RKMR1(R2),PREREG        ;GET PREVIOUS CONTENTS
 6350  041724  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
 6351  041732  016237  000026  001126            MOV     RKMR1(R2),$BDDAT        ;GET CURRENT VALUE
 6352  041740  012737  002000  001124            MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
 6353  041746  032737  020000  001126            BIT     #ECCW,$BDDAT
 6354  041754  001403                            BEQ     35$
 6355  041756  052737  020000  001124            BIS     #ECCW,$GDDAT
 6356  041764                          35$:
 6357  041764  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
 6358  041772  001407                            BEQ     19$             ;YES, CHECK IF FINISHED
 6359  041774  012737  063403  001330            MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
 6360  042002  012737  067072  001332            MOV     #EM1009,EM4N+2
 6361  042010  104004                            ERROR   4
 6362  042012  104415                  19$:      SCOP1                   ;CHECK IF LOOP ON ERROR
 6363  042014  000241                            CLC                     ;SHIFT IN ZERO
 6364  042016  006137  002010                    ROL     CONFIG
 6365  042022  005301                            DEC     R1              ;CHECK IF FINISHED
 6366  042024  001402                            BEQ     TST53           ;;YES, GO ON TO NEXT TEST
 6367  042026  000137  041232                    JMP     1$
 6368
 6369                                    ;;*****************************************************************
 6370                                    ;*TEST 53       REGISTER INTERACTION USING MAINT REG 1 (PART 2)
 6371                                    ;*
 6372                                    ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
 6373                                    ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
 6374                                    ;*      REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
 6375                                    ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
 6376                                    ;*
 6377                                    ;*      177777  177767  177577  173777  077777
 6378                                    ;*      177776  177757  177377  167777
```

```
                                        ;*      177775  177737  176777  157777
                                        ;*      177773  177677  175777  137777
                                        ;*
                                        ;;*****************************************************************
042032  000004                          TST53:  SCOPE
042034  012737  000764  001200                  MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
042042  012701  000021                          MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
042046  012737  177776  002010                  MOV     #177776,CONFIG  ;LOAD INITIAL CONFIGURATION
042054  012737  064517  001320                  MOV     #EM20,EM3N      ;LOAD ERROR MESSAGE
042062  012762  100000  000000                  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
042070  012737  042076  001110                  MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                        ;  SUBTEST LOOP
042076                                  1$:
042076  005062  000002                          CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
042102  013762  002010  000026                  MOV     CONFIG,RKMR1(R2)        ;WRITE RKMR1
042110  016237  000026  001126                  MOV     RKMR1(R2),$BDDAT        ;STORE RKMR1
042116  013737  002010  001124                  MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
042124  042737  174000  001124                  BIC     #PCA!PCD!ECCW!WRTGAT!RDGATE,$GDDAT ;INITIALIZE READ ONLY BITS
042132  052737  002000  001124                  BIS     #MEWD,$GDDAT
042140  032737  020000  001126                  BIT     #ECCW,$BDDAT
042146  001403                                  BEQ     30$
042150  052737  020000  001124                  BIS     #ECCW,$GDDAT
042156                                  30$:
042156  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
042164  001404                                  BEQ     2$              ;YES, TEST IF ANY OTHER REG MODIFIED
042166  012737  067072  001322                  MOV     #EM1009,EM3N+2  ;LOAD ERROR MESSAGE
042174  104003                                  ERROR   3
042176                                  2$:
042176  016237  000000  001126                  MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
042204  022737  000200  001126                  CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
042212  001407                                  BEQ     3$              ;YES, CONTINUE
042214  012737  000100  001124                  MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
042222  012737  067314  001322                  MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
042230  104003                                  ERROR   3
042232                                  3$:
042232  016237  000004  001126                  MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
042240  001406                                  BEQ     4$              ;CHECK IF ZERO
042242  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
042246  012737  067365  001322                  MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
042254  104003                                  ERROR   3
042256                                  4$:
042256  016237  000002  001126                  MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
042264  001406                                  BEQ     5$              ;CHECK IF ZERO
042266  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
042272  012737  067335  001322                  MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
042300  104003                                  ERROR   3
042302                                  5$:
042302  016237  000006  001126                  MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
042310  001406                                  BEQ     6$              ;CHECK IF ZERO
042312  005037  001124                          CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
042316  012737  067412  001322                  MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
042324  104003                                  ERROR   3
042326                                  6$:
042326  016237  000016  001126                  MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY OFFSET REG.
042334  001406                                  BEQ     7$              ;CHECK IF ZERO
```

```
 6435  042336  005037  001174                        CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 6436  042342  012737  067516  001322                MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
 6437  042350  104003                                ERROR   3
 6438  042352                                7$:
 6439  042352  016237  000010  001126                MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
 6440  042360  022737  000100  001126                CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
 6441  042366  001407                                BEQ     8$                  ;YES,CONTINUE
 6442  042370  012737  000100  001124                MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
 6443  042376  012737  067450  001322                MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
 6444  042404  104003                                ERROR   3
 6445  042406                                8$:
 6446  042406  016237  000012  001126                MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
 6447  042414  001406                                BEQ     9$                  ;CHECK IF ZERO
 6448  042416  005037  001124                        CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 6449  042422  012737  067471  001322                MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
 6450  042430  104003                                ERROR   3
 6451  042432  016237  000014  001126  9$:           MOV     RKER(R2),$BDDAT     ;STORE ERROR REG
 6452  042440  001406                                BEQ     10$                 ;CHECK IF ZERO
 6453  042442  005037  001124                        CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 6454  042446  012737  067527  001322                MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
 6455  042454  104003                                ERROR   3
 6456  042456                                10$:
 6457  042456  016237  000020  001126                MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG
 6458  042464  001406                                BEQ     12$                 ;CHECK IF ZERO
 6459  042466  005037  001124                        CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 6460  042472  012737  067624  001322                MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
 6461  042500  104003                                ERROR   3
 6462  042502                                12$:
 6463  042502  016237  000032  001126                MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
 6464  042510  001406                                BEQ     15$                 ;CHECK IF ZERO
 6465  042512  005037  001124                        CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
 6466  042516  012737  067727  001322                MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
 6467  042524  104003                                ERROR   3
 6468  042526  016237  000030  001126  15$:          MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
 6469  042534  012737  004066  001124  16$:          MOV     #4066,$GDDAT        ;USE 4066
 6470  042542  023737  001124  001126  17$:          CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
 6471  042550  001404                                BEQ     18$                 ;YES,INITIALIZE RK611
 6472  042552  012737  067702  001322                MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
 6473  042560  104003                                ERROR   3
 6474  042562  016237  000026  002014  18$:          MOV     RKMR1(R2),PREREG            ;GET PREVIOUS CONTENTS
 6475  042570  012762  100000  000000                MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
 6476  042576  016237  000026  001126                MOV     RKMR1(R2),$BDDAT            ;GET CURRENT VALUE
 6477  042604  012737  002000  001124                MOV     #MEWD,$GDDAT        ;LOAD EXPECTED CONTENTS
 6478  042612  032737  020000  001126                BIT     #ECCW,$BDDAT
 6479  042620  001403                                BEQ     35$
 6480  042622  052737  020000  001124                BIS     #ECCW,$GDDAT
 6481  042630                                35$:
 6482  042630  023737  001124  001126                CMP     $GDDAT,$BDDAT       ;CHECK IF RKMR1 CORRECT
 6483  042636  001407                                BEQ     19$                 ;YES, CHECK IF FINISHED
 6484  042640  012737  063403  001330                MOV     #EM3,EM4N           ;LOAD ERROR MESSAGE
 6485  042646  012737  067072  001332                MOV     #EM1009,EM4N+2
 6486  042654  104004                                ERROR   4
 6487  042656  104415                        19$:    SCOP1                       ;CHECK IF LOOP ON ERROR
 6488  042660  000261                                SEC                         ;SHIFT IN ONE
 6489  042662  006137  002010                        ROL     CONFIG
 6490  042666  005301                                DEC     R1                  ;CHECK IF FINISHED
```

```
6491  042670  001402                        BEQ     TST54           ;;YES, GO ON TO NEXT TEST
6492  042672  000137  042076                JMP     1$
6493
6494                                  ;;*****************************************************************
6495                                  ;*TEST 54       REGISTER INTERACTION USING MAINT REG 1 (PART 3)
6496                                  ;*
6497                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
6498                                  ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
6499                                  ;*      REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
6500                                  ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
6501                                  ;*
6502                                  ;*      000001  000037  000777  017777  000000
6503                                  ;*      000003  000077  001777  037777
6504                                  ;*      000007  000177  003777  077777
6505                                  ;*      000017  000377  007777  177777
6506                                  ;*
6507                                  ;*****************************************************************
6508  042676  000004                TST54:  SCOPE
6509  042700  012737  000764  001200        MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
6510  042706  012701  000021                MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
6511  042712  005037  002010                CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
6512  042716  012737  064517  001320        MOV     #EM20,EM3N      ;LOAD ERROR MESSAGE
6513  042724  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
6514  042732  012737  042740  001110        MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
6515                                                                ;   SUBTEST LOOP
6516
6517  042740                        1$:
6518  042740  005062  000002                CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
6519  042744  013762  002010  000026        MOV     CONFIG,RKMR1(R2)        ;WRITE RKMR1
6520  042752  016237  000026  001126        MOV     RKMR1(R2),$BDDAT        ;STORE RKMR1
6521  042760  013737  002010  001124        MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
6522  042766  042737  174000  001124        BIC     #PCA!PCD!ECCW!WRTGAT!RDGATE,$GDDAT ;INITIALIZE READ ONLY BITS
6523  042774  052737  002000  001124        BIS     #MEWD,$GDDAT
6524  043002  032737  020000  001126        BIT     #ECCW,$BDDAT
6525  043010  001403                        BEQ     30$
6526  043012  052737  020000  001124        BIS     #ECCW,$GDDAT
6527  043020                        30$:
6528  043020  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
6529  043026  001404                        BEQ     2$              ;YES,TEST IF ANY OTHER REG MODIFIED
6530  043030  012737  067072  001322        MOV     #EM1009,EM3N+2  ;LOAD ERROR MESSAGE
6531  043036  104003                        ERROR   3
6532  043040                        2$:
6533  043040  016237  000000  001126        MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
6534  043046  022737  000200  001126        CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
6535  043054  001407                        BEQ     3$              ;YES, CONTINUE
6536  043056  012737  000100  001124        MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
6537  043064  012737  067314  001322        MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
6538  043072  104003                        ERROR   3
6539  043074                        3$:
6540  043074  016237  000004  001126        MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
6541  043102  001406                        BEQ     4$              ;CHECK IF ZERO
6542  043104  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6543  043110  012737  067365  001322        MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
6544  043116  104003                        ERROR   3
6545  043120                        4$:
6546  043120  016237  000002  001126        MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
```

```
6547  043126  001406                          BEQ     5$                  ;CHECK IF ZERO
6548  043130  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6549  043134  012737  067335  001322          MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
6550  043142  104003                          ERROR   3
6551  043144                          5$:
6552  043144  016237  000006  001126          MOV     RKDA(R2),$BDDAT     ;STORE DISK AVERAGE REG
6553  043152  001406                          BEQ     6$                  ;CHECK IF ZERO
6554  043154  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6555  043160  012737  067412  001322          MOV     #EM1020,EM3N+2      ;LOAD ERROR MESSAGE
6556  043166  104003                          ERROR   3
6557  043170                          6$:
6558  043170  016237  000016  001126          MOV     RKASOF(R2),$BDDAT   ;STORE ATTENTION SUMMARY/OFFSET REG.
6559  043176  001406                          BEQ     7$                  ;CHECK IF ZERO
6560  043200  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6561  043204  012737  067556  001322          MOV     #EM1024,EM3N+2      ;LOAD ERROR MESSAGE
6562  043212  104003                          ERROR   3
6563  043214                          7$:
6564  043214  016237  000010  001126          MOV     RKCS2(R2),$BDDAT    ;STORE COMMAND AND STATUS REG.2
6565  043222  022737  000100  001126          CMP     #IR,$BDDAT          ;CHECK IF CS2 CORRECT
6566  043230  001407                          BEQ     8$                  ;YES,CONTINUE
6567  043232  012737  000100  001124          MOV     #IR,$GDDAT          ;LOAD EXPECTED CONTENTS
6568  043240  012737  067450  001322          MOV     #EM1021,EM3N+2      ;LOAD ERROR MESSAGE
6569  043246  104003                          ERROR   3
6570  043250                          8$:
6571  043250  016237  000012  001126          MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG
6572  043256  001406                          BEQ     9$                  ;CHECK IF ZERO
6573  043260  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6574  043264  012737  067471  001322          MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
6575  043272  104003                          ERROR   3
6576  043274  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT     ;STORE ERROR REG
6577  043302  001406                          BEQ     10$                 ;CHECK IF ZERO
6578  043304  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6579  043310  012737  067527  001322          MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
6580  043316  104003                          ERROR   3
6581  043320                          10$:
6582  043320  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADD REG
6583  043326  001406                          BEQ     12$                 ;CHECK IF ZERO
6584  043330  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6585  043334  012737  067624  001322          MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
6586  043342  104003                          ERROR   3
6587  043344                          12$:
6588  043344  016237  000032  001126          MOV     RKECPT(R2),$BDDAT   ;STORE ECC PATTERN REG.
6589  043352  001406                          BEQ     15$                 ;CHECK IF ZERO
6590  043354  005037  001124                  CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
6591  043360  012737  067727  001322          MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
6592  043366  104003                          ERROR   3
6593  043370  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT   ;STORE ECC POSITION REG.
6594  043376  012737  004066  001124  16$:    MOV     #4066,$GDDAT        ;USE 4066
6595  043404  023737  001124  001126  17$:    CMP     $GDDAT,$BDDAT       ;CHECK IF ECC POSITION CORRECT
6596  043412  001404                          BEQ     18$                 ;YES,INITIALIZE RK611
6597  043414  012737  067770  001322          MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
6598  043422  104003                          ERROR   3
6599  043424  016237  000026  002014  18$:    MOV     RKMR1(R2),PREREG            ;GET PREVIOUS CONTENTS
6600  043432  012762  100000  000000          MOV     #CCLR,RKCS1(R2)     ;CLEAR RK611 CONTROLLER
6601  043440  016237  000026  001126          MOV     RKMR1(R2),$BDDAT            ;GET CURRENT VALUE
6602  043446  012737  002000  001124          MOV     #MEWD,$GDDAT        ;LOAD EXPECTED CONTENTS
```

```
6603  043454  032737  020000  001126          BIT     #ECCW,$BDDAT
6604  043462  001403                          BEQ     35$
6605  043464  052737  020000  001124          BIS     #ECCW,$GDDAT
6606  043472                          35$:
6607  043472  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
6608  043500  001407                          BEQ     19$             ;YES, CHECK IF FINISHED
6609  043502  012737  063403  001330          MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
6610  043510  012737  067072  001332          MOV     #EM1009,EM4N+2
6611  043516  104004                          ERROR   4
6612  043520  104415                  19$:    SCOPI                   ;CHECK IF LOOP ON ERROR
6613  043522  000261                          SEC                     ;SHIFT IN ONE
6614  043524  006137  002010                  ROL     CONFIG
6615  043530  005301                          DEC     R1              ;CHECK IF FINISHED
6616  043532  001402                          BEQ     TST55           ;;YES, GO ON TO NEXT TEST
6617  043534  000137  042740                  JMP     1$
6618
6619                                  ;;*****************************************************************
6620                                  ;*TEST 55        REGISTER INTERACTION USING MAINT REG 1 (PART 4)
6621                                  ;*
6622                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
6623                                  ;*      WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
6624                                  ;*      REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
6625                                  ;*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
6626                                  ;*
6627                                  ;*      100000  174000  177600  177770  000000
6628                                  ;*      140000  176000  177700  177774
6629                                  ;*      160000  177000  177740  177776
6630                                  ;*      170000  177400  177760  177777
6631                                  ;*
6632                                  ;*****************************************************************
6633  043540  000004                  TST55:  SCOPE
6634  043542  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
6635  043550  012701  000021                  MOV     #17.,R1         ;LOAD NUMBER OF PATTERNS
6636  043554  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
6637  043560  012737  064517  001320          MOV     #EM20,EM3N      ;LOAD ERROR MESSAGE
6638  043566  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
6639  043574  012737  043602  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
6640                                                                  ;  SUBTEST LOOP
6641
6642  043602                          1$:
6643  043602  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
6644  043606  013762  002010  000026          MOV     CONFIG,RKMR1(R2)        ;WRITE RKMR1
6645  043614  016237  000026  001126          MOV     RKMR1(R2),$BDDAT        ;STORE RKMR1
6646  043622  013737  002010  001124          MOV     CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
6647  043630  042737  174000  001124          BIC     #PCA!PCD!ECCW!WRTGAT!RDGATE,$GDDAT ;INITIALIZE READ ONLY BITS
6648  043636  052737  002000  001124          BIS     #MEWD,$GDDAT
6649  043644  032737  020000  001126          BIT     #ECCW,$BDDAT
6650  043652  001403                          BEQ     30$
6651  043654  052737  020000  001124          BIS     #ECCW,$GDDAT
6652  043662                          30$:
6653  043662  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
6654  043670  001404                          BEQ     2$              ;YES, TEST IF ANY OTHER REG MODIFIED
6655  043672  012737  067072  001322          MOV     #EM1009,EM3N+2  ;LOAD ERROR MESSAGE
6656  043700  104003                          ERROR   3
6657  043702                          2$:
6658  043702  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
```

```
6659  043710  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
6660  043716  001407                          BEQ     3$              ;YES, CONTINUE
6661  043720  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED RESULTS
6662  043726  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
6663  043734  104003                          ERROR   3
6664  043736                          3$:
6665  043736  016237  000004  001126          MOV     RKBA(R2),$BDDAT ;STORE BUS AND REG
6666  043744  001406                          BEQ     4$              ;CHECK IF ZERO
6667  043746  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6668  043752  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
6669  043760  104003                          ERROR   3
6670  043762                          4$:
6671  043762  016237  000002  001126          MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG
6672  043770  001406                          BEQ     5$              ;CHECK IF ZERO
6673  043772  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6674  043776  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
6675  044004  104003                          ERROR   3
6676  044006                          5$:
6677  044006  016237  000006  001126          MOV     RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
6678  044014  001406                          BEQ     6$              ;CHECK IF ZERO
6679  044016  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6680  044022  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
6681  044030  104003                          ERROR   3
6682  044032                          6$:
6683  044032  016237  000016  001126          MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
6684  044040  001406                          BEQ     7$              ;CHECK IF ZERO
6685  044042  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6686  044046  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
6687  044054  104003                          ERROR   3
6688  044056                          7$:
6689  044056  016237  000010  001126          MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
6690  044064  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
6691  044072  001407                          BEQ     8$              ;YES,CONTINUE
6692  044074  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
6693  044102  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
6694  044110  104003                          ERROR   3
6695  044112                          8$:
6696  044112  016237  000012  001126          MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
6697  044120  001406                          BEQ     9$              ;CHECK IF ZERO
6698  044122  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6699  044126  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
6700  044134  104003                          ERROR   3
6701  044136  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
6702  044144  001406                          BEQ     10$             ;CHECK IF ZERO
6703  044146  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6704  044152  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
6705  044160  104003                          ERROR   3
6706  044162                          10$:
6707  044162  016237  000020  001126          MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
6708  044170  001406                          BEQ     12$             ;CHECK IF ZERO
6709  044172  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6710  044176  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
6711  044204  104003                          ERROR   3
6712  044206                          12$:
6713  044206  016237  000032  001126          MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
6714  044214  001406                          BEQ     15$             ;CHECK IF ZERO
```

```
6715  044216  005037  001124                      CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6716  044222  012737  067727  001322              MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
6717  044230  104003                              ERROR   3
6718  044232  016237  000030  001126  15$:        MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
6719  044240  012737  004066  001124  16$:        MOV     #4066,$GDDAT    ;USE 4066
6720  044246  023737  001124  001126  17$:        CMP     $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
6721  044254  001404                              BEQ     18$             ;YES,INITIALIZE RK611
6722  044256  012737  067702  001322              MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
6723  044264  104003                              ERROR   3
6724  044266  016237  000026  002014  18$:        MOV     RKMR1(R2),PREREG        ;GET PREVIOUS CONTENTS
6725  044274  012762  100000  000000              MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
6726  044302  016237  000026  001126              MOV     RKMR1(R2),$BDDAT        ;GET CURRENT VALUE
6727  044310  012737  002000  001124              MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
6728  044316  032737  020000  001126              BIT     #ECCW,$BDDAT
6729  044324  001403                              BEQ     35$
6730  044326  052737  020000  001124              BIS     #ECCW,$GDDAT
6731  044334                          35$:
6732  044334  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;CHECK IF RKMR1 CORRECT
6733  044342  001407                              BEQ     19$             ;YES, CHECK IF FINISHED
6734  044344  012737  063403  001330              MOV     #EM3,EM4N       ;LOAD ERROR MESSAGE
6735  044352  012737  067072  001332              MOV     #EM1009,EM4N+2
6736  044360  104004                              ERROR   4
6737  044362  104415                      19$:    SCOP1                   ;CHECK IF LOOP ON ERROR
6738  044364  000261                              SEC                     ;SHIFT IN ONE
6739  044366  006037  002010                      ROR     CONFIG
6740  044372  005301                              DEC     R1              ;CHECK IF FINISHED
6741  044374  001402                              BEQ     TST56           ;;YES, GO ON TO NEXT TEST
6742  044376  000137  043602                      JMP     1$
6743
6744                                  ;;*****************************************************************
6745                                  ;*TEST 56       REGISTER INTERACTION WITH PATTERN REG.
6746                                  ;*
6747                                  ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
6748                                  ;*      WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE PATTERN
6749                                  ;*      REGISTER TO 177777 AND MAKE SURE IT REMAINS 0 AND
6750                                  ;*      NO INTERACTION TAKES PLACE.
6751                                  ;*
6752                                  ;;************************************************************ *****
6753  044402  000004                  TST56:  SCOPE
6754  044404  012737  000764  001200              MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
6755  044412  013702  001270                      MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
6756  044416  012737  177777  002010              MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
6757  044424  012737  064220  001320              MOV     #EM14,EM3N      ;LOAD ERROR MESSAGE
6758  044432  012762  100000  000000              MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
6759  044440  005062  000002                      CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
6760  044444  012762  177777  000032              MOV     #177777,RKECPT(R2) ;WRITE RKECPT WITH 177777
6761  044452  016237  000000  001126              MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
6762  044460  022737  000200  001126              CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
6763  044466  001407                              BEQ     1$              ;YES,CONTINUE
6764  044470  012737  000200  001124              MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
6765  044476  012737  067314  001322              MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
6766  044504  104003                              ERROR   3
6767  044506  016237  000004  001126  1$:         MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
6768  044514  001406                              BEQ     2$              ;CHECK IF ZERO
6769  044516  005037  001124                      CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6770  044522  012737  067365  001322              MOV     #E11019,EM3N+2  ;LOAD ERROR MESSAGE
```

```
6771  044530  104003                        ERROR   3
6772  044532  016237  000002  001126  2$:   MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
6773  044540  001406                        BEQ     3$              ;CHECK IF ZERO
6774  044542  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6775  044546  012737  067335  001322        MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
6776  044554  104003                        ERROR   3
6777  044556  016237  000006  001126  3$:   MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG
6778  044564  001406                        BEQ     4$              ;CHECK IF ZERO
6779  044566  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6780  044572  012737  067412  001322        MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
6781  044600  104003                        ERROR   3
6782  044602  016237  000016  001126  4$:   MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
6783  044610  001406                        BEQ     5$              ;CHECK IF ZERO
6784  044612  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6785  044616  012737  067556  001322        MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
6786  044624  104003                        ERROR   3
6787  044626  016237  000010  001126  5$:   MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
6788  044634  022737  000100  001126        CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
6789  044642  001407                        BEQ     6$              ;YES,CONTINUE
6790  044644  012737  000100  001124        MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
6791  044652  012737  067450  001322        MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
6792  044660  104003                        ERROR   3
6793  044662  016237  000012  001126  6$:   MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
6794  044670  001406                        BEQ     7$              ;CHECK IF ZERO
6795  044672  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6796  044676  012737  067471  001322        MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
6797  044704  104003                        ERROR   3
6798  044706  016237  000014  001126  7$:   MOV     RKER(R2),$BDDAT ;STORE ERROR REG
6799  044714  001406                        BEQ     8$              ;CHECK IF ZERO
6800  044716  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6801  044722  012737  067527  001322        MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
6802  044730  104003                        ERROR   3
6803  044732  016237  000020  001126  8$:   MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
6804  044740  001406                        BEQ     10$             ;CHECK IF ZERO
6805  044742  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED RESULTS
6806  044746  012737  067624  001322        MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
6807  044754  104003                        ERROR   3
6808  044756  016237  000026  001126  10$:  MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
6809  044764  012737  002000  001124        MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
6810  044772  032737  020000  001126        BIT     #ECCW,$BDDAT
6811  045000  001403                        BEQ     11$
6812  045002  052737  020000  001124        BIS     #ECCW,$GDDAT
6813  045010  023737  001124  001126  11$:  CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
6814  045016  001404                        BEQ     12$             ;YES,CONTINUE TEST
6815  045020  012737  067651  001322        MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
6816  045026  104003                        ERROR   3
6817  045030  016237  000032  001126  12$:  MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
6818  045036  001406                        BEQ     13$             ;CHECK IF ZERO
6819  045040  005037  001124                CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6820  045044  012737  067727  001322        MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
6821  045052  104003                        ERROR   3
6822  045054  016237  000030  001126  13$:  MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
6823  045062  022737  004066  001126        CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
6824  045070  001407                        BEQ     TST57           ;;YES,GO TO NEXT TEST
6825  045072  012737  004066  001124        MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
6826  045100  012737  067702  001322        MOV     #EM1029,EM3N+2  ;LOAD ERROR MESAGE
```

```
 6827  045106  104003                          ERROR   3
 6828
 6829                                  ;;*****************************************************************
 6830                                  ;*TEST 57       REGISTER INTERACTION WITH POSITION REG.
 6831                                  ;*
 6832                                  ;*       ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
 6833                                  ;*       WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE POSITION
 6834                                  ;*       REGISTER TO 177777 AND MAKE SURE IT STAYS AT THE
 6835                                  ;*       INITIALIZED CONDITION AND NO INTERACTION TAKES PLACE.
 6836                                  ;*
 6837                                  ;;*****************************************************************
 6838  045110  000004                  T57T57: SCOPE
 6839  045112  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
 6840  045120  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE ADDRESS
 6841  045124  012737  177777  002010          MOV     #177777,CONFIG  ;LOAD CONFIGURATION WORD
 6842  045132  012737  064260  001320          MOV     #EM15,EM3N      ;LOAD ERROR MESSAGE
 6843  045140  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
 6844  045146  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
 6845  045152  012762  177777  000030          MOV     #177777,RKECPS(R2) ;WRITE RKECPS WITH 177777
 6846  045160  016237  000000  001126          MOV     RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
 6847  045166  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
 6848  045174  001407                          BEQ     1$              ;YES,CONTINUE
 6849  045176  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
 6850  045204  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
 6851  045212  104003                          ERROR   3
 6852  045214  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADD REG.
 6853  045222  001406                          BEQ     2$              ;CHECK IF ZERO
 6854  045224  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6855  045230  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
 6856  045236  104003                          ERROR   3
 6857  045240  016237  000002  001126  2$:     MOV     RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
 6858  045246  001406                          BEQ     3$              ;CHECK IF ZERO
 6859  045250  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6860  045254  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
 6861  045262  104003                          ERROR   3
 6862  045264  016237  000006  001126  3$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADD REG
 6863  045272  001406                          BEQ     4$              ;CHECK IF ZERO
 6864  045274  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6865  045300  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
 6866  045306  104003                          ERROR   3
 6867  045310  016237  000016  001126  4$:     MOV     RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
 6868  045316  001406                          BEQ     5$              ;CHECK IF ZERO
 6869  045320  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6870  045324  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
 6871  045332  104003                          ERROR   3
 6872  045334  016237  000010  001126  5$:     MOV     RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
 6873  045342  022737  000100  001126          CMP     #IR,$BDDAT      ;CHECK IF CS2 CORRECT
 6874  045350  001407                          BEQ     6$              ;YES,CONTINUE
 6875  045352  012737  000100  001124          MOV     #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
 6876  045360  012737  067450  001322          MOV     #EM1021,EM3N+2  ;LOAD ERROR MESSAGE
 6877  045366  104003                          ERROR   3
 6878  045370  016237  000012  001126  6$:     MOV     RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
 6879  045376  001406                          BEQ     7$              ;CHECK IF ZERO
 6880  045400  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 6881  045404  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 6882  045412  104003                          ERROR   3
```

```
6883  045414  016237  000014  001126  7$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REG
6884  045422  001406                          BEQ     8$              ;CHECK IF ZERO
6885  045424  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6886  045430  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
6887  045436  104003                          ERROR   3
6888  045440  016237  000020  001126  8$:     MOV     RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
6889  045446  001406                          BEQ     10$             ;CHECK IF ZERO
6890  045450  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED RESULTS
6891  045454  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
6892  045462  104003                          ERROR   3
6893  045464  016237  000026  001126  10$:    MOV     RKMR1(R2),$BDDAT ;STORE MAINTENANCE REG 1
6894  045472  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED CONTENTS
6895  045500  032737  020000  001126          BIT     #ECCW,$BDDAT
6896  045506  001403                          BEQ     11$
6897  045510  052737  020000  001124          BIS     #ECCW,$GDDAT
6898  045516  023737  001124  001126  11$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
6899  045524  001404                          BEQ     12$             ;YES,CONTINUE TEST
6900  045526  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
6901  045534  104003                          ERROR   3
6902  045536  016237  000032  001126  12$:    MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
6903  045544  001406                          BEQ     13$             ;CHECK IF ZERO
6904  045546  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
6905  045552  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
6906  045560  104003                          ERROR   3
6907  045562  016237  000030  001126  13$:    MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
6908  045570  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION CORRECT
6909  045576  001407                          BEQ     TST60           ;;YES,GO TO NEXT TEST
6910  045600  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
6911  045606  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESAGE
6912  045614  104003                          ERROR   3
6913
6914
6915                                  .SBTTL  **INTERRUPT TESTS
6916
6917                                  ;;*****************************************************************
6918                                  ;*TEST 60        RK611 INTERRUPT
6919                                  ;*
6920                                  ;*      STORE LOCATIONS 0-776, LOAD LOCATIONS 0-776 TO TRAP ALL
6921                                  ;*      POSSIBLE INTERRUPTS.  LOWER PROCESSOR PRIORITY TO ZERO.
6922                                  ;*      MAKE SURE THAT NO INTERRUPT OCCURS.  NOW SET INTERRUPT
6923                                  ;*      ENABLE AND READY.  VERIFY THAT THE INTERRUPT OCCURS AT
6924                                  ;*      PROPER VECTOR ADDRESS.  MAKE SURE THAT INTERRUPT IS
6925                                  ;*      CLEARED AFTER IT IS GIVEN.
6926                                  ;*
6927                                  ;*****************************************************************
6928  045616  000004                  TST60:  SCOPE
6929  045620  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
6930  045626  012701  067754                  MOV     #SAVVEC,R1      ;LOAD SAVED ADDRESS
6931  045632  005000                          CLR     R0              ;LOAD START OF VECTOR SPACE
6932  045634  012703  000400                  MOV     #400,R3         ;LOAD COUNT
6933  045640  012021                  1$:     MOV     (R0)+,(R1)+     ;SAVE TRAP CATCHER
6934  045642  005303                          DEC     R3
6935  045644  001375                          BNE     1$
6936  045646  012737  177777  002016          MOV     #-1,SAVFLG      ;INDICATE TRAP CATCHER SAVED
6937  045654  005000                          CLR     R0              ;LOAD ADDRESS OF START OF VECTOR SPACE
6938  045656  012701  046010                  MOV     #10$,R1         ;LOAD START OF VECTOR TABLE
```

```
 6939  045662  012703  000200            2$:   MOV     #200,R3           ;LOAD COUNT
 6940  045666  010120                          MOV     R1,(R0)+          ;LOAD VECTOR SPACE
 6941  045670  012720  000340                  MOV     #PR7,(R0)+        ;  PRIORITY 7
 6942  045674  062701  000002                  ADD     #2,R1
 6943  045700  005303                          DEC     R3
 6944  045702  001371                          BNE     2$
 6945  045704  012762  100000  OLJOOO          MOV     #CCLR,RKCS1(R2)   ;CLEAR RK611 WITH CONTROLLER CLEAR
 6946  045712  012705  046410                  MOV     #11$,R5           ;LOAD ADDRESS FOR UNEXPECTED INTERRUPT
 6947  045716  005046                          CLR     -(SP)             ;LOAD STACK TO ALLOW ALL INTERRUPTS
 6948  045720  012746  045726                  MOV     #64$,-(SP)        ;LOAD NEXT ADDRESS
 6949  045724  000002                          RTI                       ;CLEAR PSW
 6950
 6951  045726                            64$:
 6952  045726  000240                          NOP                       ;ALLOW INTERRUPT TO OCCUR
 6953  045730  012705  046462                  MOV     #15$,R5           ;LOAD ADDRESS FOR EXPECTED INTERRUPT
 6954  045734  012762  000300  000000          MOV     #RDY!IE,RKCS1(R2) ;GENERATE INTERRUPT
 6955  045742  000240                          NOP                       ;ALLOW INTERRUPT TO OCCUR
 6956  045744  012746  000340                  MOV     #PR7,-(SP)        ;LOCK OUT ALL INTERRUPTS
 6957  045750  012746  045756                  MOV     #3$,-(SP)
 6958  045754  000002                          RTI
 6959
 6960  045756  012701  067754            3$:   MOV     #SAVVEC,R1        ;LOAD SAVED TRAP CATCHER ADDRESS
 6961  045762  005000                          CLR     R0                ;LOAD START OF VECTOP SPACE
 6962  045764  012703  000400                  MOV     #400,R3           ;LOAD COUNT
 6963  045770  012120                    4$:   MOV     (R1)+,(R0)+       ;RESTORE TRAP CATCHER
 6964  045772  005303                          DEC     R3
 6965  045774  001375                          BNE     4$
 6966  045776  005037  002016                  CLR     SAVFLG            ;INDICATE TRAP CATCHER RESTORED
 6967  046002  104005                          ERROR   5                 ;REPORT INTERRUPT DID NOT OCCUR
 6968  046004  000137  046672                  JMP     60$               ;GO ON TO NEXT TEST
 6969
 6970  046010                            10$:
 6971  046010  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6972  046012  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6973  046014  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6974  046016  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6975  046020  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6976  046022  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6977  046024  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6978  046026  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6979  046030  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6980  046032  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6981  046034  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6982  046036  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6983  046040  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6984  046042  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6985  046044  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6986  046046  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6987  046050  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6988  046052  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6989  046054  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6990  046056  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6991  046060  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6992  046062  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6993  046064  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
 6994  046066  004715                          JSR     PC,(R5)           ;LOAD STACK FOR VECTOR ADDRESS
```

```
 6995  046070  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 6996  046072  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 6997  046074  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 6998  046076  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 6999  046100  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7000  046102  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7001  046104  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7002  046106  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7003  046110  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7004  046112  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7005  046114  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7006  046116  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7007  046120  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7008  046122  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7009  046124  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7010  046126  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7011  046130  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7012  046132  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7013  046134  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7014  046136  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7015  046140  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7016  046142  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7017  046144  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7018  046146  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7019  046150  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7020  046152  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7021  046154  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7022  046156  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7023  046160  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7024  046162  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7025  046164  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7026  046166  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7027  046170  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7028  046172  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7029  046174  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7030  046176  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7031  046200  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7032  046202  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7033  046204  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7034  046206  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7035  046210  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7036  046212  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7037  046214  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7038  046216  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7039  046220  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7040  046222  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7041  046224  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7042  046226  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7043  046230  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7044  046232  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7045  046234  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7046  046236  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7047  046240  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7048  046242  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7049  046244  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7050  046246  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
```

```
 7051  046250  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7052  046252  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7053  046254  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7054  046256  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7055  046260  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7056  046262  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7057  046264  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7058  046266  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7059  046270  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7060  046272  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7061  046274  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7062  046276  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7063  046300  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7064  046302  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7065  046304  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7066  046306  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7067  046310  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7068  046312  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7069  046314  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7070  046316  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7071  046[illegible]  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7072  046[illegible]  004715                JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7073  046324  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7074  046326  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7075  046330  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7076  046332  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7077  046334  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7078  046336  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7079  046340  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7080  046342  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7081  046344  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7082  046346  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7083  046350  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7084  046352  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7085  046354  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7086  046356  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7087  046360  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7088  046362  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7089  046364  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7090  046366  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7091  046370  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7092  046372  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7093  046374  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7094  046376  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7095  046400  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7096  046402  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7097  046404  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7098  046406  004715                        JSR     PC,(R5)         ;LOAD STACK FOR VECTOR ADDRESS
 7099
 7100  046410  012637  001122        11$:    MOV     (SP)+,$BDADR    ;DETERMINE VECTOR ADDRESS
 7101  046414  162737  046012  001122        SUB     #10$+2,$BDADR
 7102  046422  006337  001122                ASL     $BDADR
 7103  046426  062706  000004                ADD     #4,SP           ;ADJUST STACK
 7104  046432  012701  067754                MOV     #SAVTEC,R1      ;LOAD SAVED TRAP CATCHER ADDRESS
 7105  046436  005000                        CLR     R0              ;LOAD START OF VECTOR SPACE
 7106  046440  012703  000400                MOV     #400,R3         ;LOAD COUNT
```

```
7107  046444  012120                  12$:    MOV     (R1)+,(R0)+     ;RESTORE TRAP CATCHER
7108  046446  005303                          DEC     R3
7109  046450  001375                          BNE     12$
7110  046452  005037  002016                  CLR     SAVFLG          ;INDICATE TRAP CATCHER RESTORED
7111  046456  104006                          ERROR   6               ;REPORT UNEXPECTED INTERRUPT
7112  046460  000504                          BR      TST61           ;;GO ON TO NEXT TEST
7113
7114  046462  012637  001122          15$:    MOV     (SP)+,$BDADR    ;DETERMINE VECTOR ADDRESS
7115  046466  162737  046012  001122          SUB     #10$+2,$BDADR
7116  046474  006337  001122                  ASL     $BDADR
7117  046500  062706  000004                  ADD     #4,SP           ;ADJUST STACK
7118  046504  023737  002000  001122          CMP     RKVEC,$BDADR    ;CHECK IF VECTOR ADDRESS CORRECT
7119  046512  001414                          BEQ     20$             ;YES, CONTINUE
7120  046514  012701  067754                  MOV     #SAVVEC,R1      ;LOAD SAVED TRAP CATCHER ADDRESS
7121  046520  005000                          CLR     R0              ;START OF VECTOR SPACE
7122  046522  012703  000400                  MOV     #400,R3         ;LOAD COUNT
7123  046526  012120                  16$:    MOV     (R1)+,(R0)+     ;RESTORE TRAP CATCHER
7124  046530  005303                          DEC     R3
7125  046532  001375                          BNE     16$
7126  046534  005037  002016                  CLR     SAVFLG          ;INDICATE TRAP CATCHER RESTORED
7127  046540  104007                          ERROR   7               ;REPORT INCORRECT VECTOR ADDRESS
7128  046542  000453                          BR      TST61           ;;GO ON TO NEXT TEST
7129
7130  046544  012705  046622          20$:    MOV     #25$,R5         ;LOAD ADDRESS FOR UNEXPECTED INTERRUPT
7131  046550  005046                          CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
7132  046552  012746  046560                  MOV     #65$,-(SP)      ;LOAD NEXT ADDRESS
7133  046556  000002                          RTI                     ;CLEAR PSW
7134
7135  046560                          65$:
7136  046560  012746  000340                  MOV     #PR7,-(SP)      ;LOCK OUT RK611 INTERRUPTS
7137
7138  046564  012746  046572                  MOV     #21$,-(SP)
7139  046570  000002                          RTI
7140
7141  046572  000240                  21$:    NOP                     ;ALLOW INTERRUPT TO OCCUR
7142  046574  012701  067754                  MOV     #SAVVEC,R1      ;LOAD SAVE TRAP CATCHER ADDRESS
7143  046600  005000                          CLR     R0              ;START OF VECTOR SPACE
7144  046602  012703  000400                  MOV     #400,R3 ;LOAD COUNT
7145  046606  012120                  22$:    MOV     (R1)+,(R0)+     ;RESTORE TRAP CATCHER
7146  046610  005303                          DEC     R3
7147  046612  001375                          BNE     22$
7148  046614  005037  002016                  CLR     SAVFLG          ;INDICATE TRAP CATCHER RESTORED
7149  046620  000424                          BR      TST61           ;;GO ON TO NEXT TEST
7150
7151  046622  012637  001122          25$:    MOV     (SP)+,$BDADR    ;DETERMINE VECTOR ADDRESS
7152  046626  162737  046012  001122          SUB     #10$+2,$BDADR
7153  046634  006337  001122                  ASL     $BDADR
7154  046640  062706  000004                  ADD     #4,SP           ;ADJUST STACK
7155  046644  012701  067754                  MOV     #SAVVEC,R1      ;LOAD SAVE TRAP CATCHER ADDRESS
7156  046650  005000                          CLR     R0              ;START OF VECTOR SPACE
7157  046652  012703  000400                  MOV     #400,R3         ;LOAD COUNT
7158  046656  012120                  26$:    MOV     (R1)+,(R0)+     ;RESTORE TRAP CATCHER
7159  046660  005303                          DEC     R3
7160  046662  001375                          BNE     26$
7161  046664  005037  002016                  CLR     SAVFLG          ;INDICATE THAT TRAP CATCHER RESTART
7162
```

```
7163  046670  104010                        ERROR   10              ;REPORT ATTENTION DID NOT CLEAR
7164  046672                          60$:
7165
7166                                  ;*****************************************************************
7167                                  ;*TEST 61        INTERRUPT PRIORITY
7168                                  ;*
7169                                  ;*      SET UP PRIORITY TO 1 LESS THAN INTERRUPT PRIORITY.
7170                                  ;*      WRITE READY WITH INTERRUPT ENABLE.  MAKE SURE INTERRUPT OCCURS.
7171                                  ;*
7172                                  ;*      NOW SET UP PRIORITY EQUAL TO INTERRUPT PRIORITY.
7173                                  ;*      WRITE INTERRUPT ENABLE WITH READY.  MAKE SURE INTERRUPT
7174                                  ;*      DOES NOT  OCCUR.  NOW LOWER PRIOIRY AND MAKE
7175                                  ;*      INTERRUPT HAS BEEN STORED.
7176                                  ;*
7177                                  ;*****************************************************************
7178  046672  000004                  TST61:  SCOPE
7179  046674  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
7180  046702  013701  002000                  MOV     RKVEC,R1        ;LOAD RK611 VECTOR ADDRESS FOR INTERRUPT
7181  046706  012721  046776                  MOV     #10$,(R1)+      ;  PRIORITY 7
7182  046712  013746  002002                  MOV     RKPRI,-(SP)     ;SET PROCESSOR PRIORITY =
7183  046716  162716  000040                  SUB     #40,(SP)        ;  RK611 PRIORITY -1
7184  046722  011646                          MOV     (SP),-(SP)
7185  046724  006216                          ASR     (SP)
7186  046726  006216                          ASR     (SP)
7187  046730  006216                          ASR     (SP)
7188  046732  006216                          ASR     (SP)
7189  046734  006216                          ASR     (SP)
7190  046736  012637  002020                  MOV     (SP)+,PRIOR
7191  046742  012746  046750                  MOV     #1$,-(SP)
7192  046746  000002                          RTI
7193
7194  046750  012762  000300  000000  1$:     MOV     #RDY!IE,RKCS1(R2)       ;GENERATE RK611 INTERRUPT
7195  046756  000240                          NOP                     ;WAIT FOR INTERRUPT
7196  046760  012746  000340                  MOV     #PR7,-(SP)      ;LOCK OUT INTERRUPTS
7197  046764  012746  046772                  MOV     #2$,-(SP)
7198  046770  000002                          RTI
7199
7200  046772  104011                  2$:     ERROR   11              ;EXPECTED INTERRUPT DID NOT OCCUR AT
7201                                                                  ;  PROCESSOR PRIORITY
7202  046774  000462                          BR      60$             ;RESTORE TRAP CATCHER
7203
7204  046776  062706  000004          10$:    ADD     #4,SP           ;ADJUST STACK
7205  047002  012777  047126  132770          MOV     #20$,@RKVEC     ;LOAD RK611 VECTOR FOR UNEXPECTED
7206                                                                  ;  INTERRUPT
7207  047010  013737  002002  002020          MOV     RKPRI,PRIOR     ;STORE PRIORITY PRINT OUT
7208  047016  006237  002020                  ASR     PRIOR
7209  047022  006237  002020                  ASR     PRIOR
7210  047026  006237  002020                  ASR     PRIOR
7211  047032  006237  002020                  ASR     PRIOR
7212  047036  006237  002020                  ASR     PRIOR
7213  047042  013746  002002                  MOV     RKPRI,-(SP)     ;SET PROCESSOR PRIORITY
7214  047046  012746  047054                  MOV     #11$,-(SP)      ;  RK611 PRIORITY
7215  047052  000002                          RTI
7216
7217  047054  012762  000300  000000  11$:    MOV     #RDY!IE,RKCS1(R2)       ;GENERATE RK611 INTERRUPT
7218  047062  000240                          NOP                     ;ALLOW INTERRUPT TO OCCUR
```

```
7219  047064  005037  002020              CLR     PRIOR           ;LOAD PRIORITY FOR PRINT OUT
7220  047070  012777  047136  132702      MOV     #25$,@RKVEC     ;LOAD RK611 VECTOR FOR INTERRUPT
7221  047076  005046                      CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
7222  047100  012746  047106              MOV     #64$,-(SP)      ;LOAD NEXT ADDRESS
7223  047104  000002                      RTI                     ;CLEAR PSW
7224
7225  047106                      64$:
7226  047106  000240                      NOP                     ;ALLOW INTERRUPT TO OCCUR
7227  047110  012746  000340              MOV     #PR7,-(SP)      ;LOCK OUT INTERRUPTS
7228  047114  012746  047122              MOV     #12$,-(SP)
7229  047120  000002                      RTI
7230
7231  047122  104013              12$:    ERROR   13              ;INTERRUPT DID NOT OCCUR WHEN
7232                                                              ;  PRIORITY LOWERED
7233  047124  000406                      BR      60$             ;RESTORE TRAP CATCHER
7234
7235  047126  062706  000004      20$:    ADD     #4,SP           ;ADJUST STACK
7236  047132  104012                      ERROR   12              ;UNEXPECTED INTERRUPT OCCURRED
7237                                                              ;  AT PROCESSOR PRIORITY
7238  047134  000402                      BR      60$             ;RESTORE TRAP CATCHER
7239
7240  047136  062706  000004      25$:    ADD     #4,SP           ;ADJUST STACK
7241
7242  047142  013701  002000      60$:    MOV     RKVEC,R1        ;RESTORE TRAP CATCHER
7243  047146  010111                      MOV     R1,(R1)
7244  047150  062721  000002              ADD     #2,(R1)+
7245  047154  005011                      CLR     (R1)
7246
7247                              ;;*****************************************************************
7248                              ;*TEST 62       SETTING INTERRUPT ENABLE
7249                              ;*
7250                              ;*      CLEAR RK611 CONTROLLER WITH CONTROLLER CLEAR. ALLOW RK611
7251                              ;*      INTERRUPTS BY SETTING PROCESSOR PRIORITY TO ZERO.
7252                              ;*      SET INTERRUPT ENABLE AND MAKE SURE NO INTERRUPTS OCCUR
7253                              ;*
7254                              ;*****************************************************************
7255  047156  000004              ;ST62:  SCOPE
7256  047160  012737  000764  001200      MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
7257  047166  012762  100000  000000      MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7258  047174  013701  002000              MOV     RKVEC,R1        ;LOAD RK611 VECTOR ADDRESS FOR
7259  047200  012721  047242              MOV     #10$,(R1)+      ;  UNEXPECTED INTERRUPT
7260  047204  012711  000340              MOV     #PR7,(R1)       ;  PRIORITY 7
7261  047210  005046                      CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
7262  047212  012746  047220              MOV     #64$,-(SP)      ;LOAD NEXT ADDRESS
7263  047216  000002                      RTI                     ;CLEAR PSW
7264
7265  047220                      64$:
7266  047220  012762  000100  000000      MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
7267  047226  000240                      NOP                     ;ALLOW INTERRUPT TO OCCUR
7268  047230  012746  000340              MOV     #PR7,-(SP)      ;LOCK OUT ALL INTERRUPTS
7269  047234  012746  047250              MOV     #15$,-(SP)      ;RESTORE TRAP CATCHER
7270  047240  000002                      RTI
7271
7272  047242  062706  000004      10$:    ADD     #4,SP           ;ADJUST STACK
7273  047246  104014                      ERROR   14              ;REPORT ERROR
7274  047250  013701  002000      15$:    MOV     RKVEC,R1        ;RESTORE TRAP CATCHER
```

```
7275  047254  010111                          MOV     R1,(R1)
7276  047256  012721  000002                  MOV     #2,(R1)+
7277  047262  005011                          CLR     (R1)
7278
7279                                  ;;*****************************************************************
7280                                  ;*TEST 63        INTERRUPT CLEARING
7281                                  ;*
7282                                  ;*      SET UP PRIORITY TO SEVEN.  CREATE INTERRUPT BY SETTING
7283                                  ;*      INTERRUPT ENABLE  READY.  AND CLEAR IT WITH CONTROLLER
7284                                  ;*      CLEAR. SET INTERRUPT ENABLE.  NOW LOWER PRIORITY
7285                                  ;*      TO MAKE SURE NO INTERRUPT OCCURS.
7286                                  ;*
7287                                  ;*****************************************************************
7288  047264  000004                  $ST63:  SCOPE
7289  047266  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
7290  047274  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
7291  047302  012762  000300  000000          MOV     #IE!RDY,RKCS1(R2)       ;GENERATE INTERRUPT
7292  047310  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR INTERRUPT
7293  047316  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;GET INTERRUPT ENABLE
7294  047324  013701  002000                  MOV     RKVEC,R1        ;LOAD RK611 VECTOR ADDRESS FOR
7295  047330  012721  047364                  MOV     #10$,(R1)+      ;  UNEXPECTED INTERRUPT
7296  047334  012711  000340                  MOV     #PR7,(R1)       ;  PRIORITY 7
7297  047340  005046                          CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
7298  047342  012746  047350                  MOV     #64$,-(SP)      ;LOAD NEXT ADDRESS
7299  047346  000002                          RTI                     ;CLEAR PSW
7300
7301  047350                          64$:
7302  047350  000240                          NOP                     ;ALLOW INTERRUPT TO OCCUR
7303  047352  012746  000340                  MOV     #PR7,-(SP)      ;LOCK OUT INTERRUPTS
7304  047356  012746  047372                  MOV     #15$,-(SP)      ;RESTORE TRAP CATCHER
7305  047362  000002                          RTI
7306
7307  047364  062706  000004          10$:    ADD     #4,SP           ;ADJUST STACK
7308  047370  104015                          ERROR   15              ;REPORT ERROR
7309  047372  013701  002000          15$:    MOV     RKVEC,R1        ;RESTORE TRAP CATCHER
7310  047376  010111                          MOV     R1,(R1)
7311  047400  062721  000002                  ADD     #2,(R1)+
7312  047404  005011                          CLR     (R1)
7313
7314
7315                                  .SBTTL  **SILO  TESTS
7316
7317                                  ;;*****************************************************************
7318                                  ;*TEST 64        READ SILO WHEN EMPTY
7319                                  ;*
7320                                  ;*      READ SILO WHEN EMPTY.  CHECK FOR DATA LATE AND CONTROLLER
7321                                  ;*      ERROR.  ISSUE CONTROLLER CLEAR AND CHECK IF ERROR RESET.
7322                                  ;*
7323                                  ;*****************************************************************
7324  047406  000004                  $ST64:  SCOPE
7325  047410  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
7326  047416  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
7327  047422  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7328  047430  005762  000024                  TST     RKDB(R2)        ;READ DATA BUFFER
7329  047434  016237  000010  001710          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
7330  047442  016237  000000  001700          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
```

```
7331  047450  012737  100100  001750          MOV     #DLT!IR,E.CS2   ;LOAD EXPECTED CS2
7332  047456  012737  100200  001740          MOV     #CERR!RDY,E.CS  ;LOAD EXPECTED CS1
7333  047464  023737  001750  001710          CMP     E.CS2,T.CS2     ;CHECK FOR DAT LATE SET
7334  047472  001401                          BEQ     1$              ;YES, CHECK FOR CONTROLLER ERROR
7335  047474  104016                          ERROR   16              ;CS2 INCORRECT AFTER READING
7336                                                                  ;  EMPTY SILO
7337  047476  023737  001740  001700  1$:     CMP     E.CS1,T.CS1     ;CHECK FOR CONTROLLER ERROR
7338  047504  001401                          BEQ     2$              ;YES, CLEAR DATA LATE
7339  047506  104017                          ERROR   17              ;CS1 INCORRECT AFTER REQUIRE
7340                                                                  ;  EMPTY SILO
7341  047510  012762  100000  000000  2$:     MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7342  047516  016237  000000  001700          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7343  047524  016237  000010  001710          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
7344  047532  012737  000200  001740          MOV     #RDY,E.CS1      ;LOAD EXPECTED CS1
7345  047540  012737  000100  001750          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
7346  047546  023737  001740  001700          CMP     E.CS1,T.CS1     ;CHECK TO CONTROLLER ERROR RESET
7347  047554  001401                          BEQ     3$              ;YES, CHECK IF DATA LATE RESET
7348  047556  104020                          ERROR   20              ;CS1 INCORRECT AFTER ATTEMPTING
7349                                                                  ;  TO CLEAR DATA LATE
7350  047560  023737  001750  001710  3$:     CMP     E.CS2,T.CS2     ;CHECK IF DATA LATE RESET
7351  047566  001401                          BEQ     TST65           ;YES, GO ON TO NEXT TEST
7352  047570  104021                          ERROR   21              ;CS2 INCORRECT AFTER ATTEMPTING
7353                                                                  ;  TO OCCUR DATA LATE
7354
7355                        ;*****************************************************************
7356                        ;*TEST 65       SILO LOADING AND UNLOADING OF ONE WORD
7357                        ;*
7358                        ;*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 CONTROLLER.
7359                        ;*      CLEAR WORD COUNT REGISTER.
7360                        ;*
7361                        ;*      WRITE A WORD OF 177777 INTO THE SILO.  CHECK ALL OTHER
7362                        ;*      REGISTERS FOR INTERACTION PROBLEMS.  CHECK THAT OUTPUT
7363                        ;*      READY IS SET IN COMMAND AND STATUS REGISTER 2.  IF NOT
7364                        ;*      WAIT A REASONABLE TIME FOR IT.
7365                        ;*
7366                        ;*      IF OUTPUT READY COMES UP IN A REASONABLE TIME, READ BACK
7367                        ;*      CONTENTS AND MAKE SURE IT IS 177777.  CHECK FOR NO CONTROLLER
7368                        ;*      ERROR, NO DATA LATE, INPUT READY SET, OUTPUT READY RESET.
7369                        ;*      NOW READ ANOTHER WORD FROM THE SILO TO MAKE SURE DATA
7370                        ;*      LATE AND CONTROLLER ERROR SET.
7371                        ;*
7372                        ;*****************************************************************
7373  047572  000004        TST65:  SCOPE
7374  047574  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
7375  047602  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7376  047610  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG
7377  047614  012737  177777  002010          MOV     #177777,CONFIG  ;LOAD CONFIGURATION FOR PRINT OUT
7378  047622  012762  177777  000024          MOV     #177777,RKDB(R2)        ;WRITE DATA BUFFER WITH ALL 1'S
7379  047630  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE COMMAND AND STATUS REG 1
7380  047636  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
7381  047644  001407                          BEQ     1$              ;YES,CHECK BUS AND REG
7382  047646  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
7383  047654  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSSAGE
7384  047662  104003                          ERROR   3
7385  047664  016237  000004  001126  1$:     MOV     RKBA(R2),$BDDAT ;STORE BUS AND
7386  047672  001406                          BEQ     2$              ;CHECK IF ZERO
```

```
7387  047674  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7388  047700  012737  067365  001322              MOV     #EM1019,EM3N+2      ;LOAD ERROR MESSAGE
7389  047706  104003                              ERROR   3
7390  047710  016237  000002  001126  2$:         MOV     RKWC(R2),$BDDAT     ;STORE WORD COUNT REG
7391  047716  001406                              BEQ     3$                  ;CHECK IF ZERO
7392  047720  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7393  047724  012737  067335  001322              MOV     #EM1018,EM3N+2      ;LOAD ERROR MESSAGE
7394  047732  104003                              ERROR   3
7395  047734  016237  000006  001126  3$:         MOV     RKDA(R2),$BDDAT     ;STORE DESK ADDRESS REG
7396  047742  001406                              BEQ     4$                  ;CHECK IF ZERO
7397  047744  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7398  047750  012737  067412  001322              MOV     #EM1020,EM3N+2      ;LOAD ERROR MESSAGE
7399  047756  104003                              ERROR   3
7400  047760  016237  000016  001126  4$:         MOV     RKASOF(R2),$BDDAT       ;STORE ATTENTION SUMMARY AND OFFSET
7401  047766  001406                              BEQ     5$                  ;CHECK IF ZERO
7402  047770  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7403  047774  012737  067702  001322              MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
7404  050002  104003                              ERROR   3
7405  050004  012700  000005          5$:         MOV     #5,R0               ;LOAD COUNTER TO WAIT FOR OUTPUT READY
7406  050010  005300                  6$:         DEC     R0
7407  050012  001376                              BNE     6$
7408  050014  016237  000010  001126              MOV     RKCS2(R2),$BDDAT        ;STORE COMMAND AND STATUS REG. 2
7409  050022  022737  000300  001126              CMP     #IR!OR,$BDDAT       ;CHECK IF CS2 CORRECT
7410  050030  001407                              BEQ     7$                  ;YES, CONTINUE TEST
7411  050032  012737  000300  001124              MOV     #IR!OR,$GDDAT       ;LOAD EXPECTED CONTENTS
7412  050040  012737  066704  001322              MOV     #EM1004,EM3N+2      ;LOAD ERROR MESSAGE
7413  050046  104003                              ERROR   3
7414  050050  016237  000012  001126  7$:         MOV     RKDS(R2),$BDDAT     ;STORE DRIVE STATUS REG.
7415  050056  001406                              BEQ     8$                  ;CHECK IF ZERO
7416  050060  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7417  050064  012737  067471  001322              MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
7418  050072  104003                              ERROR   3
7419  050074  016237  000014  001126  8$:         MOV     RKER(R2),$BDDAT     ;STORE ERROR REGISTER
7420  050102  001406                              BEQ     9$                  ;CHECK IF ZERO
7421  050104  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7422  050110  012737  067527  001322              MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
7423  050116  104003                              ERROR   3
7424  050120  016237  000020  001126  9$:         MOV     RKDCYL(R2),$BDDAT   ;STORE CYLINDER ADDRESS REG.
7425  050126  001406                              BEQ     11$                 ;CHECK IF ZERO
7426  050130  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7427  050134  012737  067624  001322              MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
7428  050142  104003                              ERROR   3
7429  050144  016237  000026  001126  11$:        MOV     RKMR1(R2),$BDDAT        ;STORE MAINTENANCE REG. 1
7430  050152  012737  002000  001124              MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
7431  050160  032737  020000  001126              BIT     #ECCW,$BDDAT
7432  050166  001403                              BEQ     12$
7433  050170  052737  020000  001124              BIS     #ECCW,$GDDAT
7434  050176  023737  001124  001126  12$:        CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
7435  050204  001404                              BEQ     13$                 ;YES, CONTINUE
7436  050206  012737  067651  001322              MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
7437  050214  104003                              ERROR   3
7438  050216  016237  000032  001126  13$:        MOV     RKECPT(R2),$BDDAT       ;STORE ECC PATTERN REG.
7439  050224  001406                              BEQ     14$                 ;CHECK IF ZERO
7440  050226  005037  001124                      CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7441  050232  012737  067044  001322              MOV     #EM1008,EM3N+2      ;LOAD ERROR MESSAGE
7442  050240  104003                              ERROR   3
```

```
7443  050242  016237  000030  001126  14$:    MOV     RKECPS(R2),$BDDAT       ;STORE REC POSITION REC
7444  050250  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF POSITION CORRECT
7445  050256  001407                          BEQ     15$             ;YES, CONTINUE
7446  050260  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
7447  050266  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
7448  050274  104003                          ERROR   3
7449  050276  016237  000024  001126  15$:    MOV     RKDB(R2),$BDDAT ;STORE DATA BUFFER
7450  050304  022737  177777  001126          CMP     #177777,$BDDAT  ;CHECK CONTENTS CORRECT
7451  050312  001407                          BEQ     16$             ;YES, CHECK TO MAKE SURE
7452                                                                  ;  DATA LATE RESET AND
7453                                                                  ;  CONTROLLER ERROR RESET
7454  050314  012737  177777  001124          MOV     #177777,$GDDAT  ;LOAD EXPECTED CONTENTS
7455  050322  012737  067044  001322          MOV     #EM1008,EM3N+2  ;LOAD ERROR MESSAGE
7456  050330  104003                          ERROR   3
7457  050332  016237  000000  001700  16$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7458  050340  016237  000010  001710          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
7459  050346  012737  000200  001740          MOV     #RDY,E.CS1      ;LOAD EXPECTED CS1
7460  050354  012737  000100  001750          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
7461  050362  023737  001740  001700          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
7462  050370  001401                          BEQ     17$             ;YES CHECK OUTPUT READY RESET
7463  050372  104022                          ERROR   22              ;ATTEMPTING TO READ SILO CONTAINING
7464                                                                  ;  ONE WORD - CS1 INCORRECT
7465  050374  023737  001750  001710  17$:    CMP     E.CS2,T.CS2     ;CHECK IN OUTPUT READY RESET
7466  050402  001401                          BEQ     18$             ;YES, READ AN EXTRA WORD
7467  050404  104023                          ERROR   23              ;ATTEMPTING TO READ SILO COMMAND
7468                                                                  ;  ONE WORD - CS2 INCORRECT
7469  050406  005762  000024          18$:    TST     RKDB(R2)        ;READ DATA BUFFER
7470  050412  016237  000000  001700          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 7
7471  050420  016237  000010  001710          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
7472  050426  012737  100200  001740          MOV     #CERR!RDY,E.CS1 ;LOAD EXPECTED CS1
7473  050434  012737  100100  001750          MOV     #DLT!IR,E.CS2   ;LOAD EXPECTED CS2
7474  050442  023737  001750  001710          CMP     E.CS2,T.CS2     ;CHECK IF DATA LATE SET
7475  050450  001401                          BEQ     19$             ;YES, CHECK IF CONTROLLER ERROR
7476                                                                  ;  SET
7477  050452  104016                          ERROR   16              ;ATTEMPTING TO READ SILO WHEN
7478                                                                  ;  EMPTY - CS2 INCORRECT
7479  050454  023737  001740  001700  19$:    CMP     E.CS1,T.CS1     ;CHECK IF CONTROLLER ERROR SET
7480  050462  001401                          BEQ     20$             ;YES, CLEAR CONTROLLER
7481  050464  104017                          ERROR   17              ;ATTEMPTING TO READ SILO WHEN
7482                                                                  ;  EMPTY - CS1 INCORRECT
7483  050466  012762  100000  000000  20$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7484
7485                                  ;*****************************************************************
7486                                  ;*TEST 66       ONE WORD SILO WRITE AND REG. INTERACTION (PART 1)
7487                                  ;*
7488                                  ;*      ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
7489                                  ;*      ZERO.
7490                                  ;*
7491                                  ;*      WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERACTION
7492                                  ;*      PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
7493                                  ;*      STATUS REGISTER 2.  IF NOT WAIT.
7494                                  ;*
7495                                  ;*      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
7496                                  ;*      IF CORRECT.
7497                                  ;*
7498                                  ;*      THE FOLLOWING CONFIGURATIONS ARE USED:
```

```
                                        ;*
                                        ;*      000000  000010  000200  004000  100000
                                        ;*      000001  000020  000400  010000
                                        ;*      000002  000040  001000  020000
                                        ;*      000004  000100  002000  040000
                                        ;*
                                        ;*************************************************************
050474  000004                    TST66:  SCOPE
050476  012737  000764  001000          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
050504  012737  065565  001 ?0          MOV     #EM33,EM3N      ;LOAD ERROR MESSAGE FOR PRINT OUT
050512  012737  000001  002  0          MOV     #000001,CONFIG  ;LOAD INITIAL CONFIGURATION
050520  012701  000021                  MOV     #17.,R1         ;LOAD CONFIGURATION COUNT
050524  012737  050532  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
050532                              1$:
050532  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
050540  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
050544  013762  002010  000024          MOV     CONFIG,RKDB(R2) ;WRITE DATA BUFFER
050552  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE COMMAND AND STATUS REG.1
050560  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
050566  001407                          BEQ     2$              ;YES, CHECK OTHER REGISTERS
050570  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
050576  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
050604  104003                          ERROR   3
050606  016237  000004  001126    2$:   MOV     RKBA(R2),$BDDAT ;STORE BUS ADDRESS
050614  001406                          BEQ     3$              ;CHECK IF ZERO
050616  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
050622  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
050630  104003                          ERROR   3
050632  016237  000002  001126    3$:   MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
050640  001406                          BEQ     4$              ;CHECK IF ZERO
050642  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
050646  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
050654  104003                          ERROR   3
050656  016237  000006  001126    4$:   MOV     RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG
050664  001406                          BEQ     5$              ;CHECK IF ZERO
050666  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
050672  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
050700  104003                          ERROR   3
050702  016237  000016  001126    5$:   MOV     RKASOF(R2),$BDDAT       ;STORE ATTENTION SUMMARY AND OFFSET
050710  001406                          BEQ     6$              ;CHECK IF ZERO
050712  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
050716  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
050724  104003                          ERROR   3
050726  012700  000005            6$:   MOV     #5,R0           ;LOAD COUNTER TO WAIT FOR
050732  005300                    7$:   DEC     R0              ;  OUTPUT READY
050734  001376                          BNE     7$
050736  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;STORE COMMAND AND STATUS REG 2
050744  022737  000300  001126          CMP     #IR!OR,$BDDAT   ;CHECK IF CS2 CORRECT
050752  001407                          BEQ     8$              ;YES, CONTINUE TEST
050754  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;LOAD EXPECTED CONTENTS
050762  012737  066704  001322          MOV     #EM1004,EM3N+2  ;LOAD ERROR MESSAGE
050770  104003                          ERROR   3
050772  016237  000012  001126    8$:   MOV     RKDS(R2),$BDDAT ;STORE PRIVE STATUS REG
051000  001406                          BEQ     9$              ;CHECK IF ZERO
```

```
7555  051002  005037  001124              CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
7556  051006  012737  067471  001322      MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
7557  051014  104003                      ERROR   3
7558  051016  016237  000014  001126  9$: MOV     RKER(R2),$BDDAT ;STORE ERROR REGISTER
7559  051024  001406                      BEQ     10$             ;CHECK IF ZERO
7560  051026  005037  001124              CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
7561  051032  012737  067527  001322      MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
7562  051040  104003                      ERROR   3
7563  051042  016237  000020  001126 10$: MOV     RKDCYL(R2),$BDDAT       ;STORE CYLINDER ADDRESS REG
7564  051050  001406                      BEQ     12$             ;CHECK IF ZERO
7565  051052  005037  001124              CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
7566  051056  012737  067624  001322      MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
7567  051064  104003                      ERROR   3
7568  051066  016237  000026  001126 12$: MOV     RKMR1(R2),$BDDAT        ;STORE MAINTENANCE REG. 1
7569  051074  012737  002000  001124      MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
7570  051102  032737  020000  001126      BIT     #ECCW,$BDDAT
7571  051110  001403                      BEQ     13$
7572  051112  052737  020000  001124      BIS     #ECCW,$GDDAT
7573  051120  023737  001124  001126 13$: CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
7574  051126  001404                      BEQ     14$             ;YES,CONTINUE
7575  051130  012737  067651  001322      MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
7576  051136  104003                      ERROR   3
7577  051140  016237  000032  001126 14$: MOV     RKECPT(R2),$BDDAT       ;STORE ECC PATTERN REG.
7578  051146  001406                      BEQ     15$             ;CHECK IF ZERO
7579  051150  005037  001124              CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
7580  051154  012737  067727  001322      MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
7581  051162  104003                      ERROR   3
7582  051164  016237  000030  001126 15$: MOV     RKECPS(R2),$BDDAT       ;STORE ECC POSITION REG.
7583  051172  022737  004066  001126      CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION REG CORRECT
7584  051200  001407                      BEQ     16$             ;YES, CHECK DATA BUFFER
7585  051202  012737  004066  001124      MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
7586  051210  012737  067702  001322      MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
7587  051216  104003                      ERROR   3
7588  051220  016237  000024  001126 16$: MOV     RKDB(R2),$BDDAT ;STORE DATA BUFFER
7589  051226  023737  002010  001126      CMP     CONFIG,$BDDAT   ;CHECK FOR CORRECT CONTENTS
7590  051234  001407                      BEQ     17$             ;YES, CHECK IF FINISHED
7591  051236  013737  002010  001124      MOV     CONFIG,$GDDAT   ;LOAD EXPECTED CONTENTS
7592  051244  012737  067044  001322      MOV     #EM1008,EM3N+2  ;LOAD ERROR MESSAGE
7593  051252  104003                      ERROR   3
7594  051254  104415              17$:    SCOP1                   ;CHECK IF LOOP ON ERROR
7595  051256  000241                      CLC                     ;SHIFT IN ZERO
7596  051260  006137  002010              ROL     CONFIG
7597  051264  005301                      DEC     R1              ;CHECK IF FINISHED
7598  051266  001402                      BEQ     TST67           ;;YES, GO ON TO NEXT TEST
7599  051270  000137  050532              JMP     1$
7600
7601                                  ;;*****************************************************************
7602                                  ;*TEST 67       ONE WORD SILO WRITE AND REG. INTERACTION (PART 2)
7603                                  ;*
7604                                  ;*      ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
7605                                  ;*      ZERO.
7606                                  ;*
7607                                  ;*      WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERACTION
7608                                  ;*      PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
7609                                  ;*      STATUS REGISTER 2.  IF NOT WAIT.
7610                                  ;*
```

```
                                ;*      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
                                ;*      IF CORRECT.
                                ;*
                                ;*      THE FOLLOWING CONFIGURATIONS ARE USED:
                                ;*
                                ;*      177777  177767  177577  173777  077777
                                ;*      177776  177757  177377  167777
                                ;*      177775  177737  176777  157777
                                ;*      177773  177677  175777  137777
                                ;*
                                ;*****************************************************************
051274  000004                  TST67:  SCOPE
051276  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
051304  012737  065565  001320          MOV     #EM33,EM3N      ;LOAD ERROR MESSAGE FOR PRINT OUT
051312  012737  177776  002010          MOV     #177776,CONFIG  ;LOAD INITIAL CONFIGURATION
051320  012701  000021                  MOV     #17.,R1         ;LOAD CONFIGURATION COUNT
051324  012737  051332  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
                                1$:
051332  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
051340  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
051344  013762  002010  000024          MOV     CONFIG,RKDB(R2) ;WRITE DATA BUFFER
051352  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE COMMAND AND STATUS REG.1
051360  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
051366  001407                          BEQ     2$              ;YES, CHECK OTHER REGISTERS
051370  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
051376  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
051404  104003                          ERROR   3
051406  016237  000004  001126  2$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADDRESS
051414  001406                          BEQ     3$              ;CHECK IF ZERO
051416  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
051422  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
051430  104003                          ERROR   3
051432  016237  000002  001126  3$:     MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
051440  001406                          BEQ     4$              ;CHECK IF ZERO
051442  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
051446  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
051454  104003                          ERROR   3
051456  016237  000006  001126  4$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG
051464  001406                          BEQ     5$              ;CHECK IF ZERO
051466  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
051472  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
051500  104003                          ERROR   3
051502  016237  000016  001126  5$:     MOV     RKASOF(R2),$BDDAT       ;STORE ATTENTION SUMMARY AND OFFSET
051510  001406                          BEQ     6$              ;CHECK IF ZERO
051512  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
051516  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
051524  104003                          ERROR   3
051526  012700  000005          6$:     MOV     #5,R0           ;LOAD COUNTER TO WAIT FOR
051532  005300                  7$:     DEC     R0              ;  OUTPUT READY
051534  001376                          BNE     7$
051536  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;STORE COMMAND AND STATUS REG 2
051544  022737  000300  001126          CMP     #IR!OR,$BDDAT   ;CHECK IF CS2 CORRECT
051552  001407                          BEQ     8$              ;YES, CONTINUE TEST
051554  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;LOAD EXPECTED CONTENTS
```

```
 7667  051562  012737  066704  001322          MOV     #EM1004,EM3N+2  ;LOAD ERROR MESSAGE
 7668  051570  104003                          ERROR   3
 7669  051572  016237  000012  001126  8$:     MOV     RKDS(R2),$BDDAT ;STORE PRIVE STATUS REG
 7670  051600  001406                          BEQ     9$              ;CHECK IF ZERO
 7671  051602  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7672  051606  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 7673  051614  104003                          ERROR   3
 7674  051616  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REGISTER
 7675  051624  001406                          BEQ     10$             ;CHECK IF ZERO
 7676  051626  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7677  051632  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
 7678  051640  104003                          ERROR   3
 7679  051642  016237  000020  001126  10$:    MOV     RKDCYL(R2),$BDDAT       ;STORE CYLINDER ADDRESS REG
 7680  051650  001406                          BEQ     12$             ;CHECK IF ZERO
 7681  051652  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7682  051656  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
 7683  051664  104003                          ERROR   3
 7684  051666  016237  000026  001126  12$:    MOV     RKMR1(R2),$BDDAT        ;STORE MAINTENANCE REG. 1
 7685  051674  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
 7686  051702  032737  020000  001126          BIT     #ECCW,$BDDAT
 7687  051710  001403                          BEQ     13$
 7688  051712  052737  020000  001124          BIS     #ECCW,$GDDAT
 7689  051720  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
 7690  051726  001404                          BEQ     14$             ;YES,CONTINUE
 7691  051730  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
 7692  051736  104003                          ERROR   3
 7693  051740  016237  000032  001126  14$:    MOV     RKECPT(R2),$BDDAT       ;STORE ECC PATTERN REG.
 7694  051746  001406                          BEQ     15$             ;CHECK IF ZERO
 7695  051750  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7696  051754  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
 7697  051762  104003                          ERROR   3
 7698  051764  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT       ;STORE ECC POSITION REG.
 7699  051772  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION REG CORRECT
 7700  052000  001407                          BEQ     16$             ;YES, CHECK DATA BUFFER
 7701  052002  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
 7702  052010  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
 7703  052016  104003                          ERROR   3
 7704  052020  016237  000024  001126  16$:    MOV     RKDB(R2),$BDDAT ;STORE DATA BUFFER
 7705  052026  023737  002010  001126          CMP     CONFIG,$BDDAT   ;CHECK FOR CORRECT CONTENTS
 7706  052034  001407                          BEQ     17$             ;YES, CHECK IF FINISHED
 7707  052036  013737  002010  001124          MOV     CONFIG,$GDDAT   ;LOAD EXPECTED CONTENTS
 7708  052044  012737  067044  001322          MOV     #EM1008,EM3N+2  ;LOAD ERROR MESSAGE
 7709  052052  104003                          ERROR   3
 7710  052054  104415                  17$:    SCOP1                   ;CHECK IF LOOP ON ERROR
 7711  052056  000261                          SEC                     ;SHIFT IN ONE
 7712  052060  006137  002010                  ROL     CONFIG
 7713  052064  005301                          DEC     R1              ;CHECK IF FINISHED
 7714  052066  001402                          BEQ     TST70           ;;YES, GO ON TO NEXT TEST
 7715  052070  000137  051332                  JMP     1$
 7716
 7717                                  ;;*****************************************************************
 7718                                  ;*TEST 70       ONE WORD SILO WRITE AND REG. INTERACTION (PART 3)
 7719                                  ;*
 7720                                  ;*      ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
 7721                                  ;*      ZERO.
 7722                                  ;*
```

```
                                        ;*      WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERACTION
                                        ;*      PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
                                        ;*      STATUS REGISTER 2.  IF NOT WAIT.
                                        ;*
                                        ;*      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
                                        ;*      IF CORRECT.
                                        ;*
                                        ;*      THE FOLLOWING CONFIGURATIONS ARE USED:
                                        ;*
                                        ;*      000001  000037  000777  017777  000000
                                        ;*      000003  000077  001777  037777
                                        ;*      000007  000177  003777  077777
                                        ;*      000017  000377  007777  177777
                                        ;*
                                        ;**************************************************************
052074  000004                  TST70:  SCOPE
052076  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
052104  012737  065565  001320          MOV     #EM33,EM3N      ;LOAD ERROR MESSAGE FOR PRINT OUT
052112  005037  002010                  CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
052116  012701  000021                  MOV     #17.,R1         ;LOAD CONFIGURATION COUNT
052122  012737  052130  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP
052130                          1$:
052130  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
052136  005062  000002                  CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
052142  013762  002010  000024          MOV     CONFIG,RKDB(R2) ;WRITE DATA BUFFER
052150  016237  000000  001126          MOV     RKCS1(R2),$BDDAT        ;STORE COMMAND AND STATUS REG.1
052156  022737  000200  001126          CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
052164  001407                          BEQ     2$              ;YES, CHECK OTHER REGISTERS
052166  012737  000200  001124          MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
052174  012737  067314  001322          MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
052202  104003                          ERROR   3
052204  016237  000004  001126  2$:     MOV     RkBA(R2),$BDDAT ;STORE BUS ADDRESS
052212  001406                          BEQ     3$              ;CHECK IF ZERO
052214  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
052220  012737  067365  001322          MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
052226  104003                          ERROR   3
052230  016237  000002  001126  3$:     MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
052236  001406                          BEQ     4$              ;CHECK IF ZERO
052240  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
052244  012737  067335  001322          MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
052252  104003                          ERROR   3
052254  016237  000006  001126  4$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG
052262  001406                          BEQ     5$              ;CHECK IF ZERO
052264  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
052270  012737  067412  001322          MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
052276  104003                          ERROR   3
052300  016237  000016  001126  5$:     MOV     RKASOF(R2),$BDDAT       ;STORE ATTENTION SUMMARY AND OFFSET
052306  001406                          BEQ     6$              ;CHECK IF ZERO
052310  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
052314  012737  067556  001322          MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
052322  104003                          ERROR   3
052324  012700  000005          6$:     MOV     #5,R0           ;LOAD COUNTER TO WAIT FOR
052330  005300                  7$:     DEC     R0              ;  OUTPUT READY
052332  001376                          BNE     7$
```

```
 7779  052334  016237  000010  001126          MOV     RKCS2(R2),$BDDAT        ;STORE COMMAND AND STATUS REG 2
 7780  052342  022737  000300  001126          CMP     #IR!OR,$BDDAT   ;CHECK IF CS2 CORRECT
 7781  052350  001407                          BEQ     8$              ;YES, CONTINUE TEST
 7782  052352  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;LOAD EXPECTED CONTENTS
 7783  052360  012737  066704  001322          MOV     #EM1004,EM3N+2  ;LOAD ERROR MESSAGE
 7784  052366  104003                          ERROR   3
 7785  052370  016237  000012  001126  8$:     MOV     RKDS(R2),$BDDAT ;STORE PRIVE STATUS REG
 7786  052376  001406                          BEQ     9$              ;CHECK IF ZERO
 7787  052400  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7788  052404  012737  067471  001322          MOV     #EM1022,EM3N+2  ;LOAD ERROR MESSAGE
 7789  052412  104003                          ERROR   3
 7790  052414  016237  000014  001126  9$:     MOV     RKER(R2),$BDDAT ;STORE ERROR REGISTER
 7791  052422  001406                          BEQ     10$             ;CHECK IF ZERO
 7792  052424  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7793  052430  012737  067527  001322          MOV     #EM1023,EM3N+2  ;LOAD ERROR MESSAGE
 7794  052436  104003                          ERROR   3
 7795  052440  016237  000020  001126  10$:    MOV     RKDCYL(R2),$BDDAT       ;STORE CYLINDER ADDRESS REG
 7796  052446  001406                          BEQ     12$             ;CHECK IF ZERO
 7797  052450  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7798  052454  012737  067624  001322          MOV     #EM1025,EM3N+2  ;LOAD ERROR MESSAGE
 7799  052462  104003                          ERROR   3
 7800  052464  016237  000026  001126  12$:    MOV     RKMR1(R2),$BDDAT        ;STORE MAINTENANCE REG. 1
 7801  052472  012737  002000  001124          MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
 7802  052500  032737  020000  001126          BIT     #ECCW,$BDDAT
 7803  052506  001403                          BEQ     13$
 7804  052510  052737  020000  001124          BIS     #ECCW,$GDDAT
 7805  052516  023737  001124  001126  13$:    CMP     $GDDAT,$BDDAT   ;CHECK IF MR1 CORRECT
 7806  052524  001404                          BEQ     14$             ;YES,CONTINUE
 7807  052526  012737  067651  001322          MOV     #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
 7808  052534  104003                          ERROR   3
 7809  052536  016237  000032  001126  14$:    MOV     RKECPT(R2),$BDDAT       ;STORE ECC PATTERN REG.
 7810  052544  001406                          BEQ     15$             ;CHECK IF ZERO
 7811  052546  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
 7812  052552  012737  067727  001322          MOV     #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
 7813  052560  104003                          ERROR   3
 7814  052562  016237  000030  001126  15$:    MOV     RKECPS(R2),$BDDAT       ;STORE ECC POSITION REG.
 7815  052570  022737  004066  001126          CMP     #4066,$BDDAT    ;CHECK IF ECC POSITION REG CORRECT
 7816  052576  001407                          BEQ     16$             ;YES, CHECK DATA BUFFER
 7817  052600  012737  004066  001124          MOV     #4066,$GDDAT    ;LOAD EXPECTED CONTENTS
 7818  052606  012737  067702  001322          MOV     #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
 7819  052614  104003                          ERROR   3
 7820  052616  016237  000024  001126  16$:    MOV     RKDB(R2),$BDDAT ;STORE DATA BUFFER
 7821  052624  023737  002010  001126          CMP     CONFIG,$BDDAT   ;CHECK FOR CORRECT CONTENTS
 7822  052632  001407                          BEQ     17$             ;YES, CHECK IF FINISHED
 7823  052634  013737  002010  001124          MOV     CONFIG,$GDDAT   ;LOAD EXPECTED CONTENTS
 7824  052642  012737  067044  001322          MOV     #EM1008,EM3N+2  ;LOAD ERROR MESSAGE
 7825  052650  104003                          ERROR   3
 7826  052652  104415                  17$:    SCOP1                   ;CHECK IF LOOP ON ERROR
 7827  052654  000261                          SEC                     ;SHIFT IN ONE
 7828  052656  006137  002010                  ROL     CONFIG
 7829  052662  005301                          DEC     R1              ;CHECK IF FINISHED
 7830  052664  001402                          BEQ     TST71           ;;YES, GO ON TO NEXT TEST
 7831  052666  000137  052130                  JMP     1$
 7832
 7833                                  ;;*****************************************************************
 7834                                  ;*TEST 71        ONE WORD SILO WRITE AND REG. INTERACTION (PART 4)
```

```
                                        ;*
                                        ;*      ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
                                        ;*      ZERO.
                                        ;*
                                        ;*      WRITE ONE WORD IN SILO.  CHECK ALL REGISTERS FOR INTERACTION
                                        ;*      PROBLEMS.  CHECK THAT OUTPUT READY IS SET IN COMMAND AND
                                        ;*      STATUS REGISTER 2.  IF NOT WAIT.
                                        ;*
                                        ;*      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
                                        ;*      IF CORRECT.
                                        ;*
                                        ;*      THE FOLLOWING CONFIGURATIONS ARE USED:
                                        ;*
                                        ;*      100000  174000  177600  177770  000000
                                        ;*      140000  176000  177700  177774
                                        ;*      160000  177000  177740  177776
                                        ;*      170000  177400  177760  177777
                                        ;*
                                        ;**************************************************************************
052672  000004                    TST71:  SCOPE
052674  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
052702  012737  065565  001320            MOV     #EM33,EM3N      ;LOAD ERROR MESSAGE FOR PRINT OUT
052710  005037  002010                    CLR     CONFIG          ;LOAD INITIAL CONFIGURATION
052714  012701  000021                    MOV     #17.,R1         ;LOAD CONFIGURATION COUNT
052720  012737  052726  001110            MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                  ;  SUBTEST LOOP
052726                              1$:
052726  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
052734  005062  000002                    CLR     RKWC(R2)        ;CLEAR WORD COUNT REG.
052740  013762  002010  000024            MOV     CONFIG,RKDB(R2) ;WRITE DATA BUFFER
052746  016237  000000  001126            MOV     RKCS1(R2),$BDDAT        ;STORE COMMAND AND STATUS REG.1
052754  022737  000200  001126            CMP     #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
052762  001407                            BEQ     2$              ;YES, CHECK OTHER REGISTERS
052764  012737  000200  001124            MOV     #RDY,$GDDAT     ;LOAD EXPECTED CONTENTS
052772  012737  067314  001322            MOV     #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
053000  104003                            ERROR   3
053002  016237  000004  001126    2$:     MOV     RKBA(R2),$BDDAT ;STORE BUS ADDRESS
053010  001406                            BEQ     3$              ;CHECK IF ZERO
053012  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
053016  012737  067365  001322            MOV     #EM1019,EM3N+2  ;LOAD ERROR MESSAGE
053024  104003                            ERROR   3
053026  016237  000002  001126    3$:     MOV     RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
053034  001406                            BEQ     4$              ;CHECK IF ZERO
053036  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
053042  012737  067335  001322            MOV     #EM1018,EM3N+2  ;LOAD ERROR MESSAGE
053050  104003                            ERROR   3
053052  016237  000006  0011..    4$:     MOV     RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG
053060  001406                            BEQ     5$              ;CHECK IF ZERO
053062  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
053066  012737  067412  001322            MOV     #EM1020,EM3N+2  ;LOAD ERROR MESSAGE
053074  104003                            ERROR   3
053076  016237  000016  001126    5$:     MOV     RKASOF(R2),$BDDAT       ;STORE ATTENTION SUMMARY AND OFFSET
053104  001406                            BEQ     6$              ;CHECK IF ZERO
053106  005037  001124                    CLR     $GDDAT          ;LOAD EXPECTED CONTENTS
053112  012737  067556  001322            MOV     #EM1024,EM3N+2  ;LOAD ERROR MESSAGE
```

```
7891  053120  104003                                  ERROR   3
7892  053122  012700  000005                  6$:     MOV     #5,R0               ;LOAD COUNTER TO WAIT FOR
7893  053126  005300                          7$:     DEC     R0                  ;  OUTPUT READY
7894  053130  001376                                  BNE     7$
7895  053132  016237  000010  001126                  MOV     RKCS2(R2),$BDDAT            ;STORE COMMAND AND STATUS REG 2
7896  053140  022737  000300  001126                  CMP     #IR!OR,$BDDAT       ;CHECK IF CS2 CORRECT
7897  053146  001407                                  BEQ     8$                  ;YES, CONTINUE TEST
7898  053150  012737  000300  001124                  MOV     #IR!OR,$GDDAT       ;LOAD EXPECTED CONTENTS
7899  053156  012737  066704  001322                  MOV     #EM1004,EM3N+2      ;LOAD ERROR MESSAGE
7900  053164  104003                                  ERROR   3
7901  053166  016237  000012  001126          8$:     MOV     RKDS(R2),$BDDAT     ;STORE PRIVE STATUS REG
7902  053174  001406                                  BEQ     9$                  ;CHECK IF ZERO
7903  053176  005037  001124                          CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7904  053202  012737  067471  001322                  MOV     #EM1022,EM3N+2      ;LOAD ERROR MESSAGE
7905  053210  104003                                  ERROR   3
7906  053212  016237  000014  001126          9$:     MOV     RKER(R2),$BDDAT     ;STORE ERROR REGISTER
7907  053220  001406                                  BEQ     10$                 ;CHECK IF ZERO
7908  053222  005037  001124                          CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7909  053226  012737  067527  001322                  MOV     #EM1023,EM3N+2      ;LOAD ERROR MESSAGE
7910  053234  104003                                  ERROR   3
7911  053236  016237  000020  001126          10$:    MOV     RKDCYL(R2),$BDDAT           ;STORE CYLINDER ADDRESS REG
7912  053244  001406                                  BEQ     12$                 ;CHECK IF ZERO
7913  053246  005037  001124                          CLR     $GDDA               ;LOAD EXPECTED CONTENTS
7914  053252  012737  067624  001322                  MOV     #EM1025,EM3N+2      ;LOAD ERROR MESSAGE
7915  053260  104003                                  ERROR   3
7916  053262  016237  000026  001126          12$:    MOV     RKMR1(R2),$BDDAT            ;STORE MAINTENANCE REG. 1
7917  053270  012737  002000  001124                  MOV     #MEWD,$GDDAT        ;LOAD EXPECTED MR1
7918  053276  032737  020000  001126                  BIT     #ECCW,$BDDAT
7919  053304  001403                                  BEQ     13$
7920  053306  052737  020000  001124                  BIS     #ECCW,$GDDAT
7921  053314  023737  001124  001126          13$:    CMP     $GDDAT,$BDDAT       ;CHECK IF MR1 CORRECT
7922  053322  001404                                  BEQ     14$                 ;YES CONTINUE
7923  053324  012737  067651  001322                  MOV     #EM1026,EM3N+2      ;LOAD ERROR MESSAGE
7924  053332  104003                                  ERROR   3
7925  053334  016237  000032  001126          14$:    MOV     RKECPT(R2),$BDDAT           ;STORE ECC PATTERN REG.
7926  053342  001406                                  BEQ     15$                 ;CHECK IF ZERO
7927  053344  005037  001124                          CLR     $GDDAT              ;LOAD EXPECTED CONTENTS
7928  053350  012737  067727  001322                  MOV     #EM1030,EM3N+2      ;LOAD ERROR MESSAGE
7929  053356  104003                                  ERROR   3
7930  053360  016237  000030  001126          15$:    MOV     RKECPS(R2),$BDDAT           ;STORE ECC POSITION REG.
7931  053366  022737  004066  001126                  CMP     #4066,$BDDAT        ;CHECK IF ECC POSITION REG CORRECT
7932  053374  001407                                  BEQ     16$                 ;YES, CHECK DATA BUFFER
7933  053376  012737  004066  001124                  MOV     #4066,$GDDAT        ;LOAD EXPECTED CONTENTS
7934  053404  012737  067702  001322                  MOV     #EM1029,EM3N+2      ;LOAD ERROR MESSAGE
7935  053412  104003                                  ERROR   3
7936  053414  016237  000024  001126          16$:    MOV     RKDB(R2),$BDDAT     ;STORE DATA BUFFER
7937  053422  023737  002010  001126                  CMP     CONFIG,$BDDAT       ;CHECK FOR CORRECT CONTENTS
7938  053430  001407                                  BEQ     17$                 ;YES, CHECK IF FINISHED
7939  053432  013737  002010  001124                  MOV     CONFIG,$GDDAT       ;LOAD EXPECTED CONTENTS
7940  053440  012737  067044  001322                  MOV     #EM1008,EM3N+2      ;LOAD ERROR MESSAGE
7941  053446  104003                                  ERROR   3
7942  053450  104415                          17$:    SCOP1                       ;CHECK IF LOOP ON ERROR
7943  053452  000261                                  SEC                         ;SHIFT IN ONE
7944  053454  006037  002010                          ROR     CONFIG
7945  053460  005301                                  DEC     R1                  ;CHECK IF FINISHED
7946  053462  001402                                  BEQ     TST72               ;;YES. GO ON TO NEXT TEST
```

```
7947  053464  000137  052726                    JMP     1$
7948
7949                                    ;;*****************************************************************
7950                                    ;*TEST 72       SILO FILL
7951                                    ;*
7952                                    ;*      THIS TEST WILL WRITE THE SILO WITH 66 DIFFERENT PATTERNS
7953                                    ;*      CHECK INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH
7954                                    ;*      WORD WRITTEN.  IT WILL THEN READ ALL 66 WORDS BACK CHECKING
7955                                    ;*      CONTENTS, INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH
7956                                    ;*      WORD READ.  AN EXTRA READ IS THEN DONE TO MAKE SURE THE
7957                                    ;*      SILO IS EMPTY.
7958                                    ;*
7959                                    ;*****************************************************************
7960  053470  000004                    T$T72:  SCOPE
7961  053472  012737  000764  001200            MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
7962  053500  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE
7963  053504  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7964  053512  012703  070754                    MOV     #SILO,R3        ;LOAD ADDRESS OF INPUT DATA
7965  053516  005037  002022                    CLR     SILCNT          ;CLEAR SILO COUNT
7966  053522  012362  000024                    MOV     (R3)+,RKDB(R2)  ;LOAD SILO
7967  053526  012700  000020                    MOV     #20,R0          ;WAIT FOR OUTPUT READY
7968  053532  005300                    1$:     DEC     R0
7969  053534  001376                            BNE     1$
7970  053536  016237  000000  001700            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
7971  053544  016237  000010  001710            MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
7972  053552  012737  000200  001740            MOV     #RDY,E.CS1      ;LOAD EXPECTED COMMAND AND STATUS REG. 1
7973  053560  012737  000300  001750            MOV     #IR!OR,E.CS2    ;LOAD EXPECTED COMMAND AND STATUS REG 2
7974  053566  023737  001740  001700            CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
7975  053574  001401                            BEQ     2$              ;YES, CONTINUE
7976  053576  104024                            ERROR   24              ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT
7977  053600  023737  001750  001710  2$:       CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
7978  053606  001401                            BEQ     3$              ;YES,CONTINUE
7979  053610  104025                            ERROR   25              ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
7980  053612  005237  002022            3$:     INC     SILCNT          ;INCREMENT SILO COUNT
7981  053616  012362  000024                    MOV     (R3)+,RKDB(R2)  ;LOAD SILO
7982  053622  016237  000000  001700            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7983  053630  016237  000010  001710            MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
7984  053636  023737  001740  001700            CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
7985  053644  001401                            BEQ     4$              ;YES, CONTINUE
7986  053646  104024                            ERROR   24              ;ATTEMPTING TO WRITE SILO- CS1 INCORRECT
7987  053650  023737  001750  001710  4$:       CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
7988  053656  001401                            BEQ     5$              ;YES, CONTINUE
7989  053660  104025                            ERROR   25              ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
7990  053662  022737  000100  002022  5$:       CMP     #64.,SILCNT     ;CHECK IF ALL EXCEPT LAST WORD WRITTEN
7991  053670  001350                            BNE     3$              ;YES, GO WRITE NEXT WORD
7992  053672  005237  002022                    INC     SILCNT          ;INCREMENT SILO COUNT
7993  053676  011362  000024                    MOV     (R3),RKDB(R2)   ;LOAD SILO
7994  053702  016237  000000  001700            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7995  053710  016237  000010  001710            MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
7996  053716  012737  000200  001750            MOV     #OR,E.CS2       ;LOAD EXPECTED CS2
7997  053724  023737  001740  001700            CMP     E.CS1,T CS1     ;CHECK IF CS1 CORRECT
7998  053732  001401                            BEQ     6$              ;YES, CONTINUE
7999  053734  104024                            ERROR   24              ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT
8000  053736  023737  001750  001710  6$:       CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
8001  053744  001401                            BEQ     7$              ;YES, UNLOAD SILO
8002  053746  104025                            ERROR   25              ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
```

```
 8003  053750  005037  002022          7$:     CLR     SILCNT          ;CLEAR SILO COUNT
 8004  053754  012703  070754                  MOV     #SILO,R3        ;LOAD ADDRESS OF INPUT DATA
 8005  053760  012737  000300  001750          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
 8006  053766  016237  000024  001126          MOV     RKDB(R2),$BDDAT ;STORE NEXT WORD ON SILO
 8007  053774  012700  000020                  MOV     #20,R0          ;WAIT FOR INPUT READY
 8008  054000  005300                  8$:     DEC     R0
 8009  054002  001376                          BNE     8$
 8010  054004  000403                          BR      10$             ;CONTINUE WITH TEST
 8011
 8012  054006  016237  000024  001126  9$:     MOV     RKDB(R2),$BDDAT ;STORE NEXT WORD ON SILO
 8013  054014  016237  000000  001700  10$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
 8014  054022  016237  000010  001710          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
 8015  054030  012337  001124                  MOV     (R3)+,$GDDAT    ;LOAD EXPECTED DATA
 8016  054034  023737  001740  001700          CMP     E.CS1,T.CS1     ;CHECK IF CONTROLLER ERROR RESET
 8017  054042  001401                          BEQ     11$             ;YES, CONTINUE
 8018  054044  104026                          ERROR   26              ;ATTEMPTING TO READ SILO-CS1 INCORRECT
 8019  054046  023737  001750  001710  11$:    CMP     E.CS2,T.CS2     ;CHECK IF OUTPUT READY AND INPUT READY SET
 8020  054054  001401                          BEQ     12$             ;YES,CHECK DATA
 8021  054056  104027                          ERROR   27              ;ATTEMPTING TO REPO SILO-CS2 INCORRECT
 8022  054060  023737  001124  001126  12$:    CMP     $GDDAT,$BDDAT   ;CHECK IF SILO CONTENTS CORRECT
 8023  054066  001401                          BEQ     13$             ;YES, CHECK IF LAST WORD
 8024  054070  104030                          ERROR   30              ;ATTEMPTING TO READ SILO-RKDB INCORRECT
 8025  054072  005237  002022          13$:    INC     SILCNT          ;SET UP FOR NEXT WORD
 8026  054076  022737  000101  002022          CMP     #65.,SILCNT     ;CHECK READY FOR LAST WORD
 8027  054104  101340                          BHI     9$              ;NO, READ NEXT WORD
 8028  054106  103404                          BLO     14$             ;CHECK IF SILO EMPTY
 8029  054110  012737  000100  001750          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
 8030  054116  000733                          BR      9$              ;READ LAST WORD
 8031
 8032  054120  005762  000024          14$:    TST     RKDB(R2)        ;READ DATA BUFFER
 8033  054124  016237  000010  001710          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
 8034  054132  016237  000000  001700          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
 8035  054140  012737  100100  001750          MOV     #DLT!IR,E.CS2   ;LOAD EXPECTED CS2
 8036  054146  012737  100200  001740          MOV     #CERR!RDY,E.CS1 ;LOAD EXPECTED CS1
 8037  054154  023737  001750  001710          CMP     E.CS2,T.CS2     ;CHECK FOR DATA LATE SET
 8038  054162  001401                          BEQ     15$             ;YES, CHECK FOR CONTROLLER ERROR
 8039  054164  104006                          ERROR   6               ;CS2 INCORRECT AFTER LEAVING EMPTY
 8040                                                                  ;  SILO
 8041  054166  023737  001740  001700  15$:    CMP     E.CS1,T.CS1     ;CHECK FOR CONTROLLER ERROR
 8042  054174  001401                          BEQ     16$             ;YES, CLEAR DATA LATE
 8043  054176  104017                          ERROR   17              ;CS1 INCORRECT AFTER READING EMPTY
 8044                                                                  ;  SILO
 8045  054200  012762  100000  000000  16$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 8046
 8047                                  ;;*****************************************************************
 8048                                  ;*TEST 73        SILO CAPACITY DATA LATE
 8049                                  ;*
 8050                                  ;*       WRITE 67 WORDS IN THE SILO AND MAKE SURE DATA LATE ONLY
 8051                                  ;*       OCCURS ON THE 67TH WORD.  CLEAR RK611 WITH CONTROLLER CLEAR.
 8052                                  ;*       CHECK INPUT READY AND OUTPUT READY FOR INITIALIZED STATE.
 8053                                  ;*
 8054                                  ;;*****************************************************************
 8055  054206  000004                  TST73:  SCOPE
 8056  054210  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
 8057  054216  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
 8058  054222  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
```

```
8059  054230  012700  000103                  MOV     #67.,R0          ;LOAD 67 WORDS INTO SILO CAUSING
8060  054234  012762  177777  000024  1$:     MOV     #177777,RKDB(R2)         ;  DATA LATE
8061  054242  005300                          DEC     R0
8062  054244  001373                          BNE     1$
8063  054246  016237  000010  001710          MOV     RKCS2(R2),T.CS2  ;STORE COMMAND AND STATUS REG 2
8064
8065  054254  016237  000000  001700          MOV     RKCS1(R2),T.CS1  ;STORE COMMAND AND STATUS REG. 1
8066  054262  012737  100200  001750          MOV     #DLT!OR,E.CS2    ;LOAD EXPECTED CS2
8067  054270  012737  100200  001740          MOV     #CERR!RDY,E.CS1  ;LOAD EXPECTED CS1
8068  054276  023737  001750  001710          CMP     E.CS2,T.CS2      ;CHECK IF CS2 CORRECT (DLT)
8069  054304  001401                          BEQ     2$               ;YES, CONTINUE
8070  054306  104031                          ERROR   31               ;CS2 INCORRECT WHEN LOADING FULL SILO
8071  054310  023737  001740  001700  2$:     CMP     E.CS1,T.CS1      ;CHECK IF CS1 CORRECT (CERR)
8072  054316  001401                          BEQ     3$               ;YES, CLEAR CONTROLLER
8073  054320  104032                          ERROR   32               ;CS1 INCORRECT WHEN LOADING FULL SILO
8074  054322  012762  100000  000000  3$:     MOV     #CCLR,RKCS1(R2)  ;CLEAR RK611 CONTROLLER
8075  054330  016237  000000  001700          MOV     RKCS1(R2),T.CS1  ;STORE COMMAND AND STATUS REG.1
8076  054336  016237  000010  001710          MOV     RKCS2(R2),T.CS2  ;STORE COMMAND AND STATUS REG. 2
8077  054344  012737  000200  001740          MOV     #RDY,E.CS1       ;LOAD EXPECTED CS1
8078  054352  012737  000100  001750          MOV     #IR,E.CS2        ;LOAD EXPECTED CS2
8079  054360  023737  001740  001700          CMP     E.CS1,T.CS1      ;CHECK IF CONTROLLER ERROR RESET
8080  054366  001401                          BEQ     4$               ;YES, CHECK IF DATA LATE RESET
8081  054370  104020                          ERROR   20               ;CS1 INCORRECT AFTER ATTEMPTING
8082                                                                   ;  TO CLEAR DATA LATE
8083  054372  023737  001750  001710  4$:     CMP     E.CS2,T.CS2      ;CHECK IF DATA LATE RESET
8084  054400  001401                          BEQ     TST74            ;YES, GO ON TO NEXT TEST
8085  054402  104021                          ERROR   21               ;CS2 INCORRECT AFTER ATTEMPTING
8086                                                                   ;  TO CLEAR DATA LATE
8087
8088                          ;*****************************************************************
8089                          ;*TEST 74       INTERRUPT DUE TO DATA LATE
8090                          ;*
8091                          ;*      ALLOW RK611 INTERRUPTS.  SET INTERRUPT ENABLE.
8092                          ;*      NOW READ ONE WORD FROM DATA BUFFER AND MAKE SURE THAT
8093                          ;*      DATA LATE CAUSES INTERRUPT.  BEFORE CLEARING ERROR ALLOW
8094                          ;*      RK611 INTERRUPTS AND MAKE SURE IT DOES NOT OCCUR AGAIN.
8095                          ;*      NOW CLEAR CONTROLLER WITH A CONTROLLER CLEAR.
8096                          ;*
8097                          ;*****************************************************************
8098  054404  000004          TST74:  SCOPE
8099  054406  012737  000764  001200          MOV     #500.,$TIMES     ;;DO 500. ITERATIONS
8100  054414  013702  001270                  MOV     $BASE,R2         ;LOAD RK611 BASE
8101  054420  012762  100000  000000          MOV     #CCLR,RKCS1(R2)  ;CLEAR RK611
8102  054426  013701  002000                  MOV     RKVEC,R1         ;LOAD VECTOR FOR EXPECTED INTERRUPT
8103  054432  012721  054504                  MOV     #10$,(R1)+
8104  054436  012711  000340                  MOV     #PR7,(R1)
8105  054442  005046                          CLR     -(SP)            ;LOAD STACK TO ALLOW ALL INTERRUPTS
8106  054444  012746  054452                  MOV     #64$,-(SP)       ;LOAD NEXT ADDRESS
8107  054450  000002                          RTI                      ;CLEAR PSW
8108
8109  054452                          64$:
8110  054452  012762  000100  000000          MOV     #IE,RKCS1(R2)    ;SET INTERRUPT ENABLE
8111  054460  005762  000024                  TST     RKDB(R2)         ;READ DATA BUFFER
8112  054464  000240                          NOP                      ;ALLOW INTERRUPT TO OCCUR
8113  054466  012746  000340                  MOV     #PR7,-(SP)       ;LOCK OUT INTERRUPTS
8114  054472  012746  054500                  MOV     #1$,-(SP)
```

```
8115  054476  000002                          RTI
8116
8117  054500  104033                  1$:     ERROR   33              ;DATA LATE DID NOT CAUSE EXPECTED INTERRUPT
8118  054502  000422                          BR      60$             ;CLEAR UP FOR NEXT TEST
8119
8120  054504  062706  000004          10$:    ADD     #4,SP           ;ADJUST STACK
8121  054510  012777  054542  125262          MOV     #15$,@RKVEC     ;LOAD VECTOR FOR EXPECTED INTERRUPT
8122  054516  005046                          CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
8123  054520  012746  054526                  MOV     #65$,-(SP)      ;LOAD NEXT ADDRESS
8124  054524  000002                          RTI                     ;CLEAR PSW
8125
8126  054526                          65$:
8127  054526  000240                          NOP                     ;ALLOW INTERRUPT TO OCCUR
8128  054530  012746  000340                  MOV     #PR7,-(SP)      ;LOCK OUT INTERRUPTS
8129  054534  012746  054550                  MOV     #60$,-(SP)      ;CLEAN UP FOR NEXT TEST
8130  054540  000002                          RTI
8131
8132  054542  062706  000004          15$:    ADD     #4,SP           ;ADJUST STACK
8133  054546  104034                          ERROR   34              ;UNEXPECTED INTERRUPT DUE TO
8134                                                                  ;  UNCLEARED CONTROLLER ERROR
8135  054550  012762  100000  000000  60$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
8136  054556  013701  002000                  MOV     RKVEC,R1        ;RESTORE TRAP CATCHER
8137  054562  010111                          MOV     R1,(R1)
8138  054564  062721  000002                  ADD     #2,(R1)+
8139  054570  005011                          CLR     (R1)
8140
8141                                  ;;*****************************************************************
8142                                  ;*TEST 75       INTERRUPT CLEARING AND DATA LATE
8143                                  ;*
8144                                  ;*      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. CREATE
8145                                  ;*      A CONTROLLER ERROR (DATA LATE) BY READING THE DATA BUFFER
8146                                  ;*      WHEN EMPTY.  CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
8147                                  ;*      SET INTERRUPT ENABLE AND LOWER PROCESSOR PRIORITY.
8148                                  ;*      MAKE SURE AN INTERRUPT DOES NOT OCCUR.
8149                                  ;*
8150                                  ;;*****************************************************************
8151  054572  000004                  TST75:  SCOPE
8152  054574  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
8153  054602  013702  001270                  MOV     $BASE,R2        ;LOAD  RK611 BASE
8154  054606  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
8155  054614  005762  000024                  TST     RKDB(R2)        ;CREATE DATA LATE
8156  054620  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR DATA LATE
8157  054626  013701  002000                  MOV     RKVEC,R1        ;LOAD VECTOR FOR UNEXPECTED INTERRUPT
8158  054632  012721  054664                  MOV     #5$,(R1)+
8159  054636  012711  000340                  MOV     #PR7,(R1)
8160  054642  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
8161  054650  005046                          CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
8162  054652  012746  054660                  MOV     #64$,-(SP)      ;LOAD NEXT ADDRESS
8163  054656  000002                          RTI                     ;CLEAR PSW
8164
8165  054660                          64$:
8166  054660  000240                          NOP                     ;ALLOW INTERRUPT TO OCCUR
8167  054662  000403                          BR      60$             ;CLEAN UP FOR NEXT TEST
8168
8169  054664  062706  000004          5$:     ADD     #4,SP           ;ADJUST STACK
8170  054670  104035                          ERROR   35              ;CONTROLLER CLEAR DID NOT CLEAR
```

```
8171                                                            ;  PENDING INTERRUPT DUE
8172                                                            ;  TO CONTROLLER ERROR
8173  054672  013701  002000          60$:    MOV     RKVEC,R1      ;RESTORE TRAP CATCHER
8174  054676  010111                  MOV     R1,(R1)
8175  054700  062721  000002          ADD     #2,(R1)+
8176  054704  005011                  CLR     (R1)
8177
8178                                  ;;*****************************************************************
8179                                  ;*TEST 76        INTERRUPT ENABLE AND DATA LATE
8180                                  ;*
8181                                  ;*      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.  ALLOW
8182                                  ;*      RK611 INTERRUPTS.  READ DATA BUFFER TO GENERATE INTERRUPT
8183                                  ;*      PENDING.  MAKE SURE INTERRUPT DOES NOT OCCUR.
8184                                  ;*
8185                                  ;*      NOW SET INTERRUPT ENABLE AND MAKE SURE INTERRUPTS OCCURS.
8186                                  ;*
8187                                  ;;*****************************************************************
8188  054706  000004                  TST76:  SCOPE
8189  054710  012737  000764  001200          MOV     #500.,$TIMES    ;;DO 500. ITERATIONS
8190  054716  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
8191  054722  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
8192  054730  013701  002000                  MOV     RKVEC,R1        ;LOAD VECTOR FOR UNEXPECTED INTERRUPT
8193  054734  012721  055016                  MOV     #10$,(R1)+
8194  054740  012711  000340                  MOV     #PR7,(R1)
8195  054744  005046                          CLR     -(SP)           ;LOAD STACK TO ALLOW ALL INTERRUPTS
8196  054746  012746  054754                  MOV     #64$,-(SP)      ;LOAD NEXT ADDRESS
8197  054752  000002                          RTI                     ;CLEAR PSW
8198
8199  054754                          64$:
8200  054754  005762  000024                  TST     RKDB(R2)        ;READ DATA BUFFER (GENERATE DATA LATE)
8201  054760  000240                          NOP                     ;ALLOW INTERRUPT TO OCCUR
8202  054762  012777  055026  125010          MOV     #15$,@RKVEC     ;LOAD VECTOR FOR EXPECTED INTERRUPT
8203  054770  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
8204  054776  000240                          NOP                     ;ALLOW INTERRUPT TO OCCUR
8205  055000  012746  000340                  MOV     #PR7,-(SP)      ;LOCK OUT INTERRUPTS
8206  055004  012746  055012                  MOV     #1$,-(SP)
8207  055010  000002                          RTI
8208
8209
8210  055012  104037                  1$:     ERROR   37              ;INTERRUPT DID NOT OCCUR WHEN
8211                                                                  ;  INTERRUPT ENABLE SET
8212  055014  000406                          BR      20$             ;CLEAN UP FOR NEXT TEST
8213
8214  055016  062706  000004          10$:    ADD     #4,SP           ;ADJUST STACK
8215  055022  104036                  ERROR   36                      ;CONTROLLER ERROR CAUSED INTERRUPT
8216                                                                  ;  WITH INTERRUPT ENABLE RESET
8217
8218  055024  000402                          BR      20$             ;CLEAN UP FOR NEXT TEST
8219
8220  055026  062706  000004          15$:    ADD     #4,SP           ;ADJUST STACK
8221  055032  012762  100000  000000  20$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
8222  055040  013701  002000                  MOV     RKVEC,R1        ;RESTORE TRAP CATCHER
8223  055044  010111                          MOV     R1,(R1)
8224  055046  062721  000002                  ADD     #2,(R1)+
8225  055052  005011                          CLR     (R1)
```

```
                                        END OF PASS ROUTINE

                                        .SBTTL  END OF PASS ROUTINE

                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
                                        ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO NEWPAS

055054                                  $EOP:
                                                SCOPE
000004                                          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
005037  001102                                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
005237  001200                                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
042737  100000  001200                        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
005327                                          DEC     (PC)+           ;;LOOP?
000001                                  $EOPCT: .WORD   1
003063                                          BGT     $DOAGN          ;;YES
012737                                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
000001                                  $ENDCT: .WORD   1
055102                                          $EOPCT
104401  055122                                  TYPE    ,65$            ;;TYPE ASCIZ STRING
000407                                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <12><15>/END PASS #/
055140                                  64$:
013746  001200                                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                        ;;TYPE PASS NUMBER
104405                                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
104401  055154                                  TYPE    ,67$            ;;TYPE ASCIZ STRING
000421                                          BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
055216                                  66$:
013746  001112                                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                        ;;TOTAL NUMBER OF ERRORS
104405                                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
104401  001211                                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
005037  001112                                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
001405                                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
000005                                          RESET                   ;;CLEAR THE WORLD
004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
000240                                          NOP                     ;;SAVE ROOM
000240                                          NOP                     ;;FOR
000240                                          NOP                     ;;ACT11
055254                          $DOAGN:
000137                                  JMP     @(PC)+          ;;RETURN
003040                          $RTNAD: .WORD   NEWPAS
055260      377     377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
055264                                  .EVEN

                                        .SBTTL  CHECK FOR MEMORY CHECK ENABLE

055264  012737  055336  000004 CHKPAR:  MOV     #20$,ERRVEC     ;SET VECTOR ;FOR MEMORY PARTITY CHECK
055272  012737  000340  000006          MOV     #PR7,ERRVEC+2
055300  012703  172100                  MOV     #MEMBAS,R3      ;LOAD REGISTER TO DETERMINE IF
                                                                ;   MEMORY CHECK ENABLE AVAILIZGLE
```

```
                        CHECK FOR MEMORY CHECK ENABLE

 8282  055304  012704  000020          MOV     #16.,R4             ;LOAD COUNT
 8283  055310  012723  000001  16$:    MOV     #PAR.EN,(R3)+       ;ENABLE MEMORY CHECK
 8284  055314  013737  055354  000114  MOV     MEMERR,MEMVEC       ;LOAD MEMORY CHECK VECTOR
 8285  055322  012737  000340  000116  MOV     #PR7,MEMVEC+2
 8286  055330  005304                  DEC     R4                  ;CHECK IF FINISHED
 8287  055332  001366                  BNE     16$                 ;NO, SET UP NEXT MEMORY PARITY MODULE
 8288  055334  000401                  BR      22$                 ;RESTORE TRAP VECTOR
 8289
 8290  055336  022626          20$:    CMP     (SP)+,(SP)+         ;ADJUST STACK
 8291  055340  012737  000006  000004  22$:    MOV     #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
 8292  055346  005037  000006          CLR     ERRVEC+2
 8293  055352  000207                  RTS     PC                  ;RETURN
 8294
 8295                          .SBTTL  MEMORY CHECK ENABLE TRAP
 8296
 8297  055354  012737  055370  001202  MEMERR: MOV     #10$,$ESCAPE        ;LOAD ESCAPE
 8298  055362  011637  002024          MOV     (SP),TRAPPC         ;STORE PC
 8299  055366  104040                  ERROR   40                  ;REPORT MEM PARITY ERROR
 8300  055370  005037  001202  10$:    CLR     $ESCAPE             ;CLEAR ESCAPE
 8301  055374  032777  001000  123536  BIT     #SW9,@SWR           ;CHECK IF LOOP ON ERROR
 8302  055402  001001                  BNE     15$                 ;YES, FORCE STACK AND TRY AGAIN
 8303  055404  000002                  RTI                         ;NO, RETURN
 8304
 8305  055406  012706  001100  15$:    MOV     #STACK,SP           ;INITIALIZE STACK
 8306  055412  000177  123472          JMP     @$LPERR             ;LOOP ON ERROR
 8307
 8308                          .SBTTL  SCOPE HANDLER ROUTINE
 8309
 8310                          ;;**************************************************************
 8311                          ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 8312                          ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 8313                          ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 8314                          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 8315                          ;*SW14=1        LOOP ON TEST
 8316                          ;*SW11=1        INHIBIT ITERATIONS
 8317                          ;*SW09=1        LOOP ON ERROR
 8318                          ;*SW08=1        LOOP ON TEST IN SWR<7:0>
 8319                          ;*CALL
 8320                          ;*      SCOPE           ;;SCOPE=IOT
 8321
 8322  055416                  $SCOPE:
 8323  055416  104407                  CKSWR                       ;;TEST  FOR CHANGE IN SOFT-SWR
 8324  055420  032777  040000  123512  1$:     BIT     #BIT14,@SWR         ;;LOOP ON PRESENT TEST?
 8325  055426  001131                  BNE     $OVER               ;;YES IF SW14=1
 8326                          ;*****START OF CODE FOR THE XOR TESTER*****
 8327  055430  000416          $XTSTR: BR      6$                  ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 8328                                                              ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 8329  055432  013746  000004          MOV     @#ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 8330  055436  012737  055456  000004  MOV     #5$,@#ERRVEC        ;;SET FOR TIMEOUT
 8331  055444  005737  177060          TST     @#177060            ;;TIME OUT ON XOR?
 8332  055450  012637  000004          MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
 8333  055454  000500                  BR      $SVLAD              ;;GO TO THE NEXT TEST
 8334  055456  022626          5$:     CMP     (SP)+,(SP)+         ;;CLEAR THE STACK AFTER A TIME OUT
 8335  055460  012637  000004          MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
 8336  055464  000440                  BR      7$                  ;;LOOP ON THE PRESENT TEST
 8337  055466          6$:;*****END OF CODE FOR THE XOR TESTER*****
```

```
8338  055466  032777  000400  123444          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
8339  055474  001421                          BEQ     2$              ;;BR IF NO
8340  055476  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
8341  055500  117716  123434                  MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
8342  055504  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
8343  055506  022716  000076                  CMP     #76,(SP)        ;;CHECK THE NUMBER IN THE SWR
8344  055512  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
8345  055514  011637  001102                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
8346  055520  005316                          DEC     (SP)            ;;BACKUP BY ONE
8347  055522  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
8348  055524  062716  055730                  ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
8349  055530  013637  001106                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
8350  055534  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
8351  055536  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
8352  055540  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
8353  055544  001421                          BEQ     3$              ;;BR IF NO
8354  055546  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
8355  055554  101015                          BHI     3$              ;;BR IF NO
8356  055556  032777  001000  123354          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
8357  055564  001404                          BEQ     4$              ;;BR IF NO
8358  055566  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
8359  055574  000446                          BR      $OVER
8360  055576  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
8361  055602  005037  001200                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
8362  055606  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
8363  055610  032777  004000  123322  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
8364  055616  001011                          BNE     1$              ;;BR IF YES
8365  055620  005737  001222                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
8366  055624  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
8367  055626  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
8368  055632  023737  001200  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
8369  055640  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
8370  055642  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
8371  055650  013737  055726  001200          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
8372  055656  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
8373  055662  113737  001102  001220          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
8374  055670  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
8375  055674  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
8376  055700  005037  001202                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
8377  055704  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
8378  055712  013777  001102  123222  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
8379  055720  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
8380  055724  000002                          RTI                     ;;FIXES PS
8381  055726  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
8382  055730                          $SW08TBL:
8383  055730  003064                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
8384  055732  003156                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
8385  055734  003716                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
8386  055736  004456                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
8387  055740  005212                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
8388  055742  005310                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
8389  055744  006120                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
8390  055746  006704                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
8391  055750  007532                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
8392  055752  010360                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
8393  055754  011204                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
```

```
8394  055756  012030                        .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
8395  055760  012716                        .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
8396  055762  013606                        .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
8397  055764  014472                        .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
8398  055766  015356                        .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
8399  055770  016064                        .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
8400  055772  016702                        .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
8401  055774  017520                        .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
8402  055776  020334                        .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
8403  056000  021150                        .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
8404  056002  021776                        .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
8405  056004  022624                        .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
8406  056006  023450                        .WORD   TST30+2         ;;STARTING ADDRESS OF TEST 30
8407  056010  024274                        .WORD   TST31+2         ;;STARTING ADDRESS OF TEST 31
8408  056012  025122                        .WORD   TST32+2         ;;STARTING ADDRESS OF TEST 32
8409  056014  025750                        .WORD   TST33+2         ;;STARTING ADDRESS OF TEST 33
8410  056016  026574                        .WORD   TST34+2         ;;STARTING ADDRESS OF TEST 34
8411  056020  027420                        .WORD   TST35+2         ;;STARTING ADDRESS OF TEST 35
8412  056022  030270                        .WORD   TST36+2         ;;STARTING ADDRESS OF TEST 36
8413  056024  031140                        .WORD   TST37+2         ;;STARTING ADDRESS OF TEST 37
8414  056026  032004                        .WORD   TST40+2         ;;STARTING ADDRESS OF TEST 40
8415  056030  032650                        .WORD   TST41+2         ;;STARTING ADDRESS OF TEST 41
8416  056032  033414                        .WORD   TST42+2         ;;STARTING ADDRESS OF TEST 42
8417  056034  034122                        .WORD   TST43+2         ;;STARTING ADDRESS OF TEST 43
8418  056036  034630                        .WORD   TST44+2         ;;STARTING ADDRESS OF TEST 44
8419  056040  035336                        .WORD   TST45+2         ;;STARTING ADDRESS OF TEST 45
8420  056042  036044                        .WORD   TST46+2         ;;STARTING ADDRESS OF TEST 46
8421  056044  036672                        .WORD   TST47+2         ;;STARTING ADDRESS OF TEST 47
8422  056046  037520                        .WORD   TST50+2         ;;STARTING ADDRESS OF TEST 50
8423  056050  040344                        .WORD   TST51+2         ;;STARTING ADDRESS OF TEST 51
8424  056052  041170                        .WORD   TST52+2         ;;STARTING ADDRESS OF TEST 52
8425  056054  042034                        .WORD   TST53+2         ;;STARTING ADDRESS OF TEST 53
8426  056056  042700                        .WORD   TST54+2         ;;STARTING ADDRESS OF TEST 54
8427  056060  043542                        .WORD   TST55+2         ;;STARTING ADDRESS OF TEST 55
8428  056062  044404                        .WORD   TST56+2         ;;STARTING ADDRESS OF TEST 56
8429  056064  045112                        .WORD   TST57+2         ;;STARTING ADDRESS OF TEST 57
8430  056066  045620                        .WORD   TST60+2         ;;STARTING ADDRESS OF TEST 60
8431  056070  046674                        .WORD   TST61+2         ;;STARTING ADDRESS OF TEST 61
8432  056072  047160                        .WORD   TST62+2         ;;STARTING ADDRESS OF TEST 62
8433  056074  047266                        .WORD   TST63+2         ;;STARTING ADDRESS OF TEST 63
8434  056076  047410                        .WORD   TST64+2         ;;STARTING ADDRESS OF TEST 64
8435  056100  047574                        .WORD   TST65+2         ;;STARTING ADDRESS OF TEST 65
8436  056102  050476                        .WORD   TST66+2         ;;STARTING ADDRESS OF TEST 66
8437  056104  051276                        .WORD   TST67+2         ;;STARTING ADDRESS OF TEST 67
8438  056106  052076                        .WORD   TST70+2         ;;STARTING ADDRESS OF TEST 70
8439  056110  052674                        .WORD   TST71+2         ;;STARTING ADDRESS OF TEST 71
8440  056112  053472                        .WORD   TST72+2         ;;STARTING ADDRESS OF TEST 72
8441  056114  054210                        .WORD   TST73+2         ;;STARTING ADDRESS OF TEST 73
8442  056116  054406                        .WORD   TST74+2         ;;STARTING ADDRESS OF TEST 74
8443  056120  054574                        .WORD   TST75+2         ;;STARTING ADDRESS OF TEST 75
8444  056122  054710                        .WORD   TST76+2         ;;STARTING ADDRESS OF TEST 76
8445                          ;;**************************************************************
8446
8447                          .SBTTL  LOOP ON INTERNAL ERROR
8448
8449  056124  032777  001000  123006  SCOP1$: BIT     #SW9,@SWR       ;CHECK IF LOOP IN ERROR
```

```
 8450  056132  001405                        BEQ     5$                      ;NO, CONTINUE TEST
 8451  056134  105737  001103                TSTB    $ERFLG                  ;CHECK IF ERROR OCCURRED
 8452  056140  001402                        BEQ     5$                      ;NO, CONTINUE
 8453  056142  013716  001110                MOV     $LPERR,(SP)             ;LOAD ERROR RETURN
 8454  056146  000002              5$:       RTI                             ;RETURN
 8455                                .SBTTL  APT COMMUNICATIONS ROUTINE
 8456
 8457                                ;;*****************************************************************
 8458  056150  112737  000001  056414 $ATY1:  MOVB    #1,$FFLG                ;;TO REPORT FATAL ERROR
 8459  056156  112737  000001  056412 $ATY3:  MOVB    #1,$MFLG                ;;TO TYPE A MESSAGE
 8460  056164  000403                        BR      $ATYC
 8461  056166  112737  000001  056414 $ATY4:  MOVB    #1,$FFLG                ;;TO ONLY REPORT FATAL ERROR
 8462  056174                        $ATYC:
 8463  056174  010046                        MOV     R0,-(SP)                ;;PUSH R0 ON STACK
 8464  056176  010146                        MOV     R1,-(SP)                ;;PUSH R1 ON STACK
 8465  056200  105737  056412                TSTB    $MFLG                   ;;SHOULD TYPE A MESSAGE?
 8466  056204  001450                        BEQ     5$                      ;;IF NOT:  BR
 8467  056206  122737  000001  001234        CMPB    #APTENV,$ENV            ;;OPERATING UNDER APT?
 8468  056214  001031                        BNE     3$                      ;;IF NOT:  BR
 8469  056216  132737  000100  001235        BITB    #APTSPOOL,$ENVM         ;;SHOULD SPOOL MESSAGES?
 8470  056224  001425                        BEQ     3$                      ;;IF NOT:  BR
 8471  056226  017600  000004                MOV     @4(SP),R0               ;;GET MESSAGE ADDR.
 8472  056232  062766  000002  000004        ADD     #2,4(SP)                        ;;BUMP RETURN ADDR.
 8473  056240  005737  001214        1$:     TST     $MSGTYPE                  SEE IF DONE W/ LAST XMISSION?
 8474  056244  001375                        BNE     1$                      ;;IF NOT:  WAIT
 8475  056246  010037  001230                MOV     R0,$MSGAD               ;;PUT ADDR IN MAILBOX
 8476  056252  105720                2$:     TSTB    (R0)+                   ;;FIND END OF MESSAGE
 8477  056254  001376                        BNE     2$
 8478  056256  163700  001230                SUB     $MSGAD,R0               ;;SUB START OF MESSAGE
 8479  056262  006200                        ASR     R0                      ;;GET MESSAGE LNGTH IN WORDS
 8480  056264  010037  001232                MOV     R0,$MSGLGT              ;;PUT LENGTH IN MAILBOX
 8481  056270  012737  000004  001214        MOV     #4,$MSGTYPE             ;;TELL APT TO TAKE MSG.
 8482  056276  000413                        BR      5$
 8483  056300  017637  000004  056324 3$:     MOV     @4(SP),4$               ;;PUT MSG ADDR IN JSR LINKAGE
 8484  056306  062766  000002  000004        ADD     #2,4(SP)                        ;;BUMP RET RN ADDRESS
 8485  056314  013746  177776                MOV     177776,-(SP)            ;;PUSH 177776 ON STACK
 8486  056320  004737  057100                JSR     PC,$TYPE                ;;CALL TYPE MACRO
 8487  056324  000000              4$:       .WORD   0
 8488  056326                        5$:
 8489  056326  105737  056414        10$:    TSTB    $FFLG                   ;;SHOULD REPORT FATAL ERROR?
 8490  056332  001416                        BEQ     12$                     ;;IF NOT:  BR
 8491  056334  005737  001234                TST     $ENV                    ;;RUNNING UNDER APT?
 8492  056340  001413                        BEQ     12$                     ;;IF NOT:  BR
 8493  056342  005737  001214        11$:    TST     $MSGTYPE                ;;FINISHED LAST MESSAGE?
 8494  056346  001375                        BNE     11$                     ;;IF NOT:  WAIT
 8495  056350  017637  000004  001216        MOV     @4(SP),$FATAL           ;;GET ERROR #
 8496  056356  062766  000002  000004        ADD     #2,4(SP)                        ;;BUMP RETURN ADDR.
 8497  056364  005237  001214                INC     $MSGTYPE                ;;TELL APT TO TAKE ERROR
 8498  056370  105037  056414        12$:    CLRB    $FFLG                   ;;CLEAR FATAL FLAG
 8499  056374  105037  056413                CLRB    $LFLG                   ;;CLEAR LOG FLAG
 8500  056400  105037  056412                CLRB    $MFLG                   ;;CLEAR MESSAGE FLAG
 8501  056404  012601                        MOV     (SP)+,R1                ;;POP STACK INTO R1
 8502  056406  012600                        MOV     (SP)+,R0                ;;POP STACK INTO R0
 8503  056410  000207                        RTS     PC                      ;;RETURN
 8504  056412     000        $MFLG:  .BYTE   0                               ;;MESSG. FLAG
 8505  056413     000        $LFLG:  .BYTE   0                               ;;LOG FLAG
```

```
8506  056414  000                 $FFLG:  .BYTE   0               ;;FATAL FLAG
8507          056416                      .EVEN
8508          000200              APTSIZE=200
8509          000001              APTENV=001
8510          000100              APTSPOOL=100
8511          000040              APTCSUP=040
8512                              .SBTTL  ERROR HANDLER ROUTINE
8513
8514                              ;;*****************************************************************
8515                              ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
8516                              ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
8517                              ;*AND GO TO TYPERR ON ERROR
8518                              ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8519                              ;*SW15=1         HALT ON ERROR
8520                              ;*SW13=1         INHIBIT ERROR TYPEOUTS
8521                              ;*SW10=1         BELL ON ERROR
8522                              ;*SW09=1         LOOP ON ERROR
8523                              ;*CALL
8524                              ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
8525
8526  056416                      $ERROR:
8527  056416  104407                      CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
8528  056420  105237  001103      7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
8529  056424  001775              BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
8530  056426  013777  001102  122506      MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
8531  056434  032777  002000  122476      BIT     #BIT10,@SWR     ;;BELL ON ERROR?
8532  056442  001402              BEQ     1$              ;;NO - SKIP
8533  056444  104401  001204              TYPE    ,$BELL          ;;RING BELL
8534  056450  005237  001112      1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
8535  056454  011637  001116              MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
8536  056460  162737  000002  001116      SUB     #2,$ERRPC
8537  056466  117737  122424  001114      MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
8538  056474  032777  020000  122436      BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
8539  056502  001004              BNE     20$             ;;SKIP TYPEOUTS
8540  056504  004737  056616              JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
8541  056510  104401  001211              TYPE    ,$CRLF
8542  056514                      20$:
8543  056514  122737  000001  001234      CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
8544  056522  001007              BNE     2$              ;;NO,SKIP APT ERROR REPORT
8545  056524  113737  001114  056536      MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
8546  056532  004737  056166              JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
8547  056536     000      21$:    .BYTE   0
8548  056537     000              .BYTE   0
8549  056540  000777      22$:    BR      22$             ;;APT ERROR LOOP
8550  056542  005777  122372      2$:     TST     @SWR            ;;HALT ON ERROR
8551  056546  100002              BPL     3$              ;;SKIP IF CONTINUE
8552  056550  000000              HALT                    ;;HALT ON ERROR!
8553  056552  104407              CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
8554  056554  032777  001000  122356  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
8555  056562  001402              BEQ     4$              ;;BR IF NO
8556  056564  013716  001110              MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
8557  056570  005737  001202      4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
8558  056574  001402              BEQ     5$              ;;BR IF NONE
8559  056576  013716  001202              MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
8560  056602                      5$:
8561  056602  022737  055244  000042      CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
```

```
8562  056610  001001                        BNE     6$              ;;BRANCH IF NO
8563  056612  000000                        HALT                    ;;YES
8564  056614                          6$:
8565  056614  000002                        RTI                     ;;RETURN
8566
8567                                  ;;*****************************************************************
8568                                  .SBTTL  TYPE ERROR ROUTINE
8569                                  ;*ENTRY JSR PC,TYPERR
8570                                  ;*RETURN RTS PC
8571                                  ;*
8572                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
8573                                  ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
8574                                  ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
8575                                  ;*THE ERROR.
8576                                  ;;*****************************************************************
8577  056616  104413                  TYPERR: SAVREG
8578  056620  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
8579  056624  042700  177400                  BIC     #177400,R0      ;CLEAR UNUSED BITS
8580  056630  005300                        DEC     R0              ;FORM INDEX FOR ERROR TABLE
8581  056632  006300                        ASL     R0
8582  056634  006300                        ASL     R0
8583  056636  006300                        ASL     R0
8584  056640  062700  001300          1$:   ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
8585  056644  012037  056660                MOV     (R0)+,2$        ;GET EM POINTER
8586  056650  001404                        BEQ     3$              ;BRANCH IF THERE ISN'T ONE
8587  056652  104401  001211                TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
8588  056656  104401                        TYPE                    ;TYPE ERROR MESSAGE (EM)
8589  056660  000000                2$:     .WORD   0               ;EM POINTER GOES HERE
8590  056662  012037  056676          3$:   MOV     (R0)+,4$        ;GET DH POINTER
8591  056666  001404                        BEQ     5$              ;BRANCH IF THERE ISN'T ONE
8592  056670  104401  001211                TYPE    ,$CRLF          ;TYPE CR-LF
8593  056674  104401                        TYPE                    ;TYPE DATA HEADER
8594  056676  000000                4$:     .WORD   0               ;DH POINTER GOES HERE
8595  056700  012001                5$:     MOV     (R0)+,R1        ;GET DT POINTER
8596  056702  001445                        BEQ     20$             ;BRANCH IF THERE ARE NONE
8597  056704  005004                        CLR     R4              ;RESET INDENT SWITCH
8598  056706  012000                        MOV     (R0)+,R0        ;GET DF POINTER
8599  056710  012002                        MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
8600  056712  104401  001211                TYPE    ,$CRLF          ;TYPE <CR><LF>'
8601  056716  112003                10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
8602  056720  105720                        TSTB    (R0)+           ;BUMP PAST FORMAT WORD
8603  056722  005703                        TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
8604  056724  001416                        BEQ     14$             ;NO - SKIP DATA PRINT
8605  056726  005704                        TST     R4              ;CHECK IF INDENT WORDS
8606  056730  001004                        BNE     12$             ;YES, GO INDENT
8607  056732  013146                11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
8608  056734  104402                        TYPOC                   ;TYPE IT
8609  056736  005303                        DEC     R3              ;MORE DATA WORDS
8610  056740  001403                        BEQ     13$             ;NO-BRANCH
8611  056742  104401  062161          12$:  TYPE    ,SPACE2         ;TYPE SEPARATORS
8612  056746  000771                        BR      11$             ;LOOP
8613  056750  104401  001211          13$:  TYPE    ,$CRLF          ;TYPE <CR><LF>
8614  056754  005710                        TST     (R0)            ;CHECK IF NEXT HEADER AVAILIBLE
8615  056756  001401                        BEQ     14$             ;NO, DO NOT CHANGE INDENT
8616  056760  005104                        COM     R4              ;CHANGE INDENT
8617  056762  005302                14$:    DEC     R2              ;MORE DH'S?
```

```
 8618  056764  003414                          BLE     20$             ;NO-BRANCH
 8619  056766  012037  057006          15$:    MOV     (R0)+,18$       ;GET NEXT DH POINTER
 8620  056772  001751                          BEQ     10$             ;IF NO HEADER GET DATA
 8621  056774  005704                          TST     R4              ;INDENT?
 8622  056776  001402                          BEQ     17$             ;NO-BRANCH
 8623  057000  104401  062161                  TYPE    ,SPACE2         ;INDENT
 8624  057004  104401                  17$:    TYPE                    ;TYPE DH
 8625  057006  000000                  18$:    .WORD   0               ;DH POINTER GOES HERE
 8626  057010  104401  001211                  TYPE    ,$CRLF
 8627  057014  000740                          BR      10$             ;LOOP
 8628  057016  104414                  20$:    RESREG
 8629  057020  005237  002006                  INC     ERRCNT          ;INCREMENT ERROR COUNT
 8630  057024  032777  010000  122106          BIT     #SW12,@SWR      ;CHECK IF ABORT AFTER 20 ERRORS
 8631  057032  001421                          BEQ     25$             ;NO, TYPE OUT ERROR
 8632  057034  022737  000024  002006          CMP     #20.,ERRCNT     ;CHECK IF ERROR THRESHOLD EXCEEDED
 8633  057042  103015                          BHIS    25$             ;NO, PRINT ERROR
 8634  057044  104401  062164                  TYPE    ,ABORT          ;TYPE "PROGRAM HAS BEEN ABORTED BECAUSE
 8635                                                                  ;  ERROR THRESHOLD EXCEEDED"
 8636  057050  005737  000042                  TST     42              ;CHECK IF IN CHAIN MODE
 8637  057054  001407                          BEQ     30$             ;NO, HALT
 8638  057056  012737  000001  055102          MOV     #1,$EOPCT       ;FORCE END OF PASS COUNT TO ONE FOR ABORT
 8639  057064  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
 8640  057070  000137  055054                  JMP     $EOP            ;BRING IN NEXT PROGRAM IN CHAIN
 8641  057074  000000                  30$:    HALT
 8642  057076  000207                  25$:    RTS     PC
 8643                                  .SBTTL  TYPE ROUTINE
 8644
 8645                                  ;;**************************************************************
 8646                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 8647                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 8648                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 8649                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 8650                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 8651                                  ;*
 8652                                  ;*CALL:
 8653                                  ;*1) USING A TRAP INSTRUCTION
 8654                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 8655                                  ;*OR
 8656                                  ;*      TYPE
 8657                                  ;*      MESADR
 8658                                  ;*
 8659
 8660  057100  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 8661  057104  100002                  BPL     1$              ;;BR IF YES
 8662  057106  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
 8663  057110  000430                  BR      3$              ;;LEAVE
 8664  057112  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
 8665  057114  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 8666  057120  122737  000001  001234  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 8667  057126  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 8668  057130  132737  000100  001235  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 8669  057136  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 8670  057140  010037  057150          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 8671  057144  004737  056156          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 8672  057150  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
 8673  057152  132737  000040  001235  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
```

```
8674  057160  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
8675  057162  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
8676  057164  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
8677  057166  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
8678  057170  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
8679  057172  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
8680  057176  000002                        RTI                     ;;RETURN
8681  057200  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
8682  057204  001430                        BEQ     8$
8683  057206  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
8684  057212  001006                        BNE     5$
8685  057214  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
8686  057216  104401                        TYPE                    ;;TYPE A CR AND LF
8687  057220  001211                        $CRLF
8688  057222  105037  057356                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
8689  057226  000755                        BR      2$              ;;GET NEXT CHARACTER
8690  057230  004737  057312        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
8691  057234  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
8692  057240  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
8693  057242  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
8694                                                                ;;AND THE NULL CHAR.
8695  057246  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
8696  057252  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
8697  057254  004737  057312                JSR     PC,$TYPEC       ;;GO TYPE A NULL
8698  057260  105337  057356                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
8699  057264  000770                        BR      7$              ;;LOOP
8700
8701                                ;HORIZONTAL TAB PROCESSOR
8702
8703  057266  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
8704  057272  004737  057312        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
8705  057276  132737  000007  057356        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
8706  057304  001372                        BNE     9$              ;;TAB STOP
8707  057306  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
8708  057310  000724                        BR      2$              ;;GET NEXT CHARACTER
8709  057312  105777  121632        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
8710  057316  100375                        BPL     $TYPEC
8711  057320  116677  000002  121624        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
8712  057326  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
8713  057334  001003                        BNE     1$              ;;BRANCH IF NO
8714  057336  105037  057356                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
8715  057342  000406                        BR      $TYPEX          ;;EXIT
8716  057344  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
8717  057352  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
8718  057354  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
8719  057356  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
8720  057360  000207        $TYPEX: RTS     PC
8721
8722                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
8723
8724                                ;;**************************************************************
8725                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
8726                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
8727                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
8728                                ;*CALL:
8729                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
```

```
8730                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
8731                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
8732                                    ;*      .BYTE   M               ;;M=1 OR 0
8733                                    ;*                                      ;;1=TYPE LEADING ZEROS
8734                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
8735                                    ;*
8736                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
8737                                    ;*$TYPOS OR $TYPOC
8738                                    ;*CALL:
8739                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
8740                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
8741                                    ;*
8742                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
8743                                    ;*CALL:
8744                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
8745                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
8746
8747  057362  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
8748  057366  116637  000001  057605           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
8749  057374  112637  057607                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
8750  057400  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
8751  057404  000406                           BR      $TYPON
8752  057406  112737  000001  057605   $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
8753  057414  112737  000006  057607           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
8754  057422  112737  000005  057604   $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
8755  057430  010346                           MOV     R3,-(SP)        ;;SAVE R3
8756  057432  010446                           MOV     R4,-(SP)        ;;SAVE R4
8757  057434  010546                           MOV     R5,-(SP)        ;;SAVE R5
8758  057436  113704  057607                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
8759  057442  005404                           NEG     R4
8760  057444  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
8761  057450  110437  057606                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
8762  057454  113704  057605                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
8763  057460  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
8764  057464  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
8765  057466  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
8766  057470  000404                           BR      3$              ;;GO DO MSB
8767  057472  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
8768  057474  006105                           ROL     R5
8769  057476  006105                           ROL     R5
8770  057500  010503                           MOV     R5,R3
8771  057502  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
8772  057504  105337  057606                   DECB    $OMODE          ;;TYPE THIS DIGIT?
8773  057510  100016                           BPL     7$              ;;BR IF NO
8774  057512  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
8775  057516  001002                           BNE     4$              ;;TEST FOR 0
8776  057520  005704                           TST     R4              ;;SUPPRESS THIS 0?
8777  057522  001403                           BEQ     5$              ;;BR IF YES
8778  057524  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
8779  057526  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
8780  057532  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
8781  057536  110337  057602                   MOVB    R3,8$           ;;SAVE FOR TYPING
8782  057542  104401  057602                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
8783  057546  105337  057604   7$:     DECB    $OCNT           ;;COUNT BY 1
8784  057552  003347                           BGT     2$              ;;BR IF MORE TO DO
8785  057554  002402                           BLT     6$              ;;BR IF DONE
```

```
                        BINARY TO OCTAL (ASCII) AND TYPE

 8786  057556  005204                         INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
 8787  057560  000744                         BR      2$                      ;;GO DO THE LAST DIGIT
 8788  057562  012605                 6$:     MOV     (SP)+,R5                ;;RESTORE R5
 8789  057564  012604                         MOV     (SP)+,R4                ;;RESTORE R4
 8790  057566  012603                         MOV     (SP)+,R3                ;;RESTORE R3
 8791  057570  016666  000002  000004         MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
 8792  057576  012616                         MOV      SP,+,(SP)
 8793  057600  000002                         RTI                             ;;RETURN
 8794  057602     000                 8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
 8795  057603     000                         .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
 8796  057604     000                 $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
 8797  057605     000                 $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
 8798  057606  000000                 $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
 8799                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 8800
 8801                                  ;;*****************************************************************
 8802                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 8803                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 8804                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 8805                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 8806                                  ;*REPLACED WITH SPACES.
 8807                                  ;*CALL:
 8808                                  ;*      MOV     NUM,-(SP)               ;;PUT THE BINARY NUMBER ON THE STACK
 8809                                  ;*      TYPDS                           ;;GO TO THE ROUTINE
 8810
 8811  057610                         $TYPDS:
 8812  057610  010046                         MOV     R0,-(SP)                ;;PUSH R0 ON STACK
 8813  057612  010146                         MOV     R1,-(SP)                ;;PUSH R1 ON STACK
 8814  057614  010246                         MOV     R2,-(SP)                ;;PUSH R2 ON STACK
 8815  057616  010346                         MOV     R3,-(SP)                ;;PUSH R3 ON STACK
 8816  057620  010546                         MOV     R5,-(SP)                ;;PUSH R5 ON STACK
 8817  057622  012746  020200                 MOV     #20200,-(SP)            ;;SET BLANK SWITCH AND SIGN
 8818  057626  016605  000020                 MOV     20(SP),R5               ;;GET THE INPUT NUMBER
 8819  057632  100004                         BPL     1$                      ;;BR IF INPUT IS POS.
 8820  057634  005405                         NEG     R5                      ;;MAKE THE BINARY NUMBER POS.
 8821  057636  112766  000055  000001         MOVB    #'-,1(SP)               ;;MAKE THE ASCII NUMBER NEG.
 8822  057644  005000                 1$:     CLR     R0                      ;;ZERO THE CONSTANTS INDEX
 8823  057646  012703  060024                 MOV     #$DBLK,R3               ;;SETUP THE OUTPUT POINTER
 8824  057652  112723  000040                 MOVB    #' ,(R3)+               ;;SET THE FIRST CHARACTER TO A BLANK
 8825  057656  005002                 2$:     CLR     R2                      ;;CLEAR THE BCD NUMBER
 8826  057660  016001  060014                 MOV     $DTBL(R0),R1            ;;GET THE CONSTANT
 8827  057664  160105                 3$:     SUB     R1,R5                   ;;FORM THIS BCD DIGIT
 8828  057666  002402                         BLT     4$                      ;;BR IF DONE
 8829  057670  005202                         INC     R2                      ;;INCREASE THE BCD DIGIT BY 1
 8830  057672  000774                         BR      3$
 8831  057674  060105                 4$:     ADD     R1,R5                   ;;ADD BACK THE CONSTANT
 8832  057676  005702                         TST     R2                      ;;CHECK IF BCD DIGIT=0
 8833  057700  001002                         BNE     5$                      ;;FALL THROUGH IF 0
 8834  057702  105716                         TSTB    (SP)                    ;;STILL DOING LEADING 0'S?
 8835  057704  100407                         BMI     7$                      ;;BR IF YES
 8836  057706  106316                 5$:     ASLB    (SP)                    ;;MSD?
 8837  057710  103003                         BCC     6$                      ;;BR IF NO
 8838  057712  116663  000001  177777         MOVB    1(SP),-1(R3)            ;;YES--SET THE SIGN
 8839  057720  052702  000060                 BIS     #'0,R2                  ;;MAKE THE BCD DIGIT ASCII
 8840  057724  052702  000040                 BIS     #' ,R2                  ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 8841  057730  110223                         MOVB    R2,(R3)+                ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
```

```
8842  057732  005720                          TST     (R0)+           ;;JUST INCREMENTING
8843  057734  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
8844  057740  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
8845  057742  003002                          BGT     8$              ;;GO TO EXIT
8846  057744  010502                          MOV     R5,R2           ;;GET THE LSD
8847  057746  000764                          BR      6$              ;;GO CHANGE TO ASCII
8848  057750  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
8849  057752  100003                          BPL     9$              ;;BR IF NO
8850  057754  116563  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
8851  057762  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
8852  057764  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
8853  057766  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
8854  057770  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
8855  057772  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
8856  057774  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
8857  057776  104401  060024                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
8858  060002  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
8859  060010  012616                          MOV     (SP)+,(SP)
8860  060012  000002                          RTI                     ;;RETURN TO USER
8861  060014  023420                  $DTBL:  10000.
8862  060016  001750                          1000.
8863  060020  000144                          100.
8864  060022  000012                          10.
8865  060024  000004                  $DBLK:  .BLKW   4
8866                                   .SBTTL  TTY INPUT ROUTINE
8867
8868                                   ;;******************************************************************
8869                                   .ENABL  LSB
8870
8871                                   ;;******************************************************************
8872                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
8873                                   ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
8874                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
8875                                   ;*WHEN OPERATING IN TTY FLAG MODE.
8876  060034  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
8877  060042  001074                          BNE     15$             ;;BRANCH IF NO
8878  060044  105777  121074                  TSTB    @$TKS           ;;CHAR THERE?
8879  060050  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
8880  060052  117746  121070                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
8881  060056  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
8882  060062  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
8883  060066  001062                          BNE     15$             ;;NO, RETURN TO USER
8884  060070  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE
8885  060076  001456                          BEQ     15$             ;;BRANCH IF YES
8886
8887  060100  104401  060707                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
8888  060104  104401  060714          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
8889  060110  013746  000176                  MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
8890  060114  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8891  060116  104401  060725                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
8892  060122  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
8893  060124  005046                          CLR     -(SP)           ;;THE NEW SWR
8894  060126  105777  121012          7$:     TSTB    @$TKS           ;;CHAR THERE?
8895  060132  100375                          BPL     7$              ;;IF NOT TRY AGAIN
8896
8897  060134  117746  121006                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
```

```
8898  060140  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
8899
8900
8901
8902  060144  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
8903  060150  001005                          BNE     10$             ;;BRANCH IF NOT
8904  060152  104401  060702                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
8905  060156  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
8906  060162  000757                          BR      19$             ;;LET'S TRY IT AGAIN
8907
8908
8909  060164  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
8910  060170  001022                          BNE     16$             ;;BRANCH IF NO
8911  060172  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
8912  060176  001403                          BEQ     11$             ;;BRANCH IF YES
8913  060200  016677  000002  120732          MOV     2(SP),@SWR      ;;SAVE NEW SWR
8914  060206  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
8915  060212  104401  001211          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
8916  060216  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
8917  060224  001003                          BNE     15$             ;;BRANCH IF NOT
8918  060226  012777  000100  120710          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
8919  060234  000002          15$:    RTI                             ;;RETURN
8920  060236  004737  057312          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
8921  060242  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
8922  060246  002420                          BLT     18$             ;;BRANCH IF YES
8923  060250  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
8924  060254  003015                          BGT     18$             ;;BRANCH IF YES
8925  060256  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
8926  060262  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
8927  060266  001403                          BEQ     17$             ;;BRANCH IF YES
8928  060270  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
8929  060272  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
8930  060274  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
8931  060276  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
8932  060302  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
8933  060306  000707                          BR      7$              ;;GET THE NEXT ONE
8934  060310  104401  001210          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
8935  060314  000720                          BR      20$             ;;SIMULATE CONTROL-U
8936                                  .DSABL  LSB
8937
8938
8939                                  ;;*****************************************************************
8940                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
8941                                  ;*CALL:
8942                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TT
8943                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
8944                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
8945                                  ;
8946
8947  060316  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
8948  060320  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
8949  060326  105777  120612          1$:     TSTB    @$TKS           ;;WAIT FOR
8950  060332  100375                          BPL     1$              ;;A CHARACTER
8951  060334  117766  120606  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
8952  060342  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
8953  060350  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
```

```
8954  060356  001013                        BNE     3$              ;;BRANCH IF NO
8955  060360  105777  120560          2$:   TSTB    @$TKS           ;;WAIT FOR A CHARACTER
8956  060364  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
8957  060366  117746  120554                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
8958  060372  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
8959  060376  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
8960  060402  001366                        BNE     2$              ;;IF NOT DISCARD IT
8961  060404  000750                        BR      1$              ;;YES, RESUME
8962  060406  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
8963  060414  002407                        BLT     4$              ;;BRANCH IF YES
8964  060416  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
8965  060424  003003                        BGT     4$              ;;BRANCH IF YES
8966  060426  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
8967  060434  000002                  4$:   RTI                     ;;GO BACK TO USER
8968                                        ;;**************************************************
8969                                        ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
8970                                        ;*CALL:
8971                                        ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
8972                                        ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
8973                                        ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
8974
8975  060436  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
8976  060440  005046                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
8977  060442  012703  060672          1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
8978  060446  022703  060702          2$:   CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
8979  060452  101456                        BLOS    4$              ;;BR IF YES
8980  060454  104410                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
8981  060456  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
8982  060460  122713  000177          10$:  CMPB    #177,(R3)       ;;IS IT A RUBOUT
8983  060464  001022                        BNE     5$              ;;BR IF NO
8984  060466  005716                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
8985  060470  001007                        BNE     6$              ;;BR IF NO
8986  060472  112737  000134  060670        MOVB    #'\,9$          ;;TYPE A BACK SLASH
8987  060500  104401  060670                TYPE    ,9$
8988  060504  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
8989  060510  005303                  6$:   DEC     R3              ;;BACKUP BY ONE
8990  060512  020327  060672                CMP     R3,#$TTYIN      ;;STACK EMPTY?
8991  060516  103434                        BLO     4$              ;;BR IF YES
8992  060520  111337  060670                MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
8993  060524  104401  060670                TYPE    ,9$             ;;GO TYPE
8994  060530  000746                        BR      2$              ;;GO READ ANOTHER CHAR.
8995  060532  005716                  5$:   TST     (SP)            ;;RUBOUT KEY SET?
8996  060534  001406                        BEQ     7$              ;;BR IF NO
8997  060536  112737  000134  060670        MOVB    #'\,9$          ;;TYPE A BACK SLASH
8998  060544  104401  060670                TYPE    ,9$
8999  060550  005016                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
9000  060552  122713  000025          7$:   CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
9001  060556  001003                        BNE     8$              ;;BR IF NO
9002  060560  104401  060702                TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
9003  060564  000726                        BR      1$              ;;GO START OVER
9004  060566  122713  000022          8$:   CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
9005  060572  001011                        BNE     3$              ;;BRANCH IF NO
9006  060574  105013                        CLRB    (R3)            ;;CLEAR THE CHARACTER
9007  060576  104401  001211                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
9008  060602  104401  060672                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
9009  060606  000717                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
```

```
9010  060610  104401  001210          4$:     TYPE    $QUES           ;;TYPE A '?'
9011  060614  000712                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
9012  060616  111337  060670          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
9013  060622  104401  060670                  TYPE    9$
9014  060626  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
9015  060632  001305                  BNE     2$              ;;LOOP IF NOT RETURN
9016  060634  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
9017  060640  104401  001212                  TYPE    $LF             ;;TYPE A LINE FEED
9018  060644  005726                  TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
9019  060646  012603                  MOV     (SP)+,R3        ;;RESTORE R3
9020  060650  011646                  MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
9021  060652  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
9022  060660  012766  060672  000004          MOV     #$TTYIN,4(SP)
9023  060666  000002                  RTI                     ;;RETURN
9024  060670     000          9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
9025  060671     000                  .BYTE   0               ;;TERMINATOR
9026  060672  000010          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
9027  060702  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
9028  060707     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
9029  060714  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
9030  060722  020075     000
9031  060725     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
9032  060732  036440  000040
9033                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
9034
9035                                  ;;**************************************************************
9036                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
9037                                  ;*CHANGE IT TO BINARY.
9038                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
9039                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
9040                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
9041                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
9042                                  ;*CALL:
9043                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
9044                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
9045                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
9046
9047  060736  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
9048  060740  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
9049  060746  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9050  060750  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9051  060752  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9052  060754  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
9053  060756  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
9054  060760  010037  061064          MOV     R0,5$           ;;AND SAVE IT
9055  060764  005001                  CLR     R1              ;;CLEAR DATA WORD
9056  060766  005002                  CLR     R2
9057  060770  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
9058  060772  001420                  BEQ     3$              ;;IF ZERO GET OUT
9059  060774  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
9060  061000  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
9061  061002  122716  000067                  CMPB    #'7,(SP)
9062  061006  002423                  BLT     4$
9063  061010  006301                  ASL     R1              ;;*2
9064  061012  006102                  ROL     R2
9065  061014  006301                  ASL     R1              ;;*4
```

```
 9066  061016  006102                        ROL     R2
 9067  061020  006301                        ASL     R1                  ;;*8
 9068  061022  006102                        ROL     R2
 9069  061024  042716  177770                BIC     #^C7,(SP)           ;;STRIP THE ASCII JUNK
 9070  061030  062601                        ADD     (SP)+,R1            ;;ADD IN THIS DIGIT
 9071  061032  000756                        BR      2$                  ;;LOOP
 9072  061034  005726                3$:     TST     (SP)+               ;;CLEAN TERMINATOR FROM STACK
 9073  061036  010166  000012                MOV     R1,12(SP)           ;;SAVE THE RESULT
 9074  061042  010237  061074                MOV     R2,$HIOCT
 9075  061046  012602                        MOV     (SP)+,R2            ;;POP STACK INTO R2
   76  061050  012601                        MOV     (SP)+,R1            ;;POP STACK INTO R1
 9077  061052  012600                        MOV     (SP)+,R0            ;;POP STACK INTO R0
 9078  061054  000002                        RTI                         ;;RETURN
 9079  061056  005726                4$:     TST     (SP)+               ;;CLEAN PARTIAL FROM STACK
 9080  061060  105010                        CLRB    (R0)                ;;SET A TERMINATOR
 9081  061062  104401                        TYPE                        ;;TYPE UP THRU THE BAD CHAR.
 9082  061064  000000                5$:     .WORD   0
 9083  061066  104401  001210                TYPE    ,$QUES              ;;"?" "CR" & "LF"
 9084  061072  000730                        BR      1$                  ;;TRY AGAIN
 9085  061074  000000                $HIOCT: .WORD   0                   ;;HIGH ORDER BITS GO HERE
 9086                                        .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
 9087
 9088                                        ;;******************************************************************
 9089                                        ;*SAVE R0-R5
 9090                                        ;*CALL:
 9091                                        ;*      SAVREG
 9092                                        ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
 9093                                        ;*
 9094                                        ;*TOP---(+16)
 9095                                        ;* +2---(+18)
 9096                                        ;* +4---R5
 9097                                        ;* +6---R4
 9098                                        ;* +8---R3
 9099                                        ;*+10---R2
 9100                                        ;*+12---R1
 9101                                        ;*+14---R0
 9102
 9103  061076                                $SAVREG:
 9104  061076  010046                        MOV     R0,-(SP)            ;;PUSH R0 ON STACK
 9105  061100  010146                        MOV     R1,-(SP)            ;;PUSH R1 ON STACK
 9106  061102  010246                        MOV     R2,-(SP)            ;;PUSH R2 ON STACK
 9107  061104  010346                        MOV     R3,-(SP)            ;;PUSH R3 ON STACK
 9108  061106  010446                        MOV     R4,-(SP)            ;;PUSH R4 ON STACK
 9109  061110  010546                        MOV     R5,-(SP)            ;;PUSH R5 ON STACK
 9110  061112  016646  000022                MOV     22(SP),-(SP)        ;;SAVE PS OF MAIN FLOW
 9111  061116  016646  000022                MOV     22(SP),-(SP)        ;;SAVE PC OF MAIN FLOW
 9112  061122  016646  000022                MOV     22(SP),-(SP)        ;;SAVE PS OF CALL
 9113  061126  016646  000022                MOV     22(SP),-(SP)        ;;SAVE PC OF CALL
 9114  061132  000002                        RTI
 9115
 9116                                        ;*RESTORE R0-R5
 9117                                        ;*CALL:
 9118                                        ;*      RESREG
 9119  061134                                $RESREG:
 9120  061134  012666  000022                MOV     (SP)+,22(SP)        ;;RESTORE PC OF CALL
 9121  061140  012666  000022                MOV     (SP)+,22(SP)        ;;RESTORE PS OF CALL
```

```
 9122  061144  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 9123  061150  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 9124  061154  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
 9125  061156  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
 9126  061160  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
 9127  061162  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 9128  061164  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 9129  061166  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
 9130  061170  000002                  RTI
 9131
 9132                                  .SBTTL  POWER DOWN AND UP ROUTINES
 9133
 9134                                  ;;*****************************************************************
 9135                                  ;POWER DOWN ROUTINE
 9136  061172  017737  117742  002026  $PWRDN: MOV     @SWR,SAVSWR     ;SAVE SWITCH REGISTER
 9137  061200  012737  061220  000024          MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
 9138  061206  012737  000340  000026          MOV     #PR7,PWRVEC+2
 9139  061214  000000                  HALT
 9140  061216  000776                  BR      .-2     ;HANG UP
 9141
 9142                                  ;;*****************************************************************
 9143                                  ;POWER UP ROUTINE
 9144  061220  005037  061310  $PWRUP: CLR     $PWRCT          ;LOAD WAIT COUNT
 9145  061224  012737  000144  061312          MOV     #100.,$PWRCT+2
 9146  061232  005237  061310  1$:     INC     $PWRCT          ;WAIT FOR TELETYPE
 9147  061236  001375                  BNE     1$
 9148  061240  005337  061312          DEC     $PWRCT+2
 9149  061244  001372                  BNE     1$
 9150  061246  012737  061172  000024          MOV     #$PWRDN,PWRVEC  ;SET UP FOR POWER DOWN VECTOR
 9151  061254  012737  000340  000026          MOV     #PR7,PWRVEC+2
 9152  061262  012706  001100          MOV     #STACK,SP       ;FORCE STACK
 9153  061266  104401  061314          TYPE    $POWER          ;TYPE POWER
 9154  061272  004737  055264          JSR     PC,CHKPAR       ;REINITIALIZE MEMORY CHECK ENABLE
 9155  061276  013777  002026  117634  MOV     SAVSWR,@SWR     ;RESTORE SWITCH REGISTER
 9156  061304  000177  117576          JMP     @$LPADR         ;GO BACK TO LAST TEST
 9157
 9158  061310  000000  000000  $PWRCT: .WORD   0,0              ;TELETYPE TIME OUT
 9159  061314  047520  042527  000122  $POWER: .ASCIZ  /POWER/
 9160                                          .EVEN
 9161                                  .SBTTL  TRAP DECODER
 9162
 9163                                  ;;*****************************************************************
 9164                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 9165                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 9166                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 9167                                  ;*GO TO THAT ROUTINE.
 9168
 9169  061322  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 9170  061324  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
 9171  061330  005740                  TST     -(R0)           ;;BACKUP BY 2
 9172  061332  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 9173  061334  006300                  ASL     R0              ;;POSITION FOR INDEXING
 9174  061336  016000  061356          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 9175  061342  000200                  RTS     R0              ;;GO TO ROUTINE
 9176
 9177
```

```
 9178                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 9179
 9180  061344  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 9181  061346  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 9182  061354  000002                           RTI                     ;;RESTORE THE PSW
 9183
 9184                                   .SBTTL  TRAP TABLE
 9185
 9186                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 9187                                   ;*BY THE "TRAP" INSTRUCTION.
 9188
 9189                                   ;       ROUTINE
 9190                                   ;       -------
 9191  061356  061344                   $TRPAD: .WORD   $TRAP2
 9192  061360  057100                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 9193  061362  057406                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 9194  061364  057362                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 9195  061366  057422                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 9196  061370  057610                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 9197
 9198  061372  060104                           $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 9199
 9200  061374  060034                           $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 9201  061376  060316                           $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 9202  061400  060436                           $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 9203  061402  060736                           $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 9204  061404  061076                           $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
 9205  061406  061134                           $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
 9206  061410  056124                           SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
```

```
 9207                                           .SBTTL  DATA TABLE FOR ERROR PRINT OUT
 9208
 9209  061412  001220  002024                  DT000:  .WORD   $TESTN,TRAPPC
 9210  061416  001220  001116  001122          DT001:  .WORD   $TESTN,$ERRPC,$BDADR
 9211  061424  001220  001116  001124          DT002:  .WORD   $TESTN,$ERRPC,$GDDAT,$BDDAT
 9212  061432  001126
 9213  061434  001220  001116  002010          DT003:  .WORD   $TESTN,$ERRPC,CONFIG,$GDDAT,$BDDAT
 9214  061442  001124  001126
 9215  061446  001220  001116  002014          DT004:  .WORD   $TESTN,$ERRPC,PREREG,$GDDAT,$BDDAT
 9216  061454  001124  001126
 9217  061460  001220  001116                  DT005:  .WORD   $TESTN,$ERRPC
 9219  061464  001220  001116  002000          DT006:  .WORD   $TESTN,$ERRPC,RKVEC,$BDADR
 9219  061472  001122
 9220  061474  001220  001116  002020          DT011:  .WORD   $TESTN,$ERRPC,PRIOR
 9221  061502  001220  001116  001740          DT016:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2
 9222  061510  001700  001750  001710
 9223  061516  001220  001116  001740          DT024:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,SILCNT
 9224  061524  001700  001750  001710
 9225  061532  002022
 9226  061534  001220  001116  001740          DT026:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,SILCNT,$GDDAT,$BDDAT
 9227  061542  001700  001750  001710
 9228  061550  002022  001124  001126
```

```
                                        .SBTTL  DATA FORMAT FOR ERROR PRINT OUT

061556  000001          DF000:  .WORD   1
061560      002  000            .BYTE   2,0
061562  000004          DF001:  .WORD   4                       ;ERROR 1
061564      000  000            .BYTE   0,0
061566  062251                  .WORD   DH000A
061570      000  000            .BYTE   0,0
061572  062267                  .WORD   DH000B
061574      002  000            .BYTE   2,0
061576  062332                  .WORD   DH001
061600      001  000            .BYTE   1,0
061602  000005          DF002:  .WORD   5                       ;ERROR 2
061604      000  000            .BYTE   0,0
061606  062251                  .WORD   DH000A
061610      000  000            .BYTE   0,0
061612  062267                  .WORD   DH000B
061614      002  000            .BYTE   2,0
061616  062351                  .WORD   DH002A
061620      000  000            .BYTE   0,0
061622  062370                  .WORD   DH002B
061624      002  000            .BYTE   2,0
061626  000005          DF003:  .WORD   5                       ;ERROR 3
061630      000  000            .BYTE   0,0
061632  062251                  .WORD   DH000A
061634      000  000            .BYTE   0,0
061636  062267                  .WORD   DH000B
061640      002  000            .BYTE   2,0
061642  062406                  .WORD   DH003A
061644      000  000            .BYTE   0,0
061646  062435                  .WORD   DH003B
061650      003  000            .BYTE   3,0
061652  000005          DF004:  .WORD   5                       ;ERROR 4
061654      000  000            .BYTE   0,0
061656  062251                  .WORD   DH000A
061660      000  000            .BYTE   0,0
061662  062267                  .WORD   DH000B
061664      002  000            .BYTE   2,0
061666  062463                  .WORD   DH004A
061670      000  000            .BYTE   0,0
061672  062512                  .WORD   DH004B
061674      003  000            .BYTE   3,0
061676  000003          DF005:  .WORD   3                       ;ERROR 5
061700      000  000            .BYTE   0,0
061702  062251                  .WORD   DH000A
061704      000  000            .BYTE   0,0
061706  062267                  .WORD   DH000B
061710      002  000            .BYTE   2,0
061712  000005          DF006:  .WORD   5                       ;ERROR 6
061714      000  000            .BYTE   0,0
061716  062251                  .WORD   DH000A
061720      000  000            .BYTE   0,0
061722  062267                  .WORD   DH000B
061724      002  000            .BYTE   2,0
061726  062540                  .WORD   DH006A
061730      000  000            .BYTE   0,0
```

```
9285  061732  062557                        .WORD   DH0068
9286  061734      002     000               .BYTE   2,0
9287  061736  000004              DF011:    .WORD   4               ;ERROR 11
9288  061740      000     000               .BYTE   0,0
9289  061742  062251                        .WORD   DH000A
9290  061744      000     000               .BYTE   0,0
9291  061746  062267                        .WORD   DH000B
9292  061750      002     000               .BYTE   2,0
9293  061752  062576                        .WORD   DH011
9294  061754      001     000               .BYTE   1,0
9295  061756  000005              DF016:    .WORD   5               ;ERROR 16
9296  061760      000     000               .BYTE   0,0
9297  061762  062251                        .WORD   DH000A
9298  061764      000     000               .BYTE   0,0
9299  061766  062267                        .WORD   DH000B
9300  061770      002     000               .BYTE   2,0
9301  061772  062540                        .WORD   DH006A
9302  061774      000     000               .BYTE   0,0
9303  061776  062660                        .WORD   DH016B
9304  062000      004     000               .BYTE   4,0
9305  062002  000005              DF024:    .WORD   5               ;ERROR 24
9306  062004      000     000               .BYTE   0,0
9307  062006  062251                        .WORD   DH000A
9308  062010      000     000               .BYTE   0,0
9309  062012  062267                        .WORD   DH000B
9310  062014      002     000               .BYTE   2,0
9311  062016  062716                        .WORD   DH024A
9312  062020      000     000               .BYTE   0,0
9313  062022  062763                        .WORD   DH024B
9314  062024      005     000               .BYTE   5,0
9315  062026  000005              DF026:    .WORD   5               ;ERROR 26
9316  062030      000     000               .BYTE   0,0
9317  062032  062251                        .WORD   DH000A
9318  062034      000     000               .BYTE   0,0
9319  062036  062267                        .WORD   DH000B
9320  062040      002     000               .BYTE   2,0
9321  062042  063030                        .WORD   DH026A
9322  062044      000     000               .BYTE   0,0
9323  062046  063117                        .WORD   DH026B
9324  062050      007     000               .BYTE   7,0
```

```
 9325                                           .SBTTL  ASCII MESSAGES
 9326
 9327  062052  005015  045522  030466  OPR001: .ASCIZ  <15><12>/RK611 BUS ADDRESS ( /
 9328  062060  020061  052502  020123
 9329  062066  040101  051104  051505
 9330  062074  020123  020050     000
 9331  062101     040  020051  020075  OPR002: .ASCIZ  / ) = /
 9332  062106     000
 9333  062107     122  033113  030461  OPR003: .ASCIZ  /RK611 VECTOR ADDRESS ( /
 9334  062114  053040  041505  047524
 9335  062122  020122  042101  051104
 9336  062130  051505  020123  020050
 9337  062136     000
 9338  062137     122  033113  030461  OPR004: .ASCIZ  /RK611 PRIORITY ( /
 9339  062144  050040  044522  051117
 9340  062152  052111  020131  020050
 9341  062160     000
 9342  062161     040  000040          SPACE2: .ASCIZ  /  /
 9343  062164  005015  051120  043517  ABORT:  .ASCIZ  <15><12>/PROGRAM ABORTED BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
 9344  062172  040522  020115  041101
 9345  062200  051117  042524  020104
 9346  062206  042502  040503  051525
 9347  062214  020105  051105  047522
 9348  062222  020122  044124  042522
 9349  062230  044123  046117  020104
 9350  062236  054105  042503  042105
 9351  062244  042105  005015     000
```

```
                                        .SBTTL  DATA HEADERS

 062251  000124  051505  020124  DH000A: .ASCIZ  /TEST    ERROR/
 062256  020040  042440  051122
 062264  051117  000000
 062267  000116  046525  020040  DH000B: .ASCIZ  /NUM     PC/
 062274  020040  050040  000103
 062302  042524  052123  020040  DH000C: .ASCII  /TEST    TRAP/<15><12>
 062310  020040  051124  050101
 062316  005015
 062320  052516  020115  020040          .ASCIZ  /NUM    PC/
 062326  050040  000103
 062332  047125  041111  051525  DH001:  .ASCIZ  /UNIBUS ADDRESS/
 062340  040440  042104  042522
 062346  051523  000000
 062351  000105  050130  041505  DH002A: .ASCIZ  /EXPECT  ACTUAL/
 062356  020124  040440  052103
 062364  040525  000114
 062370  040526  053114  020105  DH002B: .ASCIZ  /VALVE   VALUE/
 062376  020040  040526  052514
 062404  000105
 062406  040526  052514  020105  DH003A: .ASCIZ  /VALUE   EXPECT  ACTUAL/
 062414  020040  054105  042520
 062422  052103  020040  041501
 062430  052524  046101  000000
 062435  000127  044522  052124  DH003B: .ASCIZ  /WRITTEN VALVE   VALUE/
 062442  047105  053040  046101
 062450  042526  020040  053040
 062456  046101  042525  000000
 062463  000120  042522  020126  DH004A: .ASCIZ  /PREV    EXPECT  ACTUAL/
 062470  020040  042440  050130
 062476  041505  020124  040440
 062504  052103  040525  000114
 062512  040526  052514  020105  DH004B: .ASCIZ  /VALUE   VALVE   VALVE/
 062520  020040  040526  053114
 062526  020105  020040  040526
 062534  053114  000105
 062540  054105  042520  052103  DH006A: .ASCIZ  /EXPECT  ACTUAL/
 062546  020040  041501  052524
 062554  046101  000000
 062557  000126  041505  047524  DH006B: .ASCIZ  /VECTOR  VECTOR/
 062564  020122  053040  041505
 062572  047524  000122
 062576  051120  041517  051505  DH011:  .ASCIZ  /PROCESSOR PRIORITY/
 062604  047523  020122  051120
 062612  047511  044522  054524
 062620  000000
 062621  000105  050130  041505  DH016A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL/
 062626  020124  040440  052103
 062634  040525  020114  042440
 062642  050130  041505  020124
 062650  040440  052103  040525
 062656  000114
 062660  045522  051503  020061  DH016B: .ASCIZ   RKCS1   RKCS1   RKCS2   RKCS2/
 062666  020040  045522  051503
 062674  020061  020040  045522
```

```
 9408  062702  051503  020062  020040
 9409  062710  045522  051503  000062
 9410  062716  054105  042520  052103  DH024A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  SILO/
 9411  062724  020040  041501  052524
 9412  062732  046101  020040  054105
 9413  062740  042520  052103  020040
 9414  062746  041501  052524  046101
 9415  062754  020040  044523  047514
 9416  062762     000
 9417  062763     122  041513  030523  DH024B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2   WORD/
 9418  062770  020040  051040  041513
 9419  062776  030523  020040  051040
 9420  063004  041513  031123  020040
 9421  063012  051040  041513  031123
 9422  063020  020040  053440  051117
 9423  063026  000104
 9424  063030  054105  042520  052103  DH026A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  SILO    EXPECT  ACTUAL/
 9425  063036  020040  041501  052524
 9426  063044  046101  020040  054105
 9427  063052  042520  052103  020040
 9428  063060  041501  052524  046101
 9429  063066  020040  044523  047514
 9430  063074  020040  020040  054105
 9431  063102  042520  052103  020040
 9432  063110  041501  052524  046101
 9433  063116     000
 9434  063117     122  041513  030523  DH026B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2   WORD    DATA    DATA/
 9435  063124  020040  051040  041513
 9436  063132  030523  020040  051040
 9437  063140  041513  031123  020040
 9438  063146  051040  041513  031123
 9439  063154  020040  053440  051117
 9440  063162  020104  020040  042040
 9441  063170  052101  020101  020040
 9442  063176  042040  052101  000101
```

```
                                        .SBTTL  ERROR MESSAGES

063204 047125 054105 042520  EM000:  .ASCIZ  /UNEXPECTED MEMORY PARITY ENABLE TRAP/
063212 052103 042105 046440
063220 046505 051117 020131
063226 040520 044522 054524
063234 042440 040516 046102
063242 020105 051124 050101
063250    000
063251    116 047117 042455  EM1:    .ASCIZ  /NON-EXISTENT MEMORY WHEN ACCESSING RK611 REG/
063256 044530 052123 047105
063264 020124 042515 047515
063272 054522 053440 042510
063300 020116 041501 042503
063306 051523 047111 020107
063314 045522 030466 020061
063322 042522 000107
063326 052101 042524 050115  EM2:    .ASCIZ  /ATTEMPTING TO CLEAR Rk611 WITH A UNIBUS INIT/
063334 044524 043516 052040
063342 020117 046103 040505
063350 020122 045522 030466
063356 020061 044527 044124
063364 040440 052440 044516
063372 052502 020123 047111
063400 052111    000
063403    101 052124 046505  EM3:    .ASCIZ  /ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR/
063410 052120 047111 020107
063416 047524 041440 042514
063424 051101 051040 033113
063432 030461 053440 052111
063440 020110 020101 047503
063446 052116 047522 046114
063454 051105 041440 042514
063462 051101    000
063465    101 052124 046505  EM4:    .ASCI   ATTEMPTING TO WRITE BUS ADD/
063472 052120 047111 020107
063500 047524 053440 044522
063506 042524 041040 051525
063514 040440 042104    000
063521    101 052124 046505  EM5:    .ASCIZ  /ATTEMPTING TO CLEAR BUS ADD WITH A UNIBUS INIT/
063526 052120 047111 020107
063534 047524 041440 042514
063542 051101 041040 051525
063550 040440 042104 053440
063556 052111 020110 020101
063564 047125 041111 051525
063572 044440 044516 000124
063600 052101 042524 050115  EM6:    .ASCII  /ATTEMPTING TO CLEAR BUS ADD/
063606 044524 043516 052040
063614 020117 046103 040505
063622 020122 052502 020123
063630 042101    104
063633    127 052111 020110          .ASCIZ   WITH A CONTROLLER CLEAR/
063640 020101 047503 052116
063646 047522 046114 051105
063654 041440 042514 051101
```

```
 9499  063662      000
 9500  063663      101  052124  046505  EM7:    .ASCIZ  /ATTEMPTING TO WRITE WORD COUNT/
 9501  063670  052120  047111  020107
 9502  063676  047524  053440  044522
 9503  063704  042524  053440  051117
 9504  063712  020104  047503  047125
 9505  063720  000124
 9506  063722  052101  042524  050115  EM8:    .ASCIZ  /ATTEMPTING TO WRITE DISK ADD/
 9507  063730  044524  043516  052040
 9508  063736  020117  051127  052111
 9509  063744  020105  044504  045523
 9510  063752  040440  042104      000
 9511  063757      101  052124  046505  EM9:    .ASCIZ  /ATTEMPTING TO WRITE SPARE REG/
 9512  063764  052120  047111  020107
 9513  063772  047524  053440  044522
 9514  064000  042524  051440  040520
 9515  064006  042522  051040  043505
 9516  064014      000
 9517  064015      101  052124  046505  EM10:   .ASCIZ   ATTEMPTING TO WRITE DRIVE STATUS REG/
 9518  064022  052120  047111  020107
 9519  064030  047524  053440  044522
 9520  064036  042524  042040  044522
 9521  064044  042526  051440  040524
 9522  064052  052524  020123  042522
 9523  064060  000107
 9524  064062  052101  042524  050115  EM11:   .ASCIZ  /ATTEMPTING TO WRITE ERROR REG/
 9525  064070  044524  043516  052040
 9526  064076  020117  051127  052111
 9527  064104  020105  051105  047522
 9528  064112  020122  042522  000107
 9529  064120  052101  042524  050115  EM12:   .ASCIZ  /ATTEMPTING TO WRITE MAINT.REG 2/
 9530  064126  044524  043516  052040
 9531  064134  020117  051127  052111
 9532  064142  020105  040515  047111
 9533  064150  027124  042522  020107
 9534  064156  000062
 9535  064160  052101  042524  050115  EM13:   .ASCIZ  /ATTEMPTING TO WRITE MAINT REG 3/
 9536  064166  044524  043516  052040
 9537  064174  020117  051127  052111
 9538  064202  020105  040515  047111
 9539  064210  020124  042522  020107
 9540  064216  000063
 9541  064220  052101  042524  050115  EM14:   .ASCIZ  /ATTEMPTING TO WRITE ECC PATTERN/
 9542  064226  044524  043516  052040
 9543  064234  020117  051127  052111
 9544  064242  020105  041505  020103
 9545  064250  040520  052124  051105
 9546  064256  000116
 9547  064260  052101  042524  050115  EM15:   .ASCIZ  /ATTEMPTING TO WRITE ECC POSITION/
 9548  064266  044524  043516  052040
 9549  064274  020117  051127  052111
 9550  064302  020105  041505  020103
 9551  064310  047520  044523  044524
 9552  064316  047117      000
 9553  064321      101  052124  046505  EM16:   .ASCIZ   ATTEMPTING TO WRITE CS1/
 9554  064326  052120  047111  020107
```

```
9555  064334  047524  053440  044522
9556  064342  042524  041440  030523
9557  064350     000
9558  064351     101  052124  046505  EM17:   .ASCIZ  "ATTEMPTING TO WRITE ATTENTION SUMMARY/OFFSET"
9559  064356  052120  047111  020107
9560  064364  047524  053440  044522
9561  064372  042524  040440  052124
9562  064400  047105  044524  047117
9563  064406  051440  046525  040515
9564  064414  054522  047457  043106
9565  064422  042523  000124
9566  064426  052101  042524  050115  EM18:   .ASCIZ  /ATTEMPTING TO WRITE CS2/
9567  064434  044524  043516  052040
9568  064442  020117  051127  052111
9569  064450  020105  051503  000062
9570  064456  052101  042524  050115  EM19:   .ASCIZ  /ATTEMPTING TO WRITE DISK CYL ADD/
9571  064464  044524  043516  052040
9572  064472  020117  051127  052111
9573  064500  020105  044504  045523
9574  064506  041440  046131  040440
9575  064514  042104     000
9576  064517     101  052124  046505  EM20.   .ASCIZ  /ATTEMPTING TO WRITE MAINT REG 1/
9577  064524  052120  047111  020107
9578  064532  047524  053440  044522
9579  064540  042524  046440  044501
9580  064546  052116  051040  043505
9581  064554  030440     000
9582  064557     101  052124  046505  EM21:   .ASCII  /ATTEMPTING TO CLEAR BUS ADD /
9583  064564  052120  047111  020107
9584  064572  047524  041440  042514
9585  064600  051101  041040  051525
9586  064606  040440  042104     040
9587  064613     127  052111  020110          .ASCIZ  /WITH A SUB CLEAR/
9588  064620  020101  052523  020102
9589  064626  046103  040505  000122
9590  064634  052101  042524  050115  EM22:   .ASCII  /ATTEMPTING TO CLEAR CS1 /
9591  064642  044524  043516  052040
9592  064650  020117  046103  040505
9593  064656  020122  051503  020061
9594  064664  044527  044124  040440          .ASCIZ  /WITH A SUB CLEAR/
9595  064672  051440  041125  041440
9596  064700  042514  051101     000
9597  064705     101  052124  046505  EM23:   .ASCIZ  /ATTEMPTING TO CLEAR Rk611 WITH A SUBSYSTEM CLEAR/
9598  064712  052120  047111  020107
9599  064720  047524  041440  042514
9600  064726  051101  051040  033113
9601  064734  030461  053440  052111
9602  064742  020110  020101  052523
9603  064750  051502  051531  042524
9604  064756  020115  046103  040505
9605  064764  000122
9606  064766  047111  051124  042040  EM24:   .ASCII  /INTR DID NOT OCCUR WHEN READY AND/<15><12>
9607  064774  042111  047040  052117
9608  065002  047440  041503  051125
9609  065010  053440  042510  020116
9610  065016  042522  042101  020131
```

```
9611  065024  047101  006504     012
9612  065031     111  052116  020122          .ASCIZ  /INTR ENABLE SET IN CS1/
9613  065036  047105  041101  042514
9614  065044  051440  052105  044440
9615  065052  020116  051503  000061
9616  065060  047125  054105  042520  EM25:   .ASCIZ  /UNEXPECTED INTR WHEN PROCESSOR PRIORITY LOWERED/
9617  065066  052103  042105  044440
9618  065074  052116  020122  044127
9619  065102  047105  050040  047522
9620  065110  042503  051523  051117
9621  065116  050040  044522  051117
9622  065124  052111  020131  047514
9623  065132  042527  042522  000104
9624  065140  047111  051124  053040  EM26:   .ASCIZ  /INTR VECTOR ADDRESS INCORRECT/
9625  065146  041505  047524  020122
9626  065154  042101  051104  051505
9627  065162  020123  047111  047503
9628  065170  051122  041505  000124
9629  065176  047111  051124  042040  EM27:   .ASCIZ  /INTR DID NOT CLEAR IN RK611/
9630  065204  042111  047040  052117
9631  065212  041440  042514  051101
9632  065220  044440  020116  045522
9633  065226  030466  000061
9634  065232  054105  042520  052103  EM28:   .ASCIZ  /EXPECTED INTR DID NOT OCCUR AT PROCESSOR PRIORITY/
9635  065240  042105  044440  052116
9636  065246  020122  044504  020104
9637  065254  047516  020124  041517
9638  065262  052503  020122  052101
9639  065270  050040  047522  042503
9640  065276  051523  051117  050040
9641  065304  044522  051117  052111
9642  065312  000131
9643  065314  047125  054105  042520  EM29:   .ASCIZ  /UNEXPECTED INTR OCCURRED AT PROCESSOR PRIORITY/
9644  065322  052103  042105  044440
9645  065330  052116  020122  041517
9646  065336  052503  051122  042105
9647  065344  040440  020124  051120
9648  065352  041517  051505  047523
9649  065360  020122  051120  047511
9650  065366  044522  054524     000
9651  065373     111  052116  020122  EM30:   .ASCIZ  /INTR DID NOT OCCUR WHEN PRIORITY LOWERED/
9652  065400  044504  020104  047516
9653  065406  020124  041517  052503
9654  065414  020122  044127  047105
9655  065422  050040  044522  051117
9656  065430  052111  020131  047514
9657  065436  042527  042522  000104
9658  065444  042523  052124  047111  EM31:   .ASCIZ  /SETTING INTR ENABLE CAUSED EXPECTED INTR/
9659  065452  020107  047111  051124
9660  065460  042440  040516  046102
9661  065466  020105  040503  051525
9662  065474  042105  042440  050130
9663  065502  041505  042524  020104
9664  065510  047111  051124     000
9665  065515     103  047117  051124  EM32:   .ASCIZ  /CONTROLLER CLEAR DID CLEAR PENDING INTR/
9666  065522  046117  042514  020122
```

```
9667  065530  046103  040505  020122
9668  065536  044504  020104  046103
9669  065544  040505  020122  042520
9670  065552  042116  047111  020107
9671  065560  047111  051124     000
9672  065565     101  052124  046505  EM33:   .ASCIZ  /ATTEMPTING TO WRITE ONE WORD OR DATA/
9673  065572  052120  047111  020107
9674  065600  047524  053440  044522
9675  065606  042524  047440  042516
9676  065614  053440  051117  020104
9677  065622  051117  042040  052101
9678  065630  000101
9679  065632  052101  042524  050115  EM34:   .ASCIZ  /ATTEMPTING TO READ SILO WHEN EMPTY/
9680  065640  044524  043516  052040
9681  065646  020117  042522  042101
9682  065654  051440  046111  020117
9683  065662  044127  047105  042440
9684  065670  050115  054524     000
9685  065675     101  052124  046505  EM35:   .ASCIZ  /ATTEMPTING TO CLEAR DATA LATA/
9686  065702  052120  047111  020107
9687  065710  047524  041440  042514
9688  065716  051101  042040  052101
9689  065724  020101  040514  040524
9690  065732     000
9691  065733     101  052124  046505  EM36:   .ASCIZ  /ATTEMPTING TO READ SILO CONTAINING ONE WORD/
9692  065740  052120  047111  020107
9693  065746  047524  051040  040505
9694  065754  020104  044523  047514
9695  065762  041440  047117  040524
9696  065770  047111  047111  020107
9697  065776  047117  020105  047527
9698  066004  042122     000
9699  066007     101  052124  046505  EM37:   .ASCIZ  /ATTEMPTING TO LOAD SILO/
9700  066014  052120  047111  020107
9701  066022  047524  046040  040517
9702  066030  020104  044523  047514
9703  066036     000
9704  066037     101  052124  046505  EM38:   .ASCIZ  /ATTEMPTING TO READ SILO/
9705  066044  052120  047111  020 07
9706  066052  047524  051040  040505
9707  066060  020104  044523  047514
9708  066066     000
9709  066067     101  052124  046505  EM39:   .ASCIZ  /ATTEMPTING TO LOAD FULL SILO (66 WORDS)/
9710  066074  052120  047111  020107
9711  066102  047524  046040  040517
9712  066110  020104  052506  046114
9713  066116  051440  046111  020117
9714  066124  033050  020066  047527
9715  066132  042122  024523     000
9716  066137     104  052101  020101  EM40:   .ASCIZ  /DATA LATE DID NOT CAUSE EXPECTED INTR/
9717  066144  040514  042524  042040
9718  066152  042111  047040  052117
9719  066160  041440  052501  042523
9720  066166  042440  050130  041505
9721  066174  042524  020104  047111
9722  066202  051124     000
```

```
 9723  066205     125  042516  050130  EM41:   .ASCIZ  /UNEXPECTED INTR DUE TO UNCLEARED CONTROLLER ERROR/
 9724  066212  041505  042524  020104
 9725  066220  047111  051124  042040
 9726  066226  042525  052040  020117
 9727  066234  047125  046103  040505
 9728  066242  042522  020104  047503
 9729  066250  052116  047522  046114
 9730  066256  051105  042440  051122
 9731  066264  051117     000
 9732  066267     103  047117  051124  EM42:   .ASCII  /CONTROLLER CLEAR DID NOT CLEAR PENDING INTR/<15><12>
 9733  066274  046117  042514  020122
 9734  066302  046103  040505  020122
 9735  066310  044504  020104  047516
 9736  066316  020124  046103  040505
 9737  066324  020122  042520  042116
 9738  066332  047111  020107  047111
 9739  066340  051124  005015
 9740  066344  052504  020105  047524          .ASCIZ  /DUE TO CONTROLLER ERROR/
 9741  066352  041440  047117  051124
 9742  066360  046117  042514  020122
 9743  066366  051105  047522  000122
 9744  066374  047503  052116  047522  EM43:   .ASCIZ  /CONTROLLER ERROR CAUSED INTR WITH INTR ENABLE RESET/
 9745  066402  046114  051105  042440
 9746  066410  051122  051117  041440
 9747  066416  052501  042523  020104
 9748  066424  047111  051124  053440
 9749  066432  052111  020110  047111
 9750  066440  051124  042440  040516
 9751  066446  046102  020105  042522
 9752  066454  042523  000124
 9753  066460  047111  051124  042040  EM44:   .ASCII  /INTR DID NOT OCCUR WHEN INTR ENABLE SET/
 9754  066466  042111  047040  052117
 9755  066474  047440  041503  051125
 9756  066502  053440  042510  020116
 9757  066510  047111  051124  042440
 9758  066516  040516  046102  020105
 9759  066524  042523     124
 9760  066527     015  053412  052111          .ASCIZ  <15><12>/WITH INTR PENDING DUE TO DATA LATE/
 9761  066534  020110  047111  051124
 9762  066542  050040  047105  044504
 9763  066550  043516  042040  042525
 9764  066556  052040  020117  040504
 9765  066564  040524  046040  052101
 9766  066572  000105
 9767  066574  051503  020061  047111  EM1000: .ASCIZ  /CS1 INCORRECT/
 9768  066602  047503  051122  041505
 9769  066610  000124
 9770  066612  047527  042122  041440  EM1001: .ASCIZ  /WORD COUNT INCORRECT/
 9771  066620  052517  052116  044440
 9772  066626  041516  051117  042522
 9773  066634  052103     000
 9774  066637     102  051525  040440  EM1002: .ASCIZ  /BUS ADD INCORRECT/
 9775  066644  042104  044440  041516
 9776  066652  051117  042522  052103
 9777  066660     000
 9778  066661     104  051511  020113  EM1003: .ASCIZ  /DISK ADD INCORRECT/
```

```
9779  066666  042101  020104  047111
9780  066674  047503  051122  041505
9781  066702  000124
9782  066704  051503  020062  047111  EM1004: .ASCIZ  /CS2 INCORRECT/
9783  066712  047503  051122  041505
9784  066720  000124
9785  066722  051104  053111  020105  EM1005: .ASCIZ  /DRIVE STATUS REG INCORRECT/
9786  066730  052123  052101  051525
9787  066736  051040  043505  044440
9788  066744  041516  051117  042522
9789  066752  052103     000
9790  066755     105  051122  051117  EM1006: .ASCIZ  /ERROR REG INCORRECT/
9791  066762  051040  043505  044440
9792  066770  041516  051117  042522
9793  066776  052103     000
9794  067001     101  052124  047105  EM1007: .ASCIZ  "ATTENTION SUMMARY/OFFSET INCORRECT"
9795  067006  044524  047117  051440
9796  067014  046525  040515  054522
9797  067022  047457  043106  042523
9798  067030  020124  047111  047503
9799  067036  051122  041505  000124
9800  067044  040504  040524  041040  EM1008: .ASCIZ  /DATA BUFFER INCORRECT/
9801  067052  043125  042506  020122
9802  067060  047111  047503  051122
9803  067066  041505  000124
9804  067072  040515  047111  020124  EM1009: .ASCIZ  /MAINT REG 1 INCORRECT/
9805  067100  042522  020107  020061
9806  067106  047111  047503  051122
9807  067114  041505  000124
9808  067120  041505  020103  047520  EM1012: .ASCIZ  /ECC POS INCORRECT/
9809  067126  020123  047111  047503
9810  067134  051122  041505  000124
9811  067142  041505  020103  040520  EM1013: .ASCIZ  /ECC PAT INCORRECT/
9812  067150  020124  047111  047503
9813  067156  051122  041505  000124
9814  067164  051503  020061  044103  EM1014: .ASCIZ  /CS1 CHANGED AFTER READING ALL REGS/
9815  067172  047101  042507  020104
9816  067200  043101  042524  020122
9817  067206  042522  042101  047111
9818  067214  020107  046101  020114
9819  067222  042522  051507     000
9820  067227     103  031123  041440  EM1015: .ASCIZ  /CS2 CHANGED AFTER READING ALL REGS/
9821  067234  040510  043516  042105
9822  067242  040440  052106  051105
9823  067250  051040  040505  044504
9824  067256  043516  040440  046114
9825  067264  051040  043505  000123
9826  067272  054503  020114  042101  EM1016: .ASCIZ  /CYL ADD INCORRECT/
9827  067300  020104  047111  047503
9828  067306  051122  041505  000124
9829  067314  051503  020061  040527  EM1017: .ASCIZ  /CS1 WAS MODIFIED/
9830  067322  020123  047515  044504
9831  067330  044506  042105     000
9832  067335     127  051117  020104  EM1018: .ASCIZ  /WORD COUNT WAS MODIFIED/
9833  067342  047503  047125  020124
9834  067350  040527  020123  047515
```

```
 9835  067356  044504  044506  042105
 9836  067364     000
 9837  067365     102  051525  040440  EM1019: .ASCIZ  /BUS ADD WAS MODIFIED/
 9838  067372  042104  053440  051501
 9839  067400  046440  042117  043111
 9840  067406  042511  000104
 9841  067412  044504  045523  040440  EM1020: .ASCIZ  /DISK ADDRESS REG WAS MODIFIED/
 9842  067420  042104  042522  051523
 9843  067426  051040  043505  053440
 9844  067434  051501  046440  042117
 9845  067442  043111  042511  000104
 9846  067450  051503  020062  040527  EM1021: .ASCIZ  /CS2 WAS MODIFIED/
 9847  067456  020123  047515  044504
 9848  067464  044506  042105     000  EM1022: .ASCIZ  /DRIVE STATUS REG WAS MODIFIED/
 9849  067471     104  044522  042526
 9850  067476  051440  040524  052524
 9851  067504  020123  042522  020107
 9852  067512  040527  020123  047515
 9853  067520  044504  044506  042105
 9854  067526     000
 9855  067527     105  051122  051117  EM1023: .ASCIZ  /ERROR REG WAS MODIFIED/
 9856  067534  051040  043505  053440
 9857  067542  051501  046440  042117
 9858  067550  043111  042511  000104
 9859  067556  052101  042524  052116  EM1024: .ASCIZ  "ATTENTION SUMMARY/OFFSET WAS MODIFIED"
 9860  067564  047511  020116  052523
 9861  067572  046515  051101  027531
 9862  067600  043117  051506  052105
 9863  067606  053440  051501  046440
 9864  067614  042117  043111  042511
 9865  067622  000104
 9866  067624  054503  020114  042101  EM1025: .ASCIZ  /CYL ADD WAS MODIFIED/
 9867  067632  020104  040527  020123
 9868  067640  047515  044504  044506
 9869  067646  042105     000
 9870  067651     115  044501  052116  EM1026: .ASCIZ  /MAINT REG 1 WAS MODIFIED/
 9871  067656  051040  043505  030440
 9872  067664  053440  051501  046440
 9873  067672  042117  043111  042511
 9874  067700  000104
 9875  067702  041505  020103  047520  EM1029: .ASCIZ  /ECC POS WAS MODIFIED/
 9876  067710  020123  040527  020123
 9877  067716  047515  044504  044506
 9878  067724  042105     000
 9879  067727     105  041503  050040  EM1030: .ASCIZ  /ECC PAT WAS MODIFIED/
 9880  067734  052101  053440  051501
 9881  067742  046440  042117  043111
 9882  067750  042511  000104
 9883                                  .EVEN
 9884  067754  000400                  SAVVEC: .BLKW   400              ;STORAGE FOR TRAP CATCHER
 9885  070754     100     100          SILO:   .BYTE   100,100         ;CONFIGURATIONION FOR SILO TEST
 9886  070756     101     101                  .BYTE   101,101
 9887  070760     102     102                  .BYTE   102,102
 9888  070762     103     103                  .BYTE   103,103
 9889  070764     104     104                  .BYTE   104,104
 9890  070766     105     105                  .BYTE   105,105
```

```
 9891  070770      106      106                    .BYTE   106,106
 9892  070772      107      107                    .BYTE   107,107
 9893  070774      110      110                    .BYTE   110,110
 9894  070776      111      111                    .BYTE   111,111
 9895  071000      112      112                    .BYTE   112,112
 9896  071002      113      113                    .BYTE   113,113
 9897  071004      114      114                    .BYTE   114,114
 9898  071006      115      115                    .BYTE   115,115
 9899  071010      116      116                    .BYTE   116,116
 9900  071012      117      117                    .BYTE   117,117
 9901  071014      120      120                    .BYTE   120,120
 9902  071016      121      121                    .BYTE   121,121
 9903  071020      122      122                    .BYTE   122,122
 9904  071022      123      123                    .BYTE   123,123
 9905  071024      124      124                    .BYTE   124,124
 9906  071026      125      125                    .BYTE   125,125
 9907  071030      126      126                    .BYTE   126,126
 9908  071032      127      127                    .BYTE   127,127
 9909  071034      130      130                    .BYTE   130,130
 9910  071036      131      131                    .BYTE   131,131
 9911  071040      132      132                    .BYTE   132,132
 9912  071042      133      133                    .BYTE   133,133
 9913  071044      134      134                    .BYTE   134,134
 9914  071046      135      135                    .BYTE   135,135
 9915  071050      136      136                    .BYTE   136,136
 9916  071052      137      137                    .BYTE   137,137
 9917  071054      140      140                    .BYTE   140,140
 9918  071056      141      141                    .BYTE   141,141
 9919  071060      142      142                    .BYTE   142,142
 9920  071062      143      143                    .BYTE   143,143
 9921  071064      144      144                    .BYTE   144,144
 9922  071066      145      145                    .BYTE   145,145
 9923  071070      146      146                    .BYTE   146,146
 9924  071072      147      147                    .BYTE   147,147
 9925  071074      150      150                    .BYTE   150,150
 9926  071076      151      151                    .BYTE   151,151
 9927  071100      152      152                    .BYTE   152,152
 9928  071102      153      153                    .BYTE   153,153
 9929  071104      154      154                    .BYTE   154,154
 9930  071106      155      155                    .BYTE   155,155
 9931  071110      156      156                    .BYTE   156,156
 9932  071112      157      157                    .BYTE   157,157
 9933  071114      160      160                    .BYTE   160,160
 9934  071116      161      161                    .BYTE   161,161
 9935  071120      162      162                    .BYTE   162,162
 9936  071122      163      163                    .BYTE   163,163
 9937  071124      164      164                    .BYTE   164,164
 9938  071126      165      165                    .BYTE   165,165
 9939  071130      166      166                    .BYTE   166,166
 9940  071132      167      167                    .BYTE   167,167
 9941  071134      170      170                    .BYTE   170,170
 9942  071136      171      171                    .BYTE   171,171
 9943  071140      172      172                    .BYTE   172,172
 9944  071142      173      173                    .BYTE   173,173
 9945  071144      174      174                    .BYTE   174,174
 9946  071146      175      175                    .BYTE   175,175
```

```
 9947  071150      176      176                    .BYTE   176,176
 9948  071152      177      177                    .BYTE   177,177
 9949  071154      200      200                    .BYTE   200,200
 9950  071156      201      201                    .BYTE   201,201
 9951              000001                  .END
```

```
ABASE = 177440           1066#   1301    1342
ABORT   062164           8634    9343#
ACDW1 = 000000           1301    1344
ACDW2 = 000000           1301    1345
ACLO  = 000010           1161#
ACPUOP= 000000           1301    1316
ADDW0 = 000000           1301
ADDW1 = 000000           1301
ADDW10= 000000           1301
ADDW11= 000000           1301
ADDW12= 000000           1301
ADDW13= 000000           1301
ADDW14= 000000           1301
ADDW15= 000000           1301
ADDW2 = 000000           1301
ADDW3 = 000000           1301
ADDW4 = 000000           1301
ADDW5 = 000000           1301
ADDW6 = 000000           1301
ADDW7 = 000000           1301
ADDW8 = 000000           1301
ADDW9 = 000000           1301
ADEVCT= 000000           1301    1307
ADEVM = 000000           1301    1343
AENV  = 000000           1301    1312
AENVM = 000000           1301    1313
AFATAL= 000000           1301    1304
AMADR1= 000000           1301    1329
AMADR2= 000000           1301    1333
AMADR3= 000000           1301    1336
AMADR4= 000000           1301    1339
AMAMS1= 000000           1301    1323
AMAMS2= 000000           1301    1331
AMAMS3= 000000           1301    1334
AMAMS4= 000000           1301    1337
AMSGAD= 000000           1301    1309
AMSGLG= 000000           1301    1310
AMSGTY= 000000           1301    1303
AMTYP1= 000000           1301    1324
AMTYP2= 000000           1301    1332
AMTYP3= 000000           1301    1335
AMTYP4= 000000           1301    1338
APASS = 000000           1301    1306
APRIOR= 000005           1065#   1301
APTCSU= 000040           8511#   8673
APTENV= 000001           8467    8509#   8543    8666
APTSIZ= 000200           1639    8508#
APTSPO= 000100           8469    8510#   8668
ASWREG= 000000           1301    1314
ATESTN= 000000           1301    1305
AUNIT = 000000           1301    1308
AUSWR = 000000           1301    1315
AVECT1= 120210           1064#   1301    1340
AVECT2= 000000           1301    1341
BA1   = 000020           1124#
BA16  = 000400           1109#
```

```
BA17   = 001000          1110#
BIT0   = 000001          1046#  1106  1139  1158
BIT00  = 000001          1036#  1046
BIT01  = 000002          1035#  1045
BIT02  = 000004          1034#  1044
BIT03  = 000010          1033#  1043
BIT04  = 000020          1032#  1042
BIT05  = 000040          1031#  1041
BIT06  = 000100          1030#  1040
BIT07  = 000200          1029#  1039
BIT08  = 000400          1028#  1038  8338
BIT09  = 001000          1027#  1037  8356  8554
BIT1   = 000002          1045#  1140
BIT10  = 002000          1026#  1111  1130  1149  1182  1197  8531
BIT11  = 004000          1025#  1112  1131  1150  1167  1183  1198  8363
BIT12  = 010000          1024#  1114  1132  1151  1184
BIT13  = 020000          1023#  1115  1133  1152  1168  1185  8538
BIT14  = 040000          1022#  1116  1134  1153  1169  1186  8324
BIT15  = 100000          1021#  1117  1118  1135  1154  1170  1187
BIT2   = 000004          1044#  1141  1160
BIT3   = 000010          1043#  1123  1142  1161
BIT4   = 000020          1042#  1124  1143  1162  1176  1191
BIT5   = 000040          1041#  1125  1144  1163  1177  1192
BIT6   = 000100          1040#  1107  1126  1145  1164  1178  1193
BIT7   = 000200          1039#  1108  1127  1146  1165  1179  1194
BIT8   = 000400          1038#  1109  1128  1147  1166  1180  1195
BIT9   = 001000          1037#  1110  1129  1148  1181  1196
BPTVEC= 000014           1053#
BSE    = 000200          1146#
CCLR   = 100000          1118#  1874  1966  2051  2053  2082  2152  2181  2253  2297  2383  2421  2507
                         2545   2631  2669  2755  2794  2885  2897  2926  3017  3029  3058  3149  3161
                         3190   3281  3293  3313  3403  3487  3522  3606  3641  3725  3760  3844  3881
                         3967   4003  4089  4125  4211  4247  4333  4369  4455  4491  4577  4613  4699
                         4735   4821  4858  4944  4983  5069  5108  5194  5233  5319  5348  5440  5525
                         5609   5693  5783  5869  5903  5989  6023  6109  6143  6229  6263  6350  6389
                         6475   6513  6600  6638  6725  6758  6843  6945  7257  7290  7292  7327  7341
                         7375   7483  7515  7631  7747  7863  7963  8045  8058  8074  8101  8135  8154
                         8156   8191  8221
CDT    = 002000          1111#
CERR   = 100000          1117#  7332  7472  8036  8067
CFMT   = 010000          1114#  2874  3006  3138  3270
CHKPAR  055264           1726   1781  2034  8278#  9154
CKSWR  = 104407          8323   8527  8553  9200#
CLEAR  = 000005          1091#
COE    = 001000          1148#
CONFG1  002012           1577#  2792*  2895*  2896  2924*  3027*  3028  3056*  3159*  3160  3188*  3291*  3292
                         4856*  4954*  4955  4981*  5079*  5080  5106*  5204*  5205  5231*  5329*  5330
CONFIG  002010           1576#  1964*  2076*  2083  2085  2087  2154  2157  2162  2164*  2179*  2295*  2303
                         2305  2393*  2419*  2427  2429  2517*  2543*  2551  2553  2641*  2667*  2675  2677
                         2765*  2791*  2800  2802  2874  2896*  2897*  2923*  2932  2934  3006  3028*  3029*
                         3055*  3064  3066  3138  3160*  3161*  3187*  3196  3198  3270  3292*  3293*  3311*
                         3401*  3408  3410  3489  3492  3498*  3520*  3527  3529  3608  3611  3617*  3639*
                         3646  3648  3727  3730  3736*  3758*  3765  3767  3846  3849  3855*  3879*  3887
                         3889  3977*  4001*  4009  4011  4099*  4123*  4131  4133  4221*  4245*  4253  4255
                         4343*  4367*  4375  4377  4465*  4489*  4497  4499  4587*  4611*  4619  4621  4709*
                         4733*  4741  4743  4831*  4855*  4864  4866  4955*  4956*  4980*  4989  4991  5080*
```

```
                        5081*   5105*   5114    5116    5205*   5206*   5230*   5239    5241    5330*   5331*   5438*   5523*
                        5607*   5691*   5781*   5789    5791    5879*   5901*   5909    5911    5999*   6021*   6029    6031
                        6119*   6141*   6149    6151    6239*   6261*   6269    6271    6364*   6386*   6394    6396    6489*
                        6511*   6519    6521    6614*   6636*   6644    6646    6739*   6756*   6841*   7377*   7509*   7517
                        7589    7591    7596*   7625*   7633    7705    7707    7712*   7741*   7749    7821    7823    7828*
                        7857*   7665    7937    7939    7944*   9213
CR      = 000015         961*   8712    8722
CRLF    = 000200         962*   1664    8683    8722
CTO     = 004000        1112*   2803    2935    3067    3199
DCK     = 100000        1154*
DDISP   = 177570         968*   1274    1627
DDT     = 000400        1166*
DF000     061556        1528    9231*
DF001     061562        1367    9233*
DF002     061602        1372    9241*
DF003     061626        1377    9251*
DF004     061652        1382    9261*
DF005     061676        1387    1422    1427    1500    1505    1511    1517    9271*
DF006     061712        1392    1397    1402    1523    9277*
DF011     061736        1407    1412    1417    9287*
DF016     061756        1432    1437    1442    1447    1453    1459    1490    1495    9295*
DF024     062002        1464    1469    9305*
DF026     062026        1475    1480    1485    9315*
DH000A    062251        1521    9235    9243    9253    9263    9273    9279    9289    9297    9307    9317    9354*
DH000B    062267        9237    9245    9255    9265    9275    9281    9291    9299    9309    9319    9357*
DH000C    062302        1526    9359*
DH001     062332        9239    9364*
DH002A    062351        9247    9367*
DH002B    062370        9249    9370*
DH003A    062406        9257    9373*
DH003B    062435        9259    9377*
DH004A    062463        9267    9381*
DH004B    062512        9269    9385*
DH006A    062540        9283    9301    9389*
DH006B    062557        9285    9392*
DH011     062576        9293    9395*
DH016A    062621        9399*
DH016B    062660        9303    9405*
DH024A    062716        9311    9410*
DH024B    062763        9313    9417*
DH026A    063030        9321    9424*
DH026B    063117        9323    9434*
DI      = 040000        1116*   2803    2935    3067    3199
DISPLA    001142        1274*   1627*   1635*   8378*   8530*
DISPRE    000174        1206*   1635
DLT     = 100000        1135*   4867    4992    5117    5242    7331    7473    8035    8066
DMD     = 000040        1177*
DRA     = 000001        1158*
DRDY    = 000200        1165*
DROT    = 000040        1163*
DRPAR   = 000010        1142*
DRVMSK= 000007          1122*
DSC     = 040000        1169*
DSWR    = 177570         967*   1273    1626
DTE     = 010000        1151*
DTYE    = 000040        1144*
```

```
DT000   061412          1527    9209#
DT001   061416          1366    9210#
DT002   061424          1371    9211#
DT003   061434          1376    9213#
DT004   061446          1381    9215#
DT005   061460          1386    1421    1426    1499    1504    1510    1516    9217#
DT006   061464          1391    1396    1401    1522    9218#
DT011   061474          1406    1411    1416    9220#
DT016   061502          1431    1436    1441    1446    1452    1458    1489    1494    9221#
DT024   061516          1463    1468    9223#
DT026   061534          1473    1479    1484    9226#
ECCW  = 020000          1185#   1826    1828    1831    1919    1921    1924    2013    2015    2134    2136    2235    2237
                        2363    2365    2487    2489    2611    2613    2735    2737    2860    2862    2992    2994    3124
                        3126    3256    3258    3365    3367    3467    3469    3586    3588    3705    3707    3824    3826
                        3947    3949    4069    4071    4191    4193    4313    4315    4435    4437    4557    4559    4679
                        4681    4801    4803    4924    4926    5049    5051    5174    5176    5299    5301    5408    5410
                        5492    5494    5577    5579    5661    5663    5745    5747    5849    5851    5969    5971    6089
                        6091    6209    6211    6272    6274    6276    6353    6355    6397    6399    6401    6478    6480
                        6522    6524    6526    6603    6605    6647    6649    6651    6728    6730    6810    6812    6895
                        6897    7431    7433    7570    7572    7686    7688    7802    7804    7918    7920
ECH   = 000100          1145#
EMTVEC= 000030          1056#   1610*   1611*
EM000   063204          1525    9445#
EM1     063251          1364    9452#
EM10    064015          5439    9517#
EM1000  066574          1435    1440    1451    1462    1472    1493    1786    1879    2807    2891    2939    3023    3071
                        3155    3203    3287    9767#
EM1001  066612          2088    3413    3532    3651    3770    5373    9770#
EM1002  066637          1791    1884    1972    2038    2059    2309    2389    2433    2513    2557    2637    2681    2761
                        5358    5367    9774#
EM1003  066661          1796    1889    2189    2258    3893    3973    4015    4095    4137    4217    4259    4339    5378
                        9778#
EM1004  066704          1445    1457    1467    1478    1488    1807    1900    4871    4950    4996    5075    5121    5200
                        5246    5325    5389    7412    7551    7667    7783    7899    9782#
EM1005  066722          1812    1905    5394    9785#
EM1006  066755          1817    1910    5399    9790#
EM1007  067001          1801    1894    4381    4461    4503    4583    4625    4705    4747    4827    5383    9794#
EM1008  067044          1483    7441    7455    7592    7708    7824    7940    9800#
EM1009  067072          1430    1832    1925    5413    6280    6360    6405    6485    6530    6610    6655    6735    9804#
EM1012  067120          1843    1936    5424    9808#
EM1013  067142          1837    1930    5418    9811#
EM1014  067164          1849    1942    9814#
EM1015  067227          1855    1948    9820#
EM1016  067272          1822    1915    5404    5795    5875    5915    5995    6035    6115    6155    6235    9826#
EM1017  067314          1978    2094    2195    2316    2440    2564    2688    3320    3420    3539    3658    3777    3900
                        4022    4144    4266    4388    4510    4632    4754    4878    5003    5128    5253    5447    5532
                        5616    5700    5802    5922    6042    6162    6287    6412    6537    6662    6765    6850    7383
                        7522    7638    7754    7870    9829#
EM1018  067335          2159    2205    2322    2446    2570    2694    2819    2951    3083    3215    3330    3494    3613
                        3732    3851    3912    4034    4156    4278    4400    4522    4644    4766    4890    5015    5140
                        5265    5457    5542    5626    5710    5814    5934    6054    6174    6299    6424    6549    6674
                        6775    6860    7393    7532    7648    7764    7880    9832#
EM1019  067365          2099    2200    2813    2945    3077    3209    3325    3426    3545    3664    3783    3906    4028
                        4150    4272    4394    4516    4638    4760    4884    5009    5134    5259    5452    5537    5621
                        5705    5808    5928    6048    6168    6293    6418    6543    6668    6770    6855    7388    7527
                        7643    7759    7875    9837#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM1020  067412          1983  2104  2328  2452  2576  2700  2825  2957  3089  3221  3335  3432  3551
                        3670  3789  4406  4528  4650  4772  4896  5021  5146  5271  5462  5547  5631
                        5715  5820  5940  6060  6180  6305  6430  6555  6680  6780  6865  7398  7537
                        7653  7769  7885  9841*
EM1021  067450          1994  2115  2216  2341  2465  2589  2713  2838  2970  3102  3234  3346  3445
                        3564  3683  3802  3925  4047  4169  4291  4413  4535  4657  4779  5473  5558
                        5642  5726  5833  5953  6073  6193  6318  6443  6568  6693  6791  6876  9846*
EM1022  067471          1999  2120  2221  2347  2471  2595  2719  2844  2976  3108  3240  3351  3451
                        3570  3689  3808  3931  4053  4175  4297  4419  4541  4663  4785  4908  5033
                        5158  5283  5478  5563  5647  5731  5839  5959  6079  6199  6324  6449  6574
                        6699  6796  6881  7417  7556  7672  7788  7904  9849*
EM1023  067527          2004  2125  2226  2352  2476  2600  2724  2849  2981  3113  3245  3356  3456
                        3575  3694  3813  3936  4058  4180  4302  4424  4546  4668  4790  4913  5038
                        5163  5288  5483  5568  5652  5736  5844  5964  6084  6204  6329  6454  6579
                        6704  6801  6886  7422  7561  7677  7793  7909  9855*
EM1024  067556          1988  2109  2210  2334  2458  2582  2706  2831  2963  3095  3227  3340  3438
                        3557  3676  3795  3918  4040  4162  4284  4902  5027  5152  5277  5467  5552
                        5636  5720  5826  5946  6066  6186  6311  6436  6561  6686  6785  6870  7542
                        7658  7774  7890  9859*
EM1025  067624          2009  2130  2231  2358  2482  2606  2730  2855  2987  3119  3251  3361  3462
                        3581  3700  3819  3942  4064  4186  4308  4430  4552  4674  4796  4919  5044
                        5169  5294  5488  5573  5657  5741  6335  6460  6585  6710  6806  6891  7427
                        7566  7682  7798  7914  9866*
EM1026  067651          2018  2139  2240  2368  2492  2616  2740  2865  2997  3129  3261  3370  3472
                        3591  3710  3829  3952  4074  4196  4318  4440  4562  4684  4806  4929  5054
                        5179  5304  5497  5582  5666  5750  5854  5974  6094  6214  6815  6900  7436
                        7575  7691  7807  7923  9870*
EM1029  067702          2029  2150  2251  2380  2504  2628  2752  2882  3014  3146  3278  3381  3484
                        3603  3722  3841  3964  4086  4208  4330  4452  4574  4696  4818  4941  5066
                        5191  5316  5508  5593  5677  5761  5866  5986  6106  6226  6347  6472  6597
                        6722  6826  6911  7403  7447  7586  7702  7818  7934  9875*
EM1030  067727          2023  2144  2245  2374  2498  2622  2746  2871  3003  3135  3267  3375  3478
                        3597  3716  3835  3958  4080  4202  4324  4446  4568  4690  4812  4935  5060
                        5185  5310  5502  5587  5671  5755  5860  5980  6100  6220  6341  6466  6591
                        6716  6820  6905  7580  7696  7812  7928  9879*
EM11    064062          5524  9524*
EM12    064120          5608  9529*
EM13    064160          5692  9535*
EM14    064220          6757  9541*
EM15    064260          6842  9547*
EM16    064321          2793  2925  3057  3189  9553*
EM17    064351          4368  4490  4612  4734  9558*
EM18    064426          4857  4982  5107  5232  9566*
EM19    064456          5782  5902  6022  6142  9570*
EM2     063326          1779  9460*
EM2N    001310          1369* 1779* 1786* 1791* 1796* 1801* 1807* 1812* 1817* 1822* 1832* 1837* 1843*
                        1849* 1855* 1873* 1879* 1884* 1889* 1894* 1900* 1905* 1910* 1915* 1925* 1930*
                        1936* 1942* 1948* 2158* 2159* 3493* 3494* 3612* 3613* 3731* 3732* 3850* 3851*
                        5369* 5373* 5378* 5383* 5389* 5394* 5399* 5404* 5413* 5418* 5424*
EM20    064517          6262  6387  6512  6637  9576*
EM21    064557          5357  9582*
EM22    064634          5366  9590*
EM23    064705          5369  9597*
EM24    064766          1384  9606*
EM25    065060          1389  9616*
EM26    065140          1394  9624*
```

```
EM27    065176          1399    9629#
EM28    065232          1404    9634#
EM29    065314          1409    1492    9643#
EM3     063403          1873    2158    2257    2388    2512    2636    2760    2890    3022    3154    3286    3493    3612
                        3731    3850    3972    4094    4216    4338    4460    4582    4704    4826    4949    5074    5199
                        5324    5874    5994    6114    6234    6359    6484    6609    6734    9468#
EM3N    001320          1374#   1965*   1972*   1978*   1963*   1988*   1994*   1999*   2004*   2009*   2018*   2023*   2029*
                        2077*   2088*   2094*   2099*   2104*   2109*   2115*   2120*   2125*   2130*   2139*   2144*   2150*
                        2180*   2189*   2195*   2200*   2205*   2210*   2216*   2221*   2226*   2231*   2240*   2245*   2251*
                        2296*   2309*   2316*   2322*   2328*   2334*   2341*   2347*   2352*   2358*   2368*   2374*   2380*
                        2420*   2433*   2440*   2446*   2452*   2458*   2465*   2471*   2476*   2482*   2492*   2498*   2504*
                        2544*   2557*   2564*   2570*   2576*   2582*   2589*   2595*   2600*   2606*   2616*   2622*   2628*
                        2668*   2681*   2688*   2694*   2700*   2706*   2713*   2719*   2724*   2730*   2740*   2746*   2752*
                        2793*   2807*   2813*   2819*   2825*   2831*   2838*   2844*   2849*   2855*   2865*   2871*   2882*
                        2925*   2939*   2945*   2951*   2957*   2963*   2970*   2976*   2981*   2987*   2997*   3003*   3014*
                        3057*   3071*   3077*   3083*   3089*   3095*   3102*   3108*   3113*   3119*   3129*   3135*   3146*
                        3189*   3203*   3209*   3215*   3221*   3227*   3234*   3240*   3245*   3251*   3261*   3267*   3278*
                        3312*   3320*   3325*   3330*   3335*   3340*   3346*   3351*   3356*   3361*   3370*   3375*   3381*
                        3402*   3413*   3420*   3426*   3432*   3438*   3445*   3451*   3456*   3462*   3472*   3478*   3484*
                        3521*   3532*   3539*   3545*   3551*   3557*   3564*   3570*   3575*   3581*   3591*   3597*   3603*
                        3640*   3651*   3658*   3664*   3670*   3676*   3683*   3689*   3694*   3700*   3710*   3716*   3722*
                        3759*   3770*   3777*   3783*   3789*   3795*   3802*   3808*   3813*   3819*   3829*   3835*   3841*
                        3880*   3893*   3900*   3906*   3912*   39.8*   3925*   3931*   3936*   3942*   3952*   3958*   3964*
                        4002*   4015*   4022*   4028*   4034*   4040*   4047*   4053*   4058*   4064*   4074*   4080*   4086*
                        4124*   4137*   4144*   4150*   4156*   4162*   4169*   4175*   4180*   4186*   4196*   4202*   4208*
                        4246*   4259*   4266*   4272*   4278*   4284*   4291*   4297*   4302*   4308*   4318*   4324*   4330*
                        4368*   4381*   4388*   4394*   4400*   4406*   4413*   4419*   4424*   4430*   4440*   4446*   4452*
                        4490*   4503*   4510*   4516*   4522*   4528*   4535*   4541*   4546*   4552*   4562*   4568*   4574*
                        4612*   4625*   4632*   4638*   4644*   4650*   4657*   4663*   4668*   4674*   4684*   4690*   4696*
                        4734*   4747*   4754*   4760*   4766*   4772*   4779*   4785*   4790*   4796*   4806*   4812*   4818*
                        4857*   4871*   4878*   4884*   4890*   4896*   4902*   4908*   4913*   4919*   4929*   4935*   4941*
                        4982*   4996*   5003*   5009*   5015*   5021*   5027*   5033*   5038*   5044*   5054*   5060*   5066*
                        5107*   5121*   5128*   5134*   5140*   5146*   5152*   5158*   5163*   5169*   5179*   5185*   5191*
                        5232*   5246*   5253*   5259*   5265*   5271*   5277*   5283*   5288*   5294*   5304*   5310*   5316*
                        5439*   5447*   5452*   5457*   5462*   5467*   5473*   5478*   5483*   5488*   5497*   5502*   5508*
                        5524*   5532*   5537*   5542*   5547*   5552*   5558*   5563*   5568*   5573*   5582*   5587*   5593*
                        5608*   5616*   5621*   5626*   5631*   5636*   5642*   5647*   5652*   5657*   5666*   5671*   5677*
                        5692*   5700*   5705*   5710*   5715*   5720*   5726*   5731*   5736*   5741*   5750*   5755*   5761*
                        5782*   5795*   5802*   5808*   5814*   5820*   5826*   5833*   5839*   5844*   5854*   5860*   5866*
                        5902*   5915*   5922*   5928*   5934*   5940*   5946*   5953*   5959*   5964*   5974*   5980*   5986*
                        6022*   6035*   6042*   6048*   6054*   6060*   6066*   6073*   6079*   6084*   6094*   6100*   6106*
                        6142*   6155*   6162*   6168*   6174*   6180*   6186*   6193*   6199*   6204*   6214*   6220*   6226*
                        6262*   6280*   6287*   6293*   6299*   6305*   6311*   6318*   6324*   6329*   6335*   6341*   6347*
                        6387*   6405*   6412*   6418*   6424*   6430*   6436*   6443*   6449*   6454*   6460*   6466*   6472*
                        6512*   6530*   6537*   6543*   6549*   6555*   6561*   6568*   6574*   6579*   6585*   6591*   6597*
                        6637*   6655*   6662*   6668*   6674*   6680*   6686*   6693*   6699*   6704*   6710*   6716*   6722*
                        6757*   6765*   6770*   6775*   6780*   6785*   6791*   6796*   6801*   6806*   6815*   6820*   6826*
                        6842*   6850*   6855*   6860*   6865*   6870*   6876*   6881*   6886*   6891*   6900*   6905*   6911*
                        7383*   7388*   7393*   7398*   7403*   7412*   7417*   7422*   7427*   7436*   7441*   7447*   7455*
                        7508*   7522*   7527*   7532*   7537*   7542*   7551*   7556*   7561*   7566*   7575*   7580*   7586*
                        7592*   7624*   7638*   7643*   7648*   7653*   7658*   7667*   7672*   7677*   7682*   7691*   7696*
                        7702*   7708*   7740*   7754*   7759*   7764*   7769*   7774*   7783*   7788*   7793*   7798*   7807*
                        7812*   7818*   7824*   7856*   7870*   7875*   7880*   7885*   7890*   7899*   7904*   7909*   7914*
                        7923*   7928*   7934*   7940*
EM30    065373          1414    9651#
EM31    065444          1419    9658#
```

```
EM32    065515          1424  9665#
EM33    065565          7508  7624  7740  7856  9672#
EM34    065632          1429  1434  9679#
EM35    065675          1439  1444  9685#
EM36    065733          1450  1456  9691#
EM37    066007          1461  1466  9699#
EM38    066037          1471  1477  1482  9704#
EM39    066067          1487  9709#
EM4     063465          1965  2296  2420  2544  2668  9477#
EM4N    001330          1379# 2032# 2038# 2058# 2059# 2257# 2258# 2388# 2389# 2512# 2513# 2636# 2637#
                        2760# 2761# 2890# 2891# 3022# 3023# 3154# 3155# 3286# 3287# 3972# 3973# 4094#
                        4095# 4216# 4217# 4338# 4339# 4460# 4461# 4582# 4583# 4704# 4705# 4826# 4827#
                        4949# 4950# 5074# 5075# 5199# 5200# 5324# 5325# 5357# 5358# 5366# 5367# 5874#
                        5875# 5994# 5995# 6114# 6115# 6234# 6235# 6359# 6360# 6484# 6485# 6609# 6610#
                        6734# 6735#
EM40    066137          1497  9716#
EM41    066205          1502  9723#
EM42    066267          1508  9732#
EM43    066374          1514  9744#
EM44    066460          1520  9753#
EM5     063521          2032  9482#
EM6     063600          2058  9490#
EM7     063663          2077  3402  3521  3640  3759  9500#
EM8     063722          2180  3880  4002  4124  4246  9506#
EM9     063757          3312  9511#
ERRCNT  002006          1575# 1643# 8629# 8632
ERRVEC= 000004          1049# 1624  1625# 1636# 1744# 1745# 1756# 1757# 8278# 8279# 8291# 8292# 8329
                        8330# 8332# 8335#
E.ASOF  001756          1557#
E.BA    001744          1552#
E.CS1   001740          1550# 7332# 7337  7344# 7346  7459# 7461  7472# 7479  7972# 7974  7984  7997
                        8016  8036# 8041  8067# 8071  8077# 8079  9221  9223  9226
E.CS2   001750          1554# 7331# 7333  7345# 7350  7460# 7465  7473# 7474  7973# 7977  7987  7996#
                        8000  8005# 8019  8029# 8035# 8037  8066# 8068  8078# 8083  9221  9223  9226
E.DA    001746          1553#
E.DB    001762          1559#
E.DCYL  001760          1558#
E.DS    001752          1555#
E.ECPS  001772          1563#
E.ECPT  001774          1564#
E.ER    001754          1556#
E.MR1   001764          1560#
E.MR2   001766          1561#
E.MR3   001770          1562#
E.SPAR  001776          1565#
E.WC    001742          1551#
FMTE  = 000020          1143#
GNS   = ****** U        1205  1663  8249  8256  9192  9193  9194  9195  9196  9198  9200  9201  9202
                        9203  9204  9205  9206
GO    = 000001          1106# 2897  3029  3161  3293
GTSWR = 104406          1658  9198#
HT    = 000011           959# 8681  8722
HVRC  = 000400          1147#
IDAE  = 002000          1149#
IE    = 000100          1107# 6954  7194  7217  7266  7291  7293  8110  8160  8203
ILF   = 000001          1139#
```

```
INTR   = 000300  1102#
IOTVEC= 000020  1054#  1608*  1609*
IR     = 000100  1126#  1804   1806   1852   1854   1897   1899   1945   1947   1991   1993   2112   2114
                 2213   2215   2315   2338   2340   2439   2462   2464   2563   2586   2588   2687   2710
                 2712   2835   2837   2967   2969   3099   3101   3231   3233   3343   3345   3419   3442
                 3444   3538   3561   3563   3657   3680   3682   3776   3799   3801   3899   3922   3924
                 4021   4044   4046   4143   4166   4168   4265   4288   4290   4387   4410   4412   4509
                 4532   4534   4631   4654   4656   4753   4776   4778   4868   4877   4946   4993   5002
                 5071   5118   5127   5196   5243   5252   5321   5386   5388   5470   5472   5555   5557
                 5639   5641   5723   5725   5801   5830   5832   5921   5950   5952   6041   6070   6072
                 6161   6190   6192   6286   6315   6317   6411   6440   6442   6536   6565   6567   6661
                 6690   6692   6788   6790   6873   6875   7331   7345   7409   7411   7460   7473   7548
                 7550   7664   7666   7780   7782   7896   7898   7973   8005   8029   8035   8078
LF     = 000012   960#  8716   8722
MCLK   = 000400  1180#
MDS    : 001000  1129#  4867   4992   5117   5242
MEMBAS: 172100  1062#  8280
MEMERR  055354  8284   8297#
MEMVEC: 000114  1061#  8284#  8285*
MERD   = 001000  1181#
MESMSK= 000017  1174#
MEWD   = 002000  1182#  1825   1831   1918   1924   2012   2133   2234   2362   2486   2610   2734   2859
                 2991   3123   3255   3364   3456   3585   3704   3823   3946   4068   4190   4312   4434
                 4556   4678   4800   4923   5048   5173   5298   5407   5491   5576   5660   5744   5848
                 5968   6088   6208   6273   6352   6398   6477   6523   6602   6648   6727   6809   6894
                 7430   7569   7685   7801   7917
MIND   = 000200  1179#
MSP    = 000100  1178#
NED    = 010000  1132#  4867   4992   5117   5242
NEM    = 004000  1131#  4867   4992   5117   5242
NEWPAS  003040  1717   1726#  8272
NXF    = 000004  1141#
OFFSET= 000015  1095#
OFST   = 000004  1160#
OPI    = 020000  1152#
OPR001  062052  1667   9327#
OPR002  062101  1670   1681   1698   9331#
OPR003  062107  1677   9333#
OPR004  062137  1689   9338#
OR     = 000200  1127#  4867   4992   5117   5242   7409   7411   7548   7550   7664   7666   7780   7782
                 7896   7898   7973   7996   8005   8066
PACK   = 000003  1090#
PARM    002030  1212   1586#
PAR.EN= 000001  1063#  8283
PAT    = 000020  1176#
PCA    = 004000  1183#  6272   6397   6522   6647
PCD    = 010000  1184#  6272   6397   6522   6647
PGE    = 002000  1130#  4867   4992   5117   5242
PIP    = 020000  1168#
PIRQ   = 177772   966#
PIRQVE= 000240  1060#
PREREG  002014  1578#  2031*  2056*  2259*  2382*  2506*  2630*  2754*  2884*  3016*  3148*  3280*  3486*
                3605*  3724*  3843*  3966*  4088*  4210*  4332*  4454*  4576*  4698*  4820*  4943*  5068*
                5193*  5318*  5355*  5364*  5868*  5988*  6108*  6228*  6349*  6474*  6599*  6724*  9215
PRIOR   002020  1580#  7190*  7207*  7208*  7209*  7210*  7211*  7212*  7219*  9220
PRO    = 000000   983#
```

```
PR1    = 000040            984*
PR2    = 000100            985*
PR3    = 000140            986*
PR4    = 000200            987*
PR5    = 000240            988#  1570
PR6    = 000300            989*
PR7    = 000340            990#  1595  1728  1745  6941  6956  7136  7196  7227  7260  7268  7296  7303
                          8104  8113  8128  8159  8194  8205  8279  8285  9138  9151
PS     = 177776            963#  964
PSW    = 177776            964*
PWRVEC= 000024            1055#  1614*  1615*  9137*  9138*  9150*  9151*
RDCHR  = 104410           8980  9201#
RDDATA= 000021            1097*
RDGATE= 100000            1187#  6272  6397  6522  6647
RDHEAD= 000025            1099*
RDLIN  = 104411           9052  9202#
RDOCT  = 104412           1671  1682  1699  9203#
RDY    = 000200           1108#  1783  1785  1846  1848  1876  1878  1939  1941  1975  1977  2091  2093
                          2192  2194  2313  2437  2561  2685  2804  2887  2936  3019  3068  3151  3200
                          3283  3317  3319  3417  3536  3655  3774  3897  4019  4141  4263  4385  4507
                          4629  4751  4875  5000  5125  5250  5361  5365  5444  5446  5529  5531  5613
                          5615  5697  5699  5799  5919  6039  6159  6284  6409  6534  6659  6762  6764
                          6847  6849  6954  7194  7217  7291  7332  7344  7380  7382  7459  7472  7519
                          7521  7635  7637  7751  7753  7867  7869  7972  8036  8067  8077
RECAL  = 000013           1094*
RESREG= 104414            8628  9205*
RESTRT  002040            1210  1589*
RESVEC= 000010            1050*
RKASOF= 000016            1077#  1798  1891  1985  2106  2207  2331  2455  2579  2703  2828  2960  3092
                          3224  3337  3435  3554  3673  3792  3915  4037  4159  4281  4375*  4376  4454
                          4456  4497*  4498  4576  4578  4619*  4620  4698  4700  4741*  4742  4820  4822
                          4899  5024  5149  5274  5380  5464  5549  5633  5717  5823  5943  6063  6183
                          6308  6433  6558  6683  6782  6867  7400  7539  7655  7771  7887
RKBA   = 000004           1072#  1788  1881  1967*  1968  2035  2052*  205  2096  2197  2303*  2304  2382
                          2384  2427*  2428  2506  2508  2551*  2552  2630  2632  2675*  2676  2754  2756
                          2810  2942  3074  3206  3322  3423  3542  3661  3780  3903  4025  4147  4269
                          4391  4513  4635  4757  4881  5006  5131  5256  5350*  5353  5449  5534  5618
                          5702  5805  5925  6045  6165  6290  6415  6540  6665  6767  6852  7385  7524
                          7640  7756  7872
RKCS1  = 000000           1070#  1782  1845  1874*  1875  1938  1966*  1974  2051*  2053*  2082*  2090  2152*
                          2181*  2191  2253*  2297*  2312  2383*  2421*  2436  2507*  2545*  2560  2631*  2669*
                          2684  2755*  2794*  2800*  2801  2884  2885*  2886  2926*  2932*  2933  3016  3017*
                          3018  3058*  3064*  3065  3148  3149*  3150  3190*  3196*  3197  3280  3281*  3282
                          3313*  3316  3403*  3416  3487*  3522*  3535  3606*  3641*  3654  3725*  3760*  3773
                          3844*  3881*  3896  3967*  4003*  4018  4089*  4125*  4140  4211*  4247*  4262  4333*
                          4369*  4384  4455*  4491*  4506  4577*  4613*  4628  4699*  4735*  4750  4821*  4858*
                          4874  4944*  4983*  4999  5069*  5108*  5124  5194*  5233*  5249  5319*  5348*  5351*
                          5360  5440*  5443  5525*  5528  5609*  5612  5693*  5696  5783*  5798  5869*  5903*
                          5918  5989*  6023*  6038  6109*  6143*  6158  6229*  6263*  6283  6350*  6388*  6408
                          6475*  6513*  6533  6600*  6638*  6658  6725*  6758*  6761  6843*  6846  6945*  6954*
                          7194*  7217*  7257*  7266*  7290*  7291*  7292*  7293*  7327*  7330  7341*  7342  7375*
                          7379  7457  7470  7[illegible]3*  7515*  7518  7631*  7634  7747*  7750  7863*  7866  7963*
                          7970  7982  7994  8[illegible]3  8034  8045*  8058*  8065  8074*  8075  8101*  8110*  8135*
                          8154*  8156*  8160*  8191*  8203*  8221*
RKCS2  = 000010           1074#  1803  1851  1896  1944  1990  2111  2212  2337  2461  2585  2709  2834
                          2966  3098  3230  3342  3441  3560  3679  3798  3921  4043  4165  4287  4409
```

```
                    4531   4653   4775   4864*  4865   4943   4945   4989*  4990   5068   5070   5114*  5115
                    5193   5195   5239*  5240   5318   5320   5352*  5385   5469   5554   5638   5722   5829
                    5949   6069   6189   6314   6439   6564   6689   6787   6872   7329   7343   7408   7458
                    7471   7547   7663   7779   7895   7971   7983   7995   8014   8033   8063   8076
RKDA   = 000006      1073#  1793   1886   1980   2101   2183*  2185   2254   2325   2449   2573   2697   2822
                    2954   3086   3218   3332   3429   3548   3667   3786   3887*  3888   3966   3968   4009*
                    4010   4088   4090   4131*  4132   4210   4212   4253*  4254   4332   4334   4403   4525
                    4647   4769   4893   5018   5143   5268   5375   5459   5544   5628   5712   5817   5937
                    6057   6177   6302   6427   6552   6677   6777   6862   7395   7534   7650   7766   7882
RKDB   = 000024      1079#  7328   7378*  7449   7469   7517*  7588   7633*  7704   7749*  7820   7865*  7936
                    7966*  7981*  7993*  8006   8012   8032   8060*  8111   8155   8200
RKDCYL= 000020       1078#  1819   1912   2006   2127   2228   2355   2479   2603   2727   2852   2984   3116
                    3248   3358   3459   3578   3697   3816   3939   4061   4183   4305   4427   4549   4671
                    4793   4916   5041   5166   5291   5401   5485   5570   5654   5738   5789*  5790   5868
                    5870   5909*  5910   5988   5990   6029*  6030   6108   6110   6149*  6150   6228   6230
                    6332   6457   6582   6707   6803   6888   7424   7563   7679   7795   7911
RKDS   = 000012      1075#  1809   1902   1996   2117   2218   2344   2468   2592   2716   2841   2973   3105
                    3237   3348   3448   3567   3686   3805   3928   4050   4172   4294   4416   4538   4660
                    4782   4905   5030   5155   5280   5391   5442*  5475   5560   5644   5728   5836   5956
                    6076   6196   6321   6446   6571   6696   6793   6878   7414   7553   7669   7785   7901
RKECPS= 000030       1083#  1839   1932   2025   2146   2247   2376   2500   2624   2748   2873   3005   3137
                    3269   3377   3480   3599   3718   3837   3960   4082   4204   4326   4448   4570   4692
                    4814   4937   5062   5187   5312   5420   5504   5589   5673   5757   5862   5982   6102
                    6222   6343   6468   6593   6718   6822   6845*  6907   7443   7582   7698   7814   7930
RKECPT= 000032       1084#  1834   1927   2020   2141   2242   2371   2495   2619   2743   2868   3000   3132
                    3264   3372   3475   3594   3713   3832   3955   4077   4199   4321   4443   4565   4687
                    4809   4932   5057   5182   5307   5415   5499   5584   5668   5752   5857   5977   6097
                    6217   6338   6463   6588   6713   6760*  6817   6902   7438   7577   7693   7809   7925
RKER   = 000014      1076#  1814   1907   2001   2122   2223   2349   2473   2597   2721   2846   2978   3110
                    3242   3353   3453   3572   3691   3810   3933   4055   4177   4299   4421   4543   4665
                    4787   4910   5035   5160   5285   5396   5480   5527*  5565   5649   5733   5841   5961
                    6081   6201   6326   6451   6576   6701   6798   6883   7419   7558   7674   7790   7906
RKMR1  = 000026      1080#  1824   1917   2011   2132   2233   2361   2485   2609   2733   2858   2990   3122
                    3254   3363   3465   3584   3703   3822   3945   4067   4189   4311   4433   4555   4677
                    4799   4922   5047   5172   5297   5406   5490   5575   5659   5743   5847   5967   6087
                    6207   6269*  6270   6349   6351   6394*  6395   6474   6476   6519*  6520   6599   6601
                    6644*  6645   6724   6726   6808   6893   7429   7568   7684   7800   7916
RKMR2  = 000034      1081#  5611*
RKMR3  = 000036      1082#  5695*
RKPRI    002002     1570#  1715*  7182   7207   7213
RKSPAR= 000022       1085#  3315*
RKVEC    002000     1569#  1713*  1714*  7118   7180   7205*  7220*  7242   7258   7274   7294   7309   8102
                    8121*  8136   8157   8173   8192   8202*  8222   9218
RKWC   = 000002      1071#  2083*  2084   2153   2182*  2202   2302*  2319   2426*  2443   2550*  2567   2674*
                    2691   2799*  2816   2931*  2948   3063*  3080   3195*  3212   3314*  3327   3408*  3409
                    3486   3488   3527*  3528   3605   3607   3646*  3647   3724   3726   3765*  3766   3843
                    3845   3886*  3909   4008*  4031   4130*  4153   4252*  4275   4374*  4397   4496*  4519
                    4618*  4641   4740*  4763   4863*  4887   4988*  5012   5113*  5137   5238*  5262   5349*
                    5370   5441*  5454   5526*  5539   5610*  5623   5694*  5707   5788*  5811   5908*  5931
                    6028*  6051   6148*  6171   6268*  6296   6393*  6421   6518*  6546   6643*  6671   6759*
                    6772   6844*  6857   7376*  7390   7516*  7529   7632*  7645   7748*  7761   7864*  7877
RLS    = 000010      1123#
SAVFLG   002016     1579#  1716   1724*  6936*  6966*  7110*  7126*  7148*  7161*
SAVREG= 104413       8577   9204#
SAVSWR   002026     1583#  9136*  9155
SAVVEC   067754     1718   6930   6960   7104   7120   7142   7155   9884#
```

```
SCLR   = 000040            1125*  4956   5081   5206   5331   5352
SCOP1  = 104415            2161   2391   2515   2639   2763   2893   3025   3157   3289   3496   3615   3734   3853
                           3975   4097   4219   4341   4463   4585   4707   4829   4952   5077   5202   5327   5877
                           5997   6117   6237   6362   6487   6612   6737   7594   7710   7826   7942   9206*
SCOP1$  056124             8449*  9206
SEEK   = 000017            1096*
SELDRV= 000001             1089*
SILCNT  002022             1581*  7965*  7980*  7990   7992*  8003*  8025*  8026   9223   9226
SILO    070754             7964   8004   9885*
SKI    = 000002            1140*
SPACE2  062161             8611   8623   9342*
SPAR   = 020000            1115*  2803   2935   3067   3199
SPDLSS= 000020             1162*
SRTFLG  002004             1571*  1586*  1589*  1592*  1665
SRTSPL= 000011             1093*
STACK  = 001100             954*  1594   1606   8305   8639   9152
START   002050             1203   1592*
START1  002054             1587   1590   1593*
STKLMT= 177774              965*
SVAL   = 100000            1170*
SWR     001140             1273*  1604   1626*  1628   1634*  1641*  1656   8301   8324   8338   8341   8356   8363
                           8449   8531   8538   8550   8554   8630   8876   8913*  9136   9155*
SWREG   000176             1207*  1634   1656   8876   8889
SW0    = 000001            1018*
SW00   = 000001            1008*  1018
SW01   = 000002            1007*  1017
SW02   = 000004            1006*  1016
SW03   = 000010            1005*  1015
SW04   = 000020            1004*  1014
SW05   = 000040            1003*  1013
SW06   = 000100            1002*  1012
SW07   = 000200            1001*  1011
SW08   = 000400            1000*  1010
SW09   = 001000             999*  1009
SW1    = 000002            1017*
SW10   = 002000             998*
SW11   = 004000             997*
SW12   = 010000             996*  8630
SW13   = 020000             995*
SW14   = 040000             994*
SW15   = 100000             993*
SW2    = 000004            1016*
SW3    = 000010            1015*
SW4    = 000020            1014*
SW5    = 000040            1013*
SW6    = 000100            1012*
SW7    = 000200            1011*
SW8    = 000400            1010*
SW9    = 001000            1009*  8301   8449
S.CLR  = 000400            1195*
S.FMT  = 001000            1196*
S.PACK= 004000             1198*
S.RECL= 000040             1192*
S.RTC  = 000200            1194*
S.SEEK= 000020             1191*
S.STSP= 000100             1193*
```

```
S.UNLD= 002000          1197#
TBITVE= 000014          1051#
TKVEC = 000060          1058#
TPVEC = 000064          1059#
TRAPPC  002024          1582#   8298*   9209
TRAPVE= 000034          1057#   1612*   1613*
TRTVEC= 000014          1052#
TST1    003062          1729    1741#   8383
TST10   006702          2255    2292#   8390
TST11   007530          2395    2416#   8391
TST12   010356          2519    2540#   8392
TST13   011202          2643    2664#   8393
TST14   012026          2767    2788#   8394
TST15   012714          2899    2920#   8395
TST16   013604          3031    3052#   8396
TST17   014470          3163    3184#   8397
TST2    003154          1776#   8384
TST20   015354          3295    3308#   8398
TST21   016062          3379    3398#   8399
TST22   016700          3500    3517#   8400
TST23   017516          3619    3636#   8401
TST24   020332          3738    3755#   8402
TST25   021146          3857    3876#   8403
TST26   021774          3979    3998#   8404
TST27   022622          4101    4120#   8405
TST3    003714          1853    1870#   8385
TST30   023446          4223    4242#   8406
TST31   024272          4345    4364#   8407
TST32   025120          4467    4486#   8408
TST33   025746          4589    4608#   8409
TST34   026572          4711    4730#   8410
TST35   027416          4833    4852#   8411
TST36   030266          4958    4977#   8412
TST37   031136          5083    5102#   8413
TST4    004454          1946    1961#   8386
TST40   032002          5208    5227#   8414
TST41   032646          5333    5345#   8415
TST42   033412          5422    5435#   8416
TST43   034120          5506    5520#   8417
TST44   034626          5591    5604#   8418
TST45   035334          5675    5688#   8419
TST46   036042          5759    5778#   8420
TST47   036670          5881    5898#   8421
TST5    005210          2036    2048#   8387
TST50   037516          6001    6018#   8422
TST51   040342          6121    6138#   8423
TST52   041166          6241    6258#   8424
TST53   042032          6366    6383#   8425
TST54   042676          6491    6508#   8426
TST55   043540          6616    6633#   8427
TST56   044402          6741    6753#   8428
TST57   045110          6824    6838#   8429
TST6    005306          2055    2073#   8388
TST60   045616          6909    6928#   8430
TST61   046672          7112    7128    7149    7178#   8431
TST62   047156          7255#   8432
```

```
TST63   047264            7288#   8433
TST64   047406            7324#   8434
TST65   047572            7351    7373#   8435
TST66   050474            7506#   8436
TST67   051274            7598    7622#   8437
TST7    006116            2163    2176#   8389
TST70   052074            7714    7738#   8438
TST71   052672            7830    7854#   8439
TST72   053470            7946    7960#   8440
TST73   054206            8055#   8441
TST74   054404            8084    8098#   8442
TST75   054572            8151#   8443
TST76   054706            8188#   8444
TYPDS = 104405            8253    8260    9196#
TYPE  = 104401            1650    1667    1670    1677    1681    1689    1698    8247    8254    8261    8533    8541    8587
                          8588    8592    8593    8600    8611    8613    8623    8624    8626    8634    8686    8782    8857
                          8887    8888    8891    8904    8915    8934    8987    8993    8998    9002    9007    9008    9010
                          9013    9017    9081    9083    9153    9192#
TYPERR  056616            8540    8577#
TYPOC = 104402            1669    1680    1697    8608    8890    9193#
TYPON = 104404            9195#
TYPOS = 104403            9194#
T.ASOF  001716            1538#
T.BA    001704            1533#
T.CS1   001700            1531#   7330#   7337    7342#   7346    7457#   7461    7470#   7479    7970#   7974    7982#   7984
                          7994#   7997    8013#   8016    8034#   8041    8065#   8071    8075#   8079    9221    9223    9226
T.CS2   001710            1535#   7329#   7333    7343#   7350    7458#   7465    7471#   7474    7971#   7977    7983#   7987
                          7995#   8000    8014#   8019    8033#   8037    8063#   8068    8076#   8083    9221    9223    9226
T.DA    001706            1534#
T.DB    001722            1540#
T.DCYL  001720            1539#
T.DS    001712            1536#
T.ECPS  001732            1544#
T.ECPT  001734            1545#
T.ER    001714            1537#
T.MR1   001724            1541#
T.MR2   001726            1542#
T.MR3   001730            1543#
T.SPAR  001736            1546#
T.WC    001702            1532#
UFE   = 000400            1128#   4867    4992    5117    5242
UNLOAD= 000007            1092#
UNS   = 040000            1153#
UPE   = 020000            1133#   4867    4992    5117    5242
VV    = 000100            1164#
WCE   = 040000            1134#   4867    4992    5117    5242
WLE   = 004000            1150#
WRDATA= 000023            1098#
WRHEAD= 000027            1100#
WRL   = 004000            1167#
WRTCHK= 000031            1101#
WRTGAT= 040000            1186#   6272    6397    6522    6647
$APTHD  001000            1233    1239#
$ASTAT= ****** U          8489    8504
$ATYC   056174            8460    8462#
$ATY1   056150            8458#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    5584*  5589*  5590   5612*  5613   5618*  5623*  5628*  5633*  5638*  5639   5644*  5649*
                    5654*  5659*  5661   5664   5668*  5673*  5674   5696*  5697   5702*  5707*  5712*  5717*
                    5722*  5723   5728*  5733*  5738*  5743*  5745   5748   5752*  5757*  5758   5790*  5793
                    5798*  5799   5805*  5811*  5817*  5823*  5829*  5830   5836*  5841*  5847*  5849   5852
                    5857*  5862*  5864   5870*  5872   5910*  5913   5918*  5919   5925*  5931*  5937*  5943*
                    5949*  5950   5956*  5961*  5967*  5969   5972   5977*  5982*  5984   5990*  5992   6030*
                    6033   6038*  6039   6045*  6051*  6057*  6063*  6069*  6070   6076*  6081*  6087*  6089
                    6092   6097*  6102*  6104   6110*  6112   6150*  6153   6158*  6159   6165*  6171*  6177*
                    6183*  6189*  6190   6196*  6201*  6207*  6209   6212   6217*  6222*  6224   6230*  6232
                    6270*  6274   6278   6283*  6284   6290*  6296*  6302*  6308*  6314*  6315   6321*  6326*
                    6332*  6338*  6343*  6345   6351*  6353   6357   6395*  6399   6403   6408*  6409   6415*
                    6421*  6427*  6433*  6439*  6440   6446*  6451*  6457*  6463*  6468*  6470   6476*  6478
                    6462   6520*  6524   6528   6533*  6534   6540*  6546*  6552*  6558*  6564*  6565   6571*
                    6576*  6582*  6588*  6593*  6595   6601*  6603   6607   6645*  6649   6653   6658*  6659
                    6665*  6671*  6677*  6683*  6689*  6690   6696*  6701*  6707*  6713*  6718*  6720   6726*
                    6728   6732   6761*  6762   6767*  6772*  6777*  6782*  6787*  6788   6793*  6798*  6803*
                    6808*  6810   6813   6817*  6822*  6823   6846*  6847   6852*  6857*  6862*  6867*  6872*
                    6873   6878*  6883*  6888*  6893*  6895   6898   6902*  6907*  6908   7379*  7380   7385*
                    7390*  7395*  7400*  7408*  7409   7414*  7419*  7424*  7429*  7431   7434   7438*  7443*
                    7444   7449*  7450   7518*  7519   7524*  7529*  7534*  7539*  7547*  7548   7553*  7558*
                    7563*  7568*  7570   7573   7577*  7582*  7583   7588*  7589   7634*  7635   7640*  7645*
                    7650*  7655*  7663*  7664   7669*  7674*  7679*  7684*  7686   7689   7693*  7698*  7699
                    7704*  7705   7750*  7751   7756*  7761*  7766*  7771*  7779*  7780   7785*  7790*  7795*
                    7800*  7802   7805   7809*  7814*  7815   7820*  7821   7866*  7867   7872*  7877*  7882*
                    7887*  7895*  7896   7901*  7906*  7911*  7916*  7918   7921   7925*  7930*  7931   7936*
                    7937   8006*  8012*  8022   9211   9213   9215   9226

$BELL   001204      1293#  8533   8566
$CDW1   001274      1344#
$CDW2   001276      1345#
$CHARC  057356      8688*  8698*  8705   8714*  8719#
$CKSWR  060034      8876#  9200
$CMTAG  001100      1253#  1601   1602   1610   1616   1617   1618
$CM3    =  000000   1283#
$CM4    =  000010   1283#  1284#  1285#  1286#  1287#  1288#  1289#  1290#  1291#
$CNTLG  060707      8887   9028#
$CNTLU  060702      8904   9002   9027#
$CPUOP  001242      1316#
$CRLF   001211      1295#  8261   8541   8566   8587   8592   8600   8613   8626   8687   8722   8915   9007
                    9027   9086
$DBLK   060024      8823   8857   8865#
$DEVCT  001224      1307#
$DEVM   001272      1343#
$DOAGN  055254      8243   8264   8270#
$DTBL   060014      8826   8861#
$ENDAD  055244      1219   1648   8266#  8561
$ENDCT  055110      1616   8245#
$ENULL  055260      8273#
$ENV    001234      1312#  1654   8467   8491   8543   8666
$ENVM   001235      1313#  1639   8469   8668   8673
$EOP    055054      8235#  8640
$EOPCT  055102      1616*  8242#  8246   8638*
$ERFLG  001103      1256#  8313   8352   8354   8360*  8382   8451   8528*  8566
$ERMAX  001115      1262#  1619*  8354   8377*  8302
$ERROR  056416      1610   8526#
$ERRPC  001116      1263#  8535*  8536*  8537   8566   9210   9211   9213   9215   9217   9218   9220   9221
                    9223   9226
```

```
$ERRTB  001300  1362# 8584
$ERTTL  001112  1260# 8258  8262* 8534* 8566
$ESCAP  001202  1292# 1618* 8297* 8300* 8376* 8557  8559  8566
$ETABL  001234  1311#
$ETEND  001300  1245  1346#
$FATAL  001216  1304# 8495*
$FFLG   056414  8458# 8461* 8489  8498* 8506#
$FILLC  001156  1281# 8691  8722
$FILLS  001155  1280# 8722
$GDADR  001120  1264#
$GDDAT  001124  1266# 1785* 1790* 1795* 1800* 1806* 1811* 1816* 1821* 1825* 1828* 1829  1831*
                1836* 1842* 1848* 1854* 1878* 1883* 1888* 1893* 1899* 1904* 1909* 1914* 1918*
                1921* 1922  1924* 1929* 1935* 1941* 1947* 1971* 1977* 1982* 1987* 1993* 1998*
                2003* 2008* 2012* 2015* 2016  2022* 2028* 2037* 2057* 2087* 2093* 2098* 2103*
                2108* 2114* 2119* 2124* 2129* 2133* 2136* 2137  2143* 2149* 2157* 2188* 2194*
                2199* 2204* 2209* 2215* 2220* 2225* 2230* 2234* 2237* 2238  2244* 2250* 2256*
                2305* 2306* 2307  2315* 2321* 2327* 2333* 2340* 2346* 2351* 2357* 2362* 2365*
                2366  2373* 2377* 2378  2385* 2386  2429* 2430* 2431  2439* 2445* 2451* 2457*
                2464* 2470* 2475* 2481* 2486* 2489* 2490  2497* 2501* 2502  2509* 2510  2553*
                2554* 2555  2563* 2569* 2575* 2581* 2588* 2594* 2599* 2605* 2610* 2613* 2614
                2621* 2625* 2626  2633* 2634  2677* 2678* 2679  2687* 2693* 2699* 2705* 2712*
                2718* 2723* 2729* 2734* 2737* 2738  2745* 2749* 2750  2757* 2758  2802* 2803*
                2804* 2805  2812* 2818* 2824* 2830* 2837* 2843* 2848* 2854* 2859* 2862* 2863
                2870* 2876* 2879* 2880  2887* 2888  2934* 2935* 2936* 2937  2944* 2950* 2956*
                2962* 2969* 2975* 2980* 2986* 2991* 2994* 2995  3002* 3008* 3011* 3012  3019*
                3020  3066* 3067* 3068* 3069  3076* 3082* 3088* 3094* 3101* 3107* 3112* 3118*
                3123* 3126* 3127  3134* 3140* 3143* 3144  3151* 3152  3198* 3199* 3200* 3201
                3208* 3214* 3220* 3226* 3233* 3239* 3244* 3250* 3255* 3258* 3259  3266* 3272*
                3275* 3276  3283* 3284  3319* 3324* 3329* 3334* 3339* 3345* 3350* 3355* 3360*
                3364* 3367* 3368  3374* 3380* 3410* 3411  3419* 3425* 3431* 3437* 3444* 3450*
                3455* 3461* 3466* 3469* 3470  3477* 3481* 3482  3492* 3529* 3530  3538* 3544*
                3550* 3556* 3563* 3569* 3574* 3580* 3585* 3588* 3589  3596* 3600* 3601  3611*
                3648* 3649  3657* 3663* 3669* 3675* 3682* 3688* 3693* 3699* 3704* 3707* 3708
                3715* 3719* 3720  3730* 3767* 3768  3776* 3782* 3788* 3794* 3801* 3807* 3812*
                3818* 3823* 3826* 3827  3834* 3838* 3839  3849* 3889* 3890* 3891  3899* 3905*
                3911* 3917* 3924* 3930* 3935* 3941* 3946* 3949* 3950  3957* 3961* 3962  3969*
                3970  4011* 4012* 4013  4021* 4027* 4033* 4039* 4046* 4052* 4057* 4063* 4068*
                4071* 4072  4079* 4083* 4084  4091* 4092  4133* 4134* 4135  4143* 4149* 4155*
                4161* 4168* 4174* 4179* 4185* 4190* 4193* 4194  4201* 4205* 4206  4213* 4214
                4255* 4256* 4257  4265* 4271* 4277* 4283* 4290* 4296* 4301* 4307* 4312* 4315*
                4316  4323* 4327* 4328  4335* 4336  4377* 4378* 4379  4387* 4393* 4399* 4405*
                4412* 4418* 4423* 4429* 4434* 4437* 4438  4445* 4449* 4450  4457* 4458  4499*
                4500* 4501  4509* 4515* 4521* 4527* 4534* 4540* 4545* 4551* 4556* 4559* 4560
                4567* 4571* 4572  4579* 4580  4621* 4622* 4623  4631* 4637* 4643* 4649* 4656*
                4662* 4667* 4673* 4678* 4681* 4682  4689* 4693* 4694  4701* 4702  4743* 4744*
                4745  4753* 4759* 4765* 4771* 4778* 4784* 4789* 4795* 4800* 4803* 4804  4811*
                4815* 4816  4823* 4824  4866* 4867* 4868* 4869  4877* 4883* 4889* 4895* 4901*
                4907* 4912* 4918* 4923* 4926* 4927  4934* 4938* 4939  4946* 4947  4991* 4992*
                4993* 4994  5002* 5008* 5014* 5020* 5026* 5032* 5037* 5043* 5048* 5051* 5052
                5059* 5063* 5064  5071* 5072  5116* 5117* 5118* 5119  5127* 5133* 5139* 5145*
                5151* 5157* 5162* 5168* 5173* 5176* 5177  5184* 5188* 5189  5196* 5197  5241*
                5242* 5243* 5244  5252* 5258* 5264* 5270* 5276* 5282* 5287* 5293* 5298* 5301*
                5302  5309* 5313* 5314  5321* 5322  5356* 5365* 5372* 5377* 5382* 5388* 5393*
                5403* 5407* 5410* 5411  5417* 5423* 5446* 5451* 5456* 5461* 5466* 5472* 5477*
                5482* 5487* 5491* 5494* 5495  5501* 5507* 5531* 5536* 5541* 5546* 5551* 5557*
                5562* 5567* 5572* 5576* 5579* 5580  5586* 5592* 5615* 5620* 5625* 5630* 5635*
```

```
                    5641*  5646*  5651*  5656*  5660*  5663*  5664   5670*  5676*  5699*  5704*  5709*  5714*
                    5719*  5725*  5730*  5735*  5740*  5744*  5747*  5748   5754*  5760*  5791*  5792*  5793
                    5801*  5807*  5813*  5819*  5825*  5832*  5838*  5843*  5848*  5851*  5852   5859*  5863*
                    5864   5871*  5872   5911*  5912*  5913   5921*  5927*  5933*  5939*  5945*  5952*  5958*
                    5963*  5968*  5971*  5972   5979*  5983*  5984   5991*  5992   6031*  6032*  6033   6041*
                    6047*  6053*  6059*  6065*  6072*  6078*  6083*  6088*  6091*  6092   6099*  6103*  6104
                    6111*  6112   6151*  6152*  6153   6161*  6167*  6173*  6179*  6185*  6192*  6198*  6203*
                    6208*  6211*  6212   6219*  6223*  6224   6231*  6232   6271*  6272*  6273*  6276*  6278
                    6286*  6292*  6298*  6304*  6310*  6317*  6323*  6328*  6334*  6340*  6344*  6345   6352*
                    6355*  6357   6396*  6397*  6398*  6401*  6403   6411*  6417*  6423*  6429*  6435*  6442*
                    6448*  6453*  6459*  6465*  6469*  6470   6477*  6480*  6482   6521*  6522*  6523*  6526*
                    6528   6536*  6542*  6548*  6554*  6560*  6567*  6573*  6578*  6584*  6590*  6594*  6595
                    6602*  6605*  6607   6646*  6647*  6648*  6651*  6653   6661*  6667*  6673*  6679*  6685*
                    6692*  6698*  6703*  6709*  6715*  6719*  6720   6727*  6730*  6732   6764*  6769*  6774*
                    6779*  6784*  6790*  6795*  6800*  6805*  6809*  6812*  6813   6819*  6825*  6849*  6854*
                    6859*  6864*  6869*  6875*  6880*  6885*  6890*  6894*  6897*  6898   6904*  6910*  7382*
                    7387*  7392*  7397*  7402*  7411*  7416*  7421*  7426*  7430*  7433*  7434   7440*  7446*
                    7454*  7521*  7526*  7531*  7536*  7541*  7550*  7555*  7560*  7565*  7569*  7572*  7573
                    7579*  7585*  7591*  7637*  7642*  7647*  7652*  7657*  7666*  7671*  7676*  7681*  7685*
                    7688*  7689   7695*  7701*  7707*  7753*  7758*  7763*  7768*  7773*  7782*  7787*  7792*
                    7797*  7801*  7804*  7805   7811*  7817*  7823*  7869*  7874*  7879*  7884*  7889*  7898*
                    7903*  7908*  7913*  7917*  7920*  7921   7927*  7933*  7939*  8015*  8022   9211   9213
                    9215   9226
$GET42  055234      8263#
$GTSWR  060104      8888#  9198
$HD     = 000000     939
$HIBTS  001000      1240#
$HIOCT  061074      9074*  9085#
$ICNT   001104      1257#  8367*  8368   8370*  8381
$INTAG  001135      1271#  8916   9033
$ITEMB  001114      1261#  8537*  8545   8566   8578
$LF     001212      1296#  8566   8722   9017   9027   9086
$LFLG   056413      8499*  8505#
$LPADR  001106      1258#  1620*  8349*  8358*  8374*  8379   8381   9156
$LPERR  001110      1259#  1621*  2078*  2298*  2422*  2546*  2670*  2795*  2927*  3059*  3191*  3404*  3523*
                    3642*  3761*  3882*  4004*  4126*  4248*  4370*  4492*  4614*  4736*  4859*  4984*  5109*
                    5234*  5784*  5904*  6024*  6144*  6264*  6389*  6514*  6639*  7511*  7627*  7743*  7859*
                    8306   8358   8375*  8381   8453   8556
$MADR1  001246      1329#
$MADR2  001252      1333#
$MADR3  001256      1336#
$MADR4  001262      1339#
$MAIL   001214      1241   1245   1302#  1638   1654   8373   8543   8666
$MAMS1  001244      1323#
$MAMS2  001250      1331#
$MAMS3  001254      1334#
$MAMS4  001260      1337#
$MBADR  001002      1241#
$MFLG   056412      8459*  8465   8500*  8504#
$MNEW   060725      8891   9031#
$MSGAD  001230      1309#  8475*  8478
$MSGLG  001232      1310#  8480*
$MSGTY  001214      1303#  8473   8481*  8493   8497*
$MSWR   060714      8888   9029#
$MTYP1  001245      1324#
$MTYP2  001251      1332#
```

```
$MTYP3  001255          1335#
$MTYP4  001261          1338#
$MXCNT  055726          8371    8381#
$NULL   001154          1279#   8693    8722
$NWTST= 000001          1731#   1733    1762#   1764    1858#   1860    1951#   1953    2041#   2043    2062#   2064    2167#
                        2169    2274#   2276    2398#   2400    2522#   2524    2646#   2648    2770#   2772    2902#   2904
                        3034#   3036    3166#   3168    3258#   3300    3384#   3386    3503#   3505    3622#   3624    3741#
                        3743    3860#   3862    3982#   3984    4104#   4106    4226#   4228    4348#   4350    4470#   4472
                        4592#   4594    4714#   4716    4836#   4838    4961#   4963    5086#   5088    5211#   5213    5336#
                        5338    5426#   5428    5511#   5513    5596#   5598    5680#   5682    5764#   5766    5884#   5886
                        6004#   6006    6124#   6126    6244#   6246    6369#   6371    6494#   6496    6619#   6621    6744#
                        6746    6829#   6831    6917#   6919    7166#   7168    7247#   7249    7279#   7281    7317#   7319
                        7355#   7357    7485#   7487    7601#   7603    7717#   7719    7833#   7835    7949#   7951    8047#
                        8049    8088#   8090    8141#   8143    8178#   8180
$OCNT   057604          8754#   8783#   8796#
$OMODE  057606          8749#   8753#   8758    8761#   8772#   8798#
$OVER   055712          8325    8350    8359    8369    8378#
$PASS   001222          1306#   1638#   8239#   8240#   8251    8273    8365    8382
$PASTM  001006          1243#
$POWER  061314          9153    9159#
$PWRCT  061310          9144#   9145#   9146#   9148#   9158#
$PWRDN  061172          1614    9136#   9150
$PWRUP  061220          9137    9144#
$QUES   001210          1294#   8566    8722    8934    9010    9027    9083    9086
$RDCHR  060316          8947#   9201
$RDDEC= ****** U        9204
$RDLIN  060436          8975#   9202
$RDOCT  060736          9047#   9203
$RDSZ = 000010          8968#
$RESRE  061134          9119#   9205
$RTNAD  055256          8272#
$R2A  = ****** U        9206
$SAVRE  061076          9103#   9204
$SCOPE  055416          1608    8322#
$SETUP= 000137          1586#   1607    1608    1610    1612    1614    1616    1617    1618    1620    1648    1651    8237
                        8323    8527    8553    8561    8871    9033
$STUP = 177777          1586#
$SVLAD  055656          8333    8372#
$SVPC = 000220          1217#   1222
$SWR  = 167400           929#    939     943     944     945     946     947     948     949     950    1291    1292    1293
                        1617    1618    1620    1621    1742    1777    1871    1962    2049    2074    2177    2293    2417
                        2541    2665    2789    2921    3053    3185    3309    3399    3518    3637    3756    3877    3999
                        4121    4243    4365    4487    4609    4731    4853    4978    5103    5228    5346    5436    5521
                        5605    5689    5779    5899    6019    6139    6259    6384    6509    6634    6754    6839    6929
                        7179    7256    7289    7325    7374    7507    7623    7739    7855    7961    8056    8099    8152
                        8189    8232    8238    8265    8271    8273    8314    8315    8316    8317    8318    8324    8336
                        8338    8339    8352    8353    8354    8361    8362    8363    8375    8378    8381    8518    8519
                        8520    8521    8522    8531    8538    8550    8554    8566
$SWREG  001236          1314#   1641
$S.RMK= 000000           950     951    8318    8319    8342
$SWOST  055730          8348    8382#
$TESTN  001220          1305#   8373*   9209    9210    9211    9213    9215    9217    9218    9220    9221    9223    9226
$TIMES  001200          1291#   1617*   1742*   1777*   1871*   1962*   2049*   2074*   2177*   2293*   2417*   2541*   2665*
                        2789*   2921*   3053*   3185*   3309*   3399*   3518*   3637*   3756*   3877*   3999*   4121*   4243*
                        4365*   4487*   4609*   4731*   4853*   4978*   5103*   5228*   5346*   5436*   5521*   5605*   5689*
                        5779*   5899*   6019*   6139*   6259*   6384*   6509*   6634*   6754*   6839*   6929*   7179*   7256*
```

```
                    7289*  7325*  7374*  7507*  7623*  7739*  7855*  7961*  8056*  8099*  8152*  8189*  8238*
                    8361*  8368   8371*  8381
$TKB    001146      1276*  8869   8880   8897   8951   8957
$TKS    001144      1275*  8869   8878   8894   8918*  8949   8955
$TMPO   001160      1283*  1672*  1674   1676   1683*  1685   1688   1700*  1702   1704   1706*  1707*  1708*
                    1709*  1710*  1712
$TMP1   001162      1284*
$TMP2   001164      1285*
$TMP3   001166      1286*
$TMP4   001170      1287*
$TMP5   001172      1288*
$TMP6   001174      1289*
$TMP7   001176      1290*
$TN   = 000077       929*   939   1731   1742*  1762   1777*  1853   1858   1871*  1946   1951   1962*  2036
                    2041   2049*  2055   2062   2074*  2163   2167   2177*  2255   2274   2293*  2395   2398
                    2417*  2519   2522   2541*  2643   2646   2665*  2767   2770   2789*  2899   2902   2921*
                    3031   3034   3053*  3163   3166   3185*  3295   3298   3309*  3379   3384   3399*  3500
                    3503   3518*  3619   3622   3637*  3738   3741   3756*  3857   3860   3877*  3979   3982
                    3999*  4101   4104   4121*  4223   4226   4243*  4345   4348   4365*  4467   4470   4487*
                    4589   4592   4609*  4711   4714   4731*  4833   4836   4853*  4958   4961   4978*  5083
                    5086   5103*  5208   5211   5228*  5333   5336   5346*  5422   5426   5436*  5506   5511
                    5521*  5591   5596   5605*  5675   5680   5689*  5759   5764   5779*  5881   5884   5899*
                    6001   6004   6019*  6121   6124   6139*  6241   6244   6259*  6366   6369   6384*  6491
                    6494   6509*  6616   6619   6634*  6741   6744   6754*  6824   6829   6839*  6909   6917
                    6999*  7112   7128   7149   7166   7179*  7247   7256*  7279   7289*  7317   7325*  7351
                    7335   7374*  7485   7507*  7598   7601   7623*  7714   7717   7739*  7830   7833   7855*
                    7946   7949   7961*  8047   8056*  8084   8088   8099*  8141   8152*  8178   8189*  8343
                    8383
$TPB    001152      1278*  8711*  8722
$TPFLG  001157      1282*  8660   8722
$TPS    001150      1277*  8709   8722
$TRAP   061322      1612   9169*
$TRAP2  061344      9180*  9191
$TRP  = 000016      9184*  9193*  9194*  9195*  9196*  9197*  9198   9199*  9200   9201*  9202*  9203*  9204*
                    9205*  9206*  9207*
$TRPAD  061356      9174   9191*
$TSTM   001004      1242*
$TSTNM  001102      1255*  8237*  8313   8345*  8372*  8373   8378   8382   8530   8566
$TTYIN  060672      8977   8978   8990   9008   9022   9026*
$TYPBN= ****** U    9197
$TYPDS  057610      8811*  9196
$TYPE   057100      8486   8660*  9184   9192
$TYPEC  057312      8690   8697   8704   8709*  8710   8920
$TYPEX  057360      8715   8717   8720*
$TYPOC  057406      8752*  9193
$TYPON  057422      8751   8754*  9195
$TYPOS  057362      8747*  9194
$UNIT   001226      1308*
$UNITM  001010      1244*
$USWR   001240      1315*
$VECT1  001264      1340*  1678   1687*  1688*  1691   1711*  1712*  1713   1715
$VECT2  001266      1341*
$XTSTR  055430      8327*
$$GET4= 000000      8265*
$$SW08= 000077      8382*  8383   8384*  8385*  8386*  8387*  8388*  8389*  8390*  8391*  8392*  8393*  8394*
                    8395*  8396*  8397*  8398*  8399*  8400*  8401*  8402*  8403*  8404*  8405*  8406*  8407*
```

```
                    8408#   8409#   8410#   8411#   8412#   8413#   8414#   8415#   8416#   8417#   8418#   8419#   8420#
                    8421#   8422#   8423#   8424#   8425#   8426#   8427#   8428#   8429#   8430#   8431#   8432#   8433#
                    8434#   8435#   8436#   8437#   8438#   8439#   8440#   8441#   8442#   8443#   8444#   8445#
$OFILL  057605      8748#   8752#   8762    8797#
$40CAT= ****** U    8324    8540
.       = 071160    1201#   1205#   1211#   1217    1218#   1220#   1222#   1223#   1229    1230#   1232#   1234#   1252#
                    1297    1605    1620    1621    8250#   8273    8274#   8381    8382    8507#   8566    8722    8865#
                    8869    9026#   9027    9033    9086    9140    9884#
.$ASTA= ****** U    8459    8462
.$X     = 001000    1229#   1234
```

CLRPSW 1246# 6947 7131 7221 7261 7297 8105 8122 8161 8195
CLRREG 1246# 1778 1872
COMMEN 1061#
ENDCOM 1061#
ERROR 955# 1755 1787 1792 1797 1802 1808 1813 1818 1823 1833 1838 1844 1850 1856
1880 1885 1890 1895 1901 1906 1911 1916 1926 1931 1937 1943 1949 1973 1979
1984 1989 1995 2000 2005 2010 2019 2024 2030 2039 2060 2089 2095 2100 2105
2110 2116 2121 2126 2131 2140 2145 2151 2160 2190 2196 2201 2206 2211 2217
2222 2227 2232 2241 2246 2252 2260 2310 2317 2323 2329 2335 2342 2348 2353
2359 2369 2375 2381 2390 2434 2441 2447 2453 2459 2466 2472 2477 2483 2493
2499 2505 2514 2558 2565 2571 2577 2583 2590 2596 2601 2607 2617 2623 2629
2638 2682 2689 2695 2701 2707 2714 2720 2725 2731 2741 2747 2753 2762 2808
2814 2820 2826 2832 2839 2845 2850 2856 2866 2872 2883 2892 2940 2946 2952
2958 2964 2971 2977 2982 2988 2998 3004 3015 3024 3072 3078 3084 3090 3096
3103 3109 3114 3120 3130 3136 3147 3156 3204 3210 3216 3222 3228 3235 3241
3246 3252 3262 3268 3279 3288 3321 3326 3331 3336 3341 3347 3352 3357 3362
3371 3376 3382 3414 3421 3427 3433 3439 3446 3452 3457 3463 3473 3479 3485
3495 3533 3540 3546 3552 3558 3565 3571 3576 3582 3592 3598 3604 3614 3652
3659 3665 3671 3677 3684 3690 3695 3701 3711 3717 3723 3733 3771 3778 3784
3790 3796 3803 3809 3814 3820 3830 3836 3842 3852 3894 3901 3907 3913 3919
3926 3932 3937 3943 3953 3959 3965 3974 4016 4023 4029 4035 4041 4048 4054
4059 4065 4075 4081 4087 4096 4138 4145 4151 4157 4163 4170 4176 4181 4187
4197 4203 4209 4218 4260 4267 4273 4279 4285 4292 4298 4303 4309 4319 4325
4331 4340 4382 4389 4395 4401 4407 4414 4420 4425 4431 4441 4447 4453 4462
4504 4511 4517 4523 4529 4536 4542 4547 4553 4563 4569 4575 4584 4626 4633
4639 4645 4651 4658 4664 4669 4675 4685 4691 4697 4706 4748 4755 4761 4767
4773 4780 4786 4791 4797 4807 4813 4819 4828 4872 4879 4885 4891 4897 4903
4909 4914 4920 4930 4936 4942 4951 4997 5004 5010 5016 5022 5028 5034 5039
5045 5055 5061 5067 5076 5122 5129 5135 5141 5147 5153 5159 5164 5170 5180
5186 5192 5201 5247 5254 5260 5266 5272 5278 5284 5289 5295 5305 5311 5317
5326 5359 5368 5374 5379 5384 5390 5395 5400 5405 5414 5419 5425 5448 5453
5458 5463 5468 5474 5479 5484 5489 5498 5503 5509 5533 5538 5543 5548 5553
5559 5564 5569 5574 5583 5588 5594 5617 5622 5627 5632 5637 5643 5648 5653
5658 5667 5672 5678 5701 5706 5711 5716 5721 5727 5732 5737 5742 5751 5756
5762 5796 5803 5809 5815 5821 5827 5834 5840 5845 5855 5861 5867 5876 5916
5923 5929 5935 5941 5947 5954 5960 5965 5975 5981 5987 5996 6036 6043 6049
6055 6061 6067 6074 6080 6085 6095 6101 6107 6116 6156 6163 6169 6175 6181
6187 6194 6200 6205 6215 6221 6227 6236 6281 6288 6294 6300 6306 6312 6319
6325 6330 6336 6342 6348 6361 6406 6413 6419 6425 6431 6437 6444 6450 6455
6461 6467 6473 6486 6531 6538 6544 6550 6556 6562 6569 6575 6580 6586 6592
6598 6611 6656 6663 6669 6675 6681 6687 6694 6700 6705 6711 6717 6723 6736
6766 6771 6776 6781 6786 6792 6797 6802 6807 6816 6821 6827 6851 6856 6861
6866 6871 6877 6882 6887 6892 6901 6906 6912 6967 7111 7127 7163 7200 7231
7236 7273 7308 7335 7339 7348 7352 7384 7389 7394 7399 7404 7413 7418 7423
7428 7437 7442 7448 7456 7463 7467 7477 7481 7523 7528 7533 7538 7543 7552
7557 7562 7567 7576 7581 7587 7593 7639 7644 7649 7654 7659 7668 7673 7678
7683 7692 7697 7703 7709 7755 7760 7765 7770 7775 7784 7789 7794 7799 7808
7813 7819 7825 7871 7876 7881 7886 7891 7900 7905 7910 7915 7924 7929 7935
7941 7976 7979 7986 7989 7999 8002 8018 8021 8024 8039 8043 8070 8073 8081
8085 8117 8133 8170 8210 8215 8299
ESCAPE 1061#
GETPRI 1061#
GETSWR 1061# 1651#
LDLPER 1246# 2078 2298 2422 2546 2670 2795 2927 3059 3191 3404 3523 3642 3761 3882
4004 4126 4248 4370 4492 4614 4736 4859 4984 5109 5234 5784 5904 6024 6144
6264 6389 6514 6639 7511 7627 7743 7859

```
MSG     1731#  1733   1762#  1764   1858#  1860   1951#  1953   2041#  2043   2062#  2064   2167#  2169   2274#
        2276   2398#  2400   2522#  2524   2646#  2648   2770#  2772   2902#  2904   3034#  3036   3166#  3168
        3298#  3300   3384#  3386   3503#  3505   3622#  3624   3741#  3743   3860#  3862   3982#  3984   4104#
        4106   4226#  4228   4348#  4350   4470#  4472   4592#  4594   4714#  4716   4836#  4838   4961#  4963
        5086#  5088   5211#  5213   5336#  5338   5426#  5428   5511#  5513   5596#  5598   5680#  5682   5764#
        5766   5884#  5886   6004#  6006   6124#  6126   6244#  6246   6369#  6371   6494#  6496   6619#  6621
        6744#  6746   6829#  6831   6917#  6919   7166#  7168   7247#  7249   7279#  7281   7317#  7319   7355#
        7357   7485#  7487   7601#  7603   7717#  7719   7833#  7835   7949#  7951   8047#  8049   8088#  8090
        8141#  8143   8178#  8180
MULT    1061#
NEWTST  1061#  1731   1762   1858   1951   2041   2062   2167   2274   2398   2522   2646   2770   2902   3034
        3166   3298   3384   3503   3622   3741   3860   3982   4104   4226   4348   4470   4592   4714   4836
        4961   5086   5211   5336   5426   5511   5596   5680   5764   5884   6004   6124   6244   6369   6494
        6619   6744   6829   6917   7166   7247   7279   7317   7355   7485   7601   7717   7833   7949   8047
        8088   8141   8178
POP     1061#  8501   8502   8852   9075   9124
PUSH    1061#  8462   8464   8485   8811   9049   9104
REPORT  1061#
SCOPE    956#  1741   1776   1870   1961   2048   2073   2176   2292   2416   2540   2664   2788   2920   3052
        3184   3308   3398   3517   3636   3755   3876   3998   4120   4242   4364   4486   4608   4730   4852
        4977   5102   5227   5345   5435   5520   5604   5688   5778   5898   6018   6138   6258   6383   6508
        6633   6753   6838   6928   7178   7255   7288   7324   7373   7506   7622   7738   7854   7960   8055
        8098   8151   8188   8236
SETPRI  1061#
SETTRA  9184#  9193   9194   9195   9196   9198   9200   9201   9202   9203   9204   9205   9206
SETUP   1061#  1599
SKIP    1061#  1853   1946   2036   2055   2163   2255   2395   2519   2643   2767   2899   3031   3163   3295
        3379   3500   3619   3738   3857   3979   4101   4223   4345   4467   4589   4711   4833   4958   5083
        5208   5333   5422   5506   5591   5675   5759   5881   6001   6121   6241   6366   6491   6616   6741
        6824   6909   7112   7128   7149   7351   7598   7714   7830   7946   8084
SLASH   1061#
SPACE   1061#
STARS   1061#  1215   1226   1228   1235   1248   1297   1300   1731   1740   1762   1775   1858   1869   1951
        1960   2041   2047   2062   2072   2167   2175   2263   2272   2274   2291   2398   2415   2522   2539
        2646   2663   2770   2787   2902   2919   3034   3051   3166   3183   3298   3307   3384   3397   3503
        3516   3622   3635   3741   3754   3860   3875   3982   3997   4104   4119   4226   4241   4348   4363
        4470   4485   4592   4607   4714   4729   4836   4851   4961   4976   5086   5101   5211   5226   5336
        5344   5426   5434   5511   5519   5596   5603   5680   5687   5764   5777   5884   5897   6004   6017
        6124   6137   6244   6257   6369   6382   6494   6507   6619   6632   6744   6752   6829   6837   6917
        6927   7166   7177   7247   7254   7279   7287   7317   7323   7355   7372   7485   7505   7601   7621
        7717   7737   7833   7853   7949   7959   8047   8054   8088   8097   8141   8150   8178   8187   8228
        8310   8445   8457   8514   8567   8576   8645   8724   8801   8868   8871   8939   8968   9035   9088
        9134   9142   9163
SWRSU   1061#  1622#
TRMTRP  9184#
TYPBIN  1061#
TYPDEC  1061#  8251   8258
TYPNAM  1061#  1644
TYPNUM  1061#
TYPOCS  1061#
TYPOCT  1061#  1668   8889
TYPTXT  1061#  8247   8254
WRTDB   1246#  7508   7624   7740   7856
WRTRG1  1246#  3310   5437   5522   5606   5690   6755   6840
WRTRG2  1246#  2294   2418   2542   2666   2790   2922   3054   3186   3400   3519   3638   3757   3878   4000
        4122   4244   4366   4488   4610   4732   4854   4979   5104   5229   5780   5900   6020   6140   6260
```

```
        6385    6510    6635
$$CMRE  1246#
$$CMTM  1246#   1283    1284    1285    1286    1287    1288    1289    1290
$$ESCA  1061#
$$NEWT  1061#   1731    1762    1858    1951    2041    2062    2167    2274    2398    2522    2646    2770    2902    3034
        3166    3298    3384    3503    3622    3741    3860    3982    4104    4226    4348    4470    4592    4714    4836
        4961    5086    5211    5336    5426    5511    5556    5680    5764    5884    6004    6124    6244    6369    6494
        6619    6744    6829    6917    7166    7247    7279    7317    7355    7485    7601    7717    7833    7949    8047
        8088    8141    8178
$$SET   9184#   9193    9194    9195    9196    9198    9200    9201    9202    9203    9204    9205    9206
$$SETM  1638#
$$SKIP  1061#   1853    1946    2036    2055    2163    2255    2395    2519    2643    2767    2899    3031    3163    3295
        3379    3500    3619    3738    3857    3979    4101    4223    4345    4467    4589    4711    4833    4958    5083
        5208    5333    5422    5506    5591    5675    5759    5881    6001    6121    6241    6366    6491    6616    6741
        6824    6909    7112    7128    7149    7351    7598    7714    7830    7946    8084
.EQUAT   929#    951
.HEADE   929#
.SETUP   929#   1586
.SWRHI   929#    939
.SWRLO   929#    951#
.$ACTI   929#   1213
.$APTB  1298#
.$APTH   929#   1224
.$APTY   929#   8455
.$CATC   929#   1199
.$CMTA   929#   1246
.$EOP    929#   8226
.$ERRO   929#   8512
.$ERRT   929#
.$POWE   929#
.$RDOC   929#   9033
.$READ   929#   8866
.$SAVE   929#   9086
.$SCOP   929#   8308
.$TRAP   929#   9161
.$TYPD   929#   8799
.$TYPE   929#   8643
.$TYPO   929#   8722

. ABS.  071160     000

 ERRORS DETECTED:  0

 RM03:CZR6AC,RM03:CZR6AC.SEQ/SOL/CRF/NL:TOC/DOC=RM03:CZR6AC.P11
 RUN-TIME:  38 35 3 SECONDS
 RUN-TIME RATIO: 1094/77=14.1
 CORE USED:  31K  (61 PAGES)

 DOCUMENT PAGES:  207

EOF1CZR6ACSEQ              00010000      780223            PDP10 411
```