

RK611

DISKLESS CONTR DIAG 1
MD-11-DZR6A-B

EP-DZR6A B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 2

NOV 1976
digital
MADE IN USA

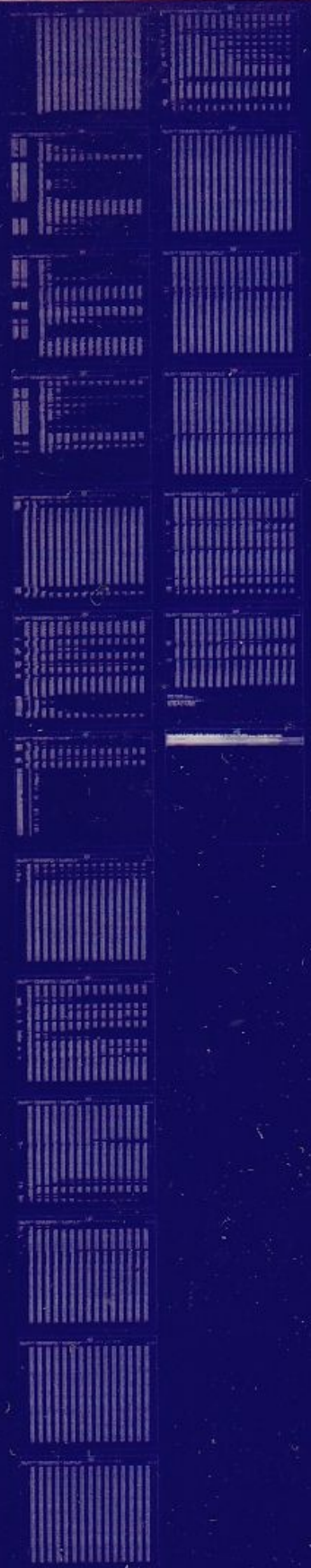
The image displays a grid of 100 small diagrams, arranged in 10 rows and 10 columns. Each diagram is a technical schematic or flowchart, likely representing a diagnostic procedure for a diskless control system. The diagrams are densely packed and contain various symbols, lines, and text, though the individual details are too small to read clearly. The overall layout is a structured grid of technical information.

RK611

DIAGNOSTIC PART 1
MD-11-DZR6A-B

EP-DZR6A-B-DL-A
COPYRIGHT © 1976
FICHE 2 OF 2

NOV 1976
digital
MADE IN USA



IDENTIFICATION

PROJECT CODE: MAINDEC-11-DZREA-b
PRODUCT NAME: RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 1
DATE: AUG. 1976
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: RO SPITZER

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

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

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

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 PRELIMINARY PROGRAMS
- 3.0 OPERATING PROGRAMS
 - 3.1 LOADING PROCEDURE
 - 3.2 STARTING PROCEDURE
 - 3.3 OPTIONAL SWITCH SETTING
 - 3.4 RUN TIME
- 4.0 OPERATING PROCEDURES
- 5.0 PROGRAM DESCRIPTION
- 6.0 ERROR REPORTING

1.0 ABSTRACT

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

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 SYSTEM (16K CORE MEMORY)
CONSOLE TERMINAL
DEC TAPE, PAPER TAPE READER, OR DEC DISK
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 XQOP 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

E01

3.3 OPTIONAL SWITCH SETTINGS

SW15 - HALT PROGRAM
SW14 - LOOP ON TEST
SW13 - INHIBIT ERROR TYPE OUT
SW12 - REJCT 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 APPROPRIATE 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 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.

F01

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 INTERACTION 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 CLEARS.

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 PLACE. 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 INTERACTION TAKES PLACE. A CONTROLLER CLEAR IS DONE AT THE END OF THE TEST TO MAKE SURE THE DISK ADDRESS CLEARS.

GO1

**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 CONTROLLER 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.

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 CONTROLLER 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	

TEST 12 REGISTER INTERACTION USING BUS ADDRESS (PART 3)

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER 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.

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

H01

TEST 13 REGISTER INTERACTION USING BUS ADDRESS (PART 4)

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER 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.

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 AND 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 AND 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	

I01

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

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND 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 AND 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 REGISTERS. WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE SPARE REGISTER WITH 177777 AND MAKE SURE NO INTERACTION TAKES PLACE.

J01

TEST 21 REGISTER INTERACTION USING WORD COUNT (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
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 REGISTERS.
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
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 REGISTERS.
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
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 REGISTERS.
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
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	

K01

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	

TEST 26 REGISTER INTERACTION USING DISK ADDRESS (PART 2)

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.

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 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.

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

LO1

test 30 register interaction using disk address (part 4)

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.

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 REGISTERS.
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.

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 REGISTERS.
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	

MO1

TEST 33 REGISTER INTERACTION USING ATTN/OFFSET (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
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.

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 REGISTERS.
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.

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 CONFIGURATIONS
SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
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 CONFIGURATIONS
SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
REG. 2 AND CHECK FOR REGISTER INTERACTION.

177737	177727	177337	167737
177736	177717	176737	157737
177735	177637	175737	137737
177733	177537	173737	077737

NO1

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 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

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

TEST 41 CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS

THIS TEST WILL TEST THE ABILITY OF THE SUBSYSTEM CLEAR TO
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 INITIALIZE RK611 REGISTERS.
WRITE WORD COUNT TO 0. WRITE DRIVE STATUS REGISTER
WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
INTERACTION TAKES PLACE.

802

TEST 43 REGISTER INTERACTION USING ERROR REGISTER

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE ERROR REGISTER
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.

C02

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. CHECK IF CONTENTS
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. CHECK IF CONTENTS
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. CHECK IF CONTENTS
ARE CORRECT AND NO INTERACTION TAKES PLACE.

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

002

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	

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 CONTENTS
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 CONTENTS
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 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 CONTENTS
ARE CORRECT AND NO INTERACTION TAKES PLACE.

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

E02

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 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	

TEST 56 REGISTER INTERACTION WITH PATTERN REG.

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE PATTERN
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 POSITION
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 OCCURS.

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

F02

TEST 62 SETTING INTERRUPT ENABLE

CLEAR RK611 CONTROLLER WITH CONTROLLER CLEAR. ALLOW RK611 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 CONTROLLER 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 CONTROLLER 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 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:

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

G02

378
F12

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 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:

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 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	

H02

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	

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 CHECKING CONTENTS, INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH 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 CLEAR. 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 BUFFER WHEN EMPTY. CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. 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 INTERRUPT
 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.

12	OPERATIONAL SWITCH SETTINGS
24	BASIC DEFINITIONS
141	RK611 CONTROLLER REGISTER DEFINITION
160	DRIVE COMMANDS
177	CONTROL AND STATUS REGISTER 1 BITS
193	CONTROL AND STATUS REGISTER 2 BITS
210	ERROR REGISTER BIT DEFINITION
229	STATUS REGISTER BIT DEFINITION
245	MAINTENANCE REGISTER 1 BIT DEFINITION
262	TRANSMITTED MESSAGE A
272	TRAP CATCHER
281	STARTING ADDRESS(ES)
286	ACT11 HOOKS
297	APT PARAMETER BLOCK
319	COMMON TAGS
371	APT MAILBOX-ETABLE
421	ERROR POINTER TABLE
602	TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
621	EXPECTED RK611 CONTROLLER REGISTERS
640	PROGRAM DEFINED VARIABLES
657	PROGRAM SETUP
673	INITIALIZE THE COMMON TAGS
717	TYPE PROGRAM NAME
724	GET VALUE FOR SOFTWARE SWITCH REGISTER
803	T1 ADDRESS ALL RK611 REGISTERS
833	**RESET, CONTROLLER CLEAR, AND TRI-STATE TESTS
835	T2 RESET RK611 AND VERIFY REGISTERS
932	T3 CONTROLLER CLEAR AND VERIFY REGISTERS
1026	T4 WRITE BUS ADDRESS WITH 177777 (PART 1)
1117	T5 WRITE BUS ADDRESS WITH 177777 (PART 2)
1139	T6 WRITE WORD COUNT REG. WITH 177777
1245	T7 WRITE DISK ADDRESS WITH 177777
1343	**REGISTER INTERACTION TESTS
1353	T10 REGISTER INTERACTION USING BUS ADDRESS (PART 1)
1478	T11 REGISTER INTERACTION USING BUS ADDRESS (PART 2)
1603	T12 REGISTER INTERACTION USING BUS ADDRESS (PART 3)
1728	T13 REGISTER INTERACTION USING BUS ADDRESS (PART 4)
1853	T14 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)
1986	T15 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)
2119	T16 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)
2252	T17 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)
2385	T20 REGISTER INTERACTION USING SPARE REG
2472	T21 REGISTER INTERACTION USING WORD COUNT (PART 1)
2592	T22 REGISTER INTERACTION USING WORD COUNT (PART 2)
2712	T23 REGISTER INTERACTION USING WORD COUNT (PART 3)
2832	T24 REGISTER INTERACTION USING WORD COUNT (PART 4)
2952	T25 REGISTER INTERACTION USING DISK ADDRESS (PART 1)
3075	T26 REGISTER INTERACTION USING DISK ADDRESS (PART 2)
3198	T27 REGISTER INTERACTION USING DISK ADDRESS (PART 3)
3321	T30 REGISTER INTERACTION USING DISK ADDRESS (PART 4)
3444	T31 REGISTER INTERACTION USING ATTN/OFFSET (PART 1)
3567	T32 REGISTER INTERACTION USING ATTN/OFFSET (PART 2)
3690	T33 REGISTER INTERACTION USING ATTN/OFFSET (PART 3)
3813	T34 REGISTER INTERACTION USING ATTN/OFFSET (PART 4)
3936	T35 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1)
4062	T36 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 2)

4188	T37	REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 3)
4314	T40	REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 4)
4440	T41	CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS
4531	T42	REGISTER INTERACTION USING DRIVE STATUS
4617	T43	REGISTER INTERACTION USING ERROR REGISTER
4703	T44	REGISTER INTERACTION USING MAINT REG 2
4788	T45	REGISTER INTERACTION USING MAINT REG 3
4873	T46	REGISTER INTERACTION WITH DISK CYLINDER (PART 1)
4994	T47	REGISTER INTERACTION WITH DISK CYLINDER (PART 2)
5115	T50	REGISTER INTERACTION WITH DISK CYLINDER (PART 3)
5236	T51	REGISTER INTERACTION WITH DISK CYLINDER (PART 4)
5357	T52	REGISTER INTERACTION USING MAINT REG 1 (PART 1)
5483	T53	REGISTER INTERACTION USING MAINT REG 1 (PART 2)
5609	T54	REGISTER INTERACTION USING MAINT REG 1 (PART 3)
5735	T55	REGISTER INTERACTION USING MAINT REG 1 (PART 4)
5861	T56	REGISTER INTERACTION WITH PATTERN REG.
5947	T57	REGISTER INTERACTION WITH POSITION REG.
6034		**INTERRUPT TESTS
6036	T60	RK611 INTERRUPT
6286	T61	INTERRUPT PRIORITY
6368	T62	SETTING INTERRUPT ENABLE
6401	T63	INTERRUPT CLEARING
6438		**SILO TESTS
6440	T64	READ SILO WHEN EMPTY
6479	T65	SILO LOADING AND UNLOADING OF ONE WORD
6610	T66	ONE WORD SILO WRITE AND REG. INTERACTION (PART 1)
6727	T67	ONE WORD SILO WRITE AND REG. INTERACTION (PART 2)
6844	T70	ONE WORD SILO WRITE AND REG. INTERACTION (PART 3)
6961	T71	ONE WORD SILO WRITE AND REG. INTERACTION (PART 4)
7078	T72	SILO FILL
7177	T73	SILO CAPACITY DATA LATE
7219	T74	INTERRUPT DUE TO DATA LATE
7273	T75	INTERRUPT CLEARING AND DATA LATE
7311	T76	INTERRUPT ENABLE AND DATA LATE
7360		END OF PASS ROUTINE
7410		CHECK FOR MEMORY CHECK ENABLE
7429		MEMORY CHECK ENABLE TRAP
7442		SCOPE HANDLER ROUTINE
7582		LOOP ON INTERNAL ERROR
7590		APT COMMUNICATIONS ROUTINE
7647		ERROR HANDLER ROUTINE
7703		TYPE ERROR ROUTINE
7778		TYPE ROUTINE
7857		BINARY TO OCTAL (ASCII) AND TYPE
7934		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
8001		TTY INPUT ROUTINE
8168		READ AN OCTAL NUMBER FROM THE TTY
8221		SAVE AND RESTORE RD-R5 ROUTINES
8267		POWER DOWN AND UP ROUTINES
8296		TRAP DECODER
8319		TRAP TABLE
8342		DATA TABLE FOR ERROR PRINT OUT
8364		DATA FORMAT FOR ERROR PRINT OUT
8460		ASCII MESSAGES
8487		DATA HEADERS
8578		ERROR MESSAGES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```

.TITLE RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B
;*COPYRIGHT (C) 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY ROY SPITZER
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.
;*
.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 ***
STACK= 1100
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

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

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0                ;;GENERAL REGISTER
R1= %1                ;;GENERAL REGISTER
R2= %2                ;;GENERAL REGISTER
R3= %3                ;;GENERAL REGISTER
R4= %4                ;;GENERAL REGISTER
R5= %5                ;;GENERAL REGISTER
R6= %6                ;;GENERAL REGISTER
R7= %7                ;;GENERAL REGISTER
.EQUIV R6,SP          ;;STACK POINTER
.EQUIV R7,PC          ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS
PRO= 0                ;;PRIORITY LEVEL 0
PRI= 40               ;;PRIORITY LEVEL 1

```

001100

000011
000012
000013
000200
177776
177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007

000000
000040

57	000100	PR2=	100	::	PRIORITY LEVEL	2
58	000140	PR3=	140	::	PRIORITY LEVEL	3
59	000200	PR4=	200	::	PRIORITY LEVEL	4
60	000240	PR5=	240	::	PRIORITY LEVEL	5
61	000300	PR6=	300	::	PRIORITY LEVEL	6
62	000340	PR7=	340	::	PRIORITY LEVEL	7

.*"SWITCH REGISTER" SWITCH DEFINITIONS

63		SW15=	100000
64		SW14=	40000
65	100000	SW13=	20000
66	040000	SW12=	10000
67	020000	SW11=	4000
68	010000	SW10=	2000
69	004000	SW09=	1000
70	002000	SW08=	400
71	001000	SW07=	200
72	000400	SW06=	100
73	000200	SW05=	40
74	000100	SW04=	20
75	000040	SW03=	10
76	000020	SW02=	4
77	000010	SW01=	2
78	000004	SW00=	1
79	000002	.EQUIV	SW09, SW9
80	000001	.EQUIV	SW08, SW8
81		.EQUIV	SW07, SW7
82		.EQUIV	SW06, SW6
83		.EQUIV	SW05, SW5
84		.EQUIV	SW04, SW4
85		.EQUIV	SW03, SW3
86		.EQUIV	SW02, SW2
87		.EQUIV	SW01, SW1
88		.EQUIV	SW00, SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

92		BIT15=	100000
93	100000	BIT14=	40000
94	040000	BIT13=	20000
95	020000	BIT12=	10000
96	010000	BIT11=	4000
97	004000	BIT10=	2000
98	002000	BIT09=	1000
99	001000	BIT08=	400
100	000400	BIT07=	200
101	000200	BIT06=	100
102	000100	BIT05=	40
103	000040	BIT04=	20
104	000020	BIT03=	10
105	000010	BIT02=	4
106	000004	BIT01=	2
107	000002	BIT00=	1
108	000001	.EQUIV	BIT09, BIT9
109		.EQUIV	BIT08, BIT8
110		.EQUIV	BIT07, BIT7
111		.EQUIV	BIT06, BIT6
112			

```

113      .EQUIV BIT05,BIT5
114      .EQUIV BIT04,BIT4
115      .EQUIV BIT03,BIT3
116      .EQUIV BIT02,BIT2
117      .EQUIV BIT01,BIT1
118      .EQUIV BIT00,BIT0
119
120      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
121      000004 ERRVEC= 4      ;: TIME OUT AND OTHER ERRORS
122      000010 RESVEC= 10   ;: RESERVED AND ILLEGAL INSTRUCTIONS
123      000014 TBITVEC=14  ;: "T" BIT
124      000014 TRTVEC= 14   ;: TRACE TRAP
125      000014 BPTVEC= 14   ;: BREAKPOINT TRAP (BPT)
126      000020 IOTVEC= 20   ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
127      000024 PWRVEC= 24   ;: POWER FAIL
128      000030 EMTVEC= 30   ;: EMULATOR TRAP (EMT) **ERROR**
129      000034 TRAPVEC=34  ;: "TRAP" TRAP
130      000060 TKVEC= 60    ;: TTY KEYBOARD VECTOR
131      000064 TPVEC= 64    ;: TTY PRINTER VECTOR
132      000240 PIRQVEC=240  ;: PROGRAM INTERRUPT REQUEST VECTOR
133      000114 MEMVEC= 114  ;: MEMORY PARITY VECTOR
134      172100 MEMBAS= 172100 ;: MEMORY PARITY BASE ADDRESS
135      000001 PAR.EN= 1    ;: ALLOW MEMORY PARITY OPTION
136      120210 AVECT1= 120210 ;: DEFINE RK611 VECTOR ADDRESS
137      000005 APRIOR= 5    ;: DEFINE RK611 PRIORITY
138      177440 ABASE= 177440 ;: DEFINE BASE OF RK611 REGISTERS
139
140      .SBTTL RK611 CONTROLLER REGISTER DEFINITION
141
142      000000 RKCS1= 0      ;: CONTROL AND STATUS REGISTER 1
143      000002 RKWC= 2      ;: WORD COUNT REGISTER
144      000004 RKBA= 4      ;: BUS ADDRESS REGISTER
145      000006 RKDA= 6      ;: DESIRED TRACK SECTOR REGISTER
146      000010 RKCS2= 10   ;: CONTROL AND STATUS REGISTER 2
147      000012 RKDS= 12   ;: DRIVE STATUS REGISTER
148      000014 RKER= 14   ;: ERROR REGISTER
149      000016 RKASOF= 16  ;: ATTENTION SUMMARY AND OFFSET REGISTER
150      000020 RKDCYL= 20  ;: DESIRED CYLINDER REGISTER
151      000024 RKDB= 24   ;: DATA BUFFER
152      000026 RKMR1= 26  ;: MAINTENANCE REGISTER 1
153      000034 RKMR2= 34  ;: MAINTENANCE REGISTER 2
154      000036 RKMR3= 36  ;: MAINTENANCE REGISTER 3
155      000030 RKECPS= 30  ;: ECC POSITION INFORMATION
156      000032 RKECPT= 32  ;: ECC PATTERN INFORMATION
157      000022 RKSPAR= 22  ;: SPARE REGISTER
158
159      .SBTTL DRIVE COMMANDS
160
161      000001 SELDRV= 01   ;: SELECT DRIVE
162      000003 PACK= 03   ;: PACK ACKNOWLEDGE
163      000005 CLEAR= 05   ;: DRIVE CLEAR
164      000007 UNLOAD= 07  ;: UNLOAD
165      000011 SRTSPL= 11  ;: START SPINDLE
166      000013 RECAL= 13   ;: RECALIBRATE
167      000015 OFFSET= 15  ;: OFFSET
168      000017 SEEK= 17    ;: SEEK
    
```

169	000021	RCDATA= 21	: READ DATA
170	000023	WRDATA= 23	: WRITE DATA
171	000025	RDHEAD= 25	: READ HEADER
172	000027	WRHEAD= 27	: WRITE HEADER AND DATA
173	000031	WRTCHK= 31	: WRITE CHECK
174	000300	INTR= 300	: GENERATE INTERRUPT TO CPU
175			
176		.SBTTL CONTROL AND STATUS REGISTER 1 BITS	
177			
178	000001	GO= BIT0	: GO BIT
179	000100	IE= BIT6	: INTERRUPT ENABLE
180	000200	RDY= BIT7	: CONTROLLER READY
181	000400	BA16= BIT8	: BUS ADDRESS BIT 16
182	001000	BA17= BIT9	: BUS ADDRESS BIT 17
183	002000	CDT= BIT10	: CONTROLLER DRIVE TYPE (0=RK06)
184	004000	CTO= BIT11	: CONTROLLER TIMED OUT WAITING FOR : DRIVE RESPONSE
185			
186	010000	CFMT= BIT12	: CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
187	020000	SPAR= BIT13	: DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
188	040000	DI= BIT14	: DRIVE INTERRUPT
189	100000	CERR= BIT15	: CONTROLLER ERROR
190	100000	CCLR= BIT15	: CONTROLLER CLEAR
191			
192		.SBTTL CONTROL AND STATUS REGISTER 2 BITS	
193			
194	000007	DRVMSK= 7	: MASK FOR DRIVE SELECTION CODE
195	000010	RLS= BIT3	: DESELECT OR RELEASE DRIVE IN BITS 0-2
196	000020	BAI= BIT4	: BUS ADDRESS INCREMENT INHIBIT
197	000040	SCLR= BIT5	: CLEAR CONTROLLER AND ALL DRIVES
198	000100	IR= BIT6	: INPUT READY
199	000200	OR= BIT7	: OUTPUT READY
200	000400	UFE= BIT8	: UNIT FIELD ERROR
201	001000	MDS= BIT9	: MULTIPLE DRIVE SELECT
202	002000	PGE= BIT10	: PROGRAMMING ERROR
203	004000	NEM= BIT11	: NON-EXISTENT MEMORY
204	010000	NED= BIT12	: NON-EXISTENT DRIVE
205	020000	UPE= BIT13	: UNIBUS PARITY ERROR
206	040000	WCE= BIT14	: WRITE CHECK ERROR
207	100000	DLT= BIT15	: DATA LATE ERROR
208			
209		.SBTTL ERROR REGISTER BIT DEFINITION	
210			
211	000001	ILF= BIT0	: ILLEGAL FUNCTION CODE
212	000002	SKI= BIT1	: SEEK INCOMPLETE
213	000004	NXF= BIT2	: NON-EXECUTABLE DRIVE FUNCTION
214	000010	DRPAR= BIT3	: DRIVE DETECTED DRIVE BUS PARITY ERROR
215	000020	FMTE= BIT4	: FORMAT ERROR
216	000040	DTYPE= BIT5	: DRIVE TYPE ERROR
217	000100	ECH= BIT6	: ECC HARD
218	000200	BSE= BIT7	: BAD SECTOR ERROR
219	000400	HVRC= BIT8	: HEADER VRC ERROR
220	001000	COE= BIT9	: CYLINDER ADDRESS OVERFLOW ERROR
221	002000	IDAE= BIT10	: INVALID DISK ADDRESS ERROR
222	004000	WLE= BIT11	: WRITE LOCK ERROR
223	010000	DTE= BIT12	: DRIVE TIMING ERROR
224	020000	OPI= BIT13	: OPERATION (SEARCH) INCOMPLETE


```

225 040000 UNS= BIT14 ;DRIVE UNSAFE
226 100000 DCK= BIT15 ;DATA CHECK
227
228 .SBTTL STATUS REGISTER BIT DEFINITION
229
230 000001 DRA= BIT0 ;DRIVE AVAILABLE (CONTROLLER IS SET IF
231 ; THIS BIT IS RESET)
232 000004 OFST= BIT2 ;DRIVE OFFSET
233 000010 ACLO= BIT3 ;AC LOW
234 000020 SPOLSS= BIT4 ;SPEED LOSS
235 000040 DROT= BIT5 ;DRIVE OFF TRACK
236 000100 VV= BIT6 ;VOLUME VALID
237 000200 DRDY= BIT7 ;DRIVE READY
238 000400 DDT= BIT8 ;DRIVE TYPE (0=RK06)
239 004000 WRL= BIT11 ;WRITE LOCK
240 020000 PIP= BIT13 ;POSITIONING IN PROGRESS
241 040000 DSC= BIT14 ;DRIVE STATUS CHANGE
242 100000 SVAL= BIT15 ;STATUS VALID
243
244 .SBTTL MAINTENANCE REGISTER 1 BIT DEFINITION
245
246 000017 MESMSK= 17 ;MESSAGE MASK
247
248 000020 PAT= BIT4 ;FORCE EVEN PARITY ON DRIVE MESSAGE LINES
249 000040 DMD= BITS ;DIAGNOSTIC MODE
250 000100 MSP= BIT6 ;MAINTENANCE SECTOR PULSE
251 000200 MIND= BIT7 ;MAINTENANCE INDEX
252 000400 MCLK= BIT8 ;MAINTENANCE CLOCK
253 001000 MERD= BIT9 ;MAINTENANCE ENCODED READ DATA
254 002000 MEWD= BIT10 ;MAINTENANCE ENCODED WRITE DATA
255 004000 PCA= BIT11 ;PRECOMPENSATION ADVANCE
256 010000 PCD= BIT12 ;PRECOMPENSATION DELAY
257 020000 ECCW= BIT13 ;ECC WORD IS BEING READ OR WRITTEN
258 040000 WRTGAT= BIT14 ;WRITE GATE
259 100000 RDGATE= BIT15 ;READ GATE
260
261 .SBTTL TRANSMITTED MESSAGE A
262
263 000020 S.SEK= BIT4 ;SEEK COMMAND
264 000040 S.RECL= BITS ;RECALIBRATE COMMAND
265 000100 S.STSP= BIT6 ;START SPINDLE COMMAND
266 000200 S.RTC= BIT7 ;DRIVE RETURN TO CENTERLINE COMMAND
267 000400 S.CLR= BIT8 ;CLEAR ERROR AND DSC
268 001000 S.FMT= BIT9 ;FORMAT
269 002000 S.UNLD= BIT10 ;UNLOAD
270 004000 S.PACK= BIT11 ;SET VOLUME VALID (PACK ACKNOWLEDGE)
271 .SBTTL TRAP CATCHER
272
273 000000 .=0
274 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "+2,HALT"
275 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
276 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
277 000174 .=174
278 000174 000000 DISPREG: .WORD 0 ;SOFTWARE DISPLAY REGISTER
279 000176 000000 SWREG: .WORD 0 ;SOFTWARE SWITCH REGISTER
280 .SBTTL STARTING ADDRESS(ES)

```

281 000200 000137 002050
 282 000204 000137 002040
 283
 284 000214 000137 002030
 285
 286
 287
 288
 289 000220
 290 000046
 291 000046 055254
 292 000052
 293 000052 000000
 294 000220
 295 001000
 296
 297
 298
 299
 300
 301 001000
 302 000024
 303 000024 000200
 304 000044
 305 000044 001000
 306 001000
 307
 308
 309
 310
 311 001000
 312 001000 000000
 313 001002 001214
 314 001004 000001
 315 001006 000007
 316 001010 000007
 317 001012 000032

```

JMP 2*START ;;JUMP TO STARTING ADDRESS OF PROGRAM
JMP RESTART ;JUMP TO RESTART ROUTINE
.=214
JMP PARM ;JUMP TO OPERATOR ASSIGNED PARAMETERS
.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
$VPC=. ;SAVE PC
.=46
$ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
.=52
.WORD 0 ;;2)SET LOC.52 TO ZERO
.=52VPC ;; RESTORE PC
.=1000
.SBTTL APT PARAMETER BLOCK
;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
.$X=. ;;SAVE CURRENT LOCATION
.=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200 ;;FOR APT START UP
.=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR ;;POINT TO APT HEADER BLOCK
.=.$X ;;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBAOR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM: .WORD 1 ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 7 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 7 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
.WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373

.SBTTL COMMON TAGS

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

001100

SCMTAG: . =1100

;; START OF COMMON TAGS

001100 000000
001102 000
001103 000
001104 000000
001106 000000
001110 000000
001112 000000
001114 000
001115 001
001116 000000
001120 000000
001122 000000
001124 000000
001126 000000
001130 000000
001132 000000
001134 000
001135 000
001136 000000
001140 177570
001142 177570
001144 177560
001146 177562
001150 177564
001152 177566
001154 000
001155 002
001156 012
001157 000
001160 000000
001162 000000
001164 000000
001166 000000
001170 000000
001172 000000
001174 000000
001176 000000
001200 000000
001202 000000
001204 177607
001210 077
001211 015
001212 000012

.WORD 0
STSTNM: .BYTE 0
SERFLG: .BYTE 0
SICNT: .WORD 0
SLPADR: .WORD 0
SLPERR: .WORD 0
SERTTL: .WORD 0
SITEMB: .BYTE 0
SERMAX: .BYTE 1
SERRPC: .WORD 0
SGDADR: .WORD 0
\$BDADR: .WORD 0
\$GDDAT: .WORD 0
\$BDDAT: .WORD 0
SWORD .WORD 0
SAUTOB: .BYTE 0
SINTAG: .BYTE 0
SWR: .WORD DSWR
DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$TPFLG: .BYTE 0
\$TMP0: .WORD 0
\$TMP1: .WORD 0
\$TMP2: .WORD 0
\$TMP3: .WORD 0
\$TMP4: .WORD 0
\$TMP5: .WORD 0
\$TMP6: .WORD 0
\$TMP7: .WORD 0
\$TIMES: 0
\$ESCAPE: 0
\$BELL: .ASCIZ <207><377><377>
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCIZ <12>

;; CONTAINS THE TEST NUMBER
;; CONTAINS ERROR FLAG
;; CONTAINS SUBTEST ITERATION COUNT
;; CONTAINS SCOPE LOOP ADDRESS
;; CONTAINS SCOPE RETURN FOR ERRORS
;; CONTAINS TOTAL ERRORS DETECTED
;; CONTAINS ITEM CONTROL BYTE
;; CONTAINS MAX. ERRORS PER TEST
;; CONTAINS PC OF LAST ERROR INSTRUCTION
;; CONTAINS ADDRESS OF 'GOOD' DATA
;; CONTAINS ADDRESS OF 'BAD' DATA
;; CONTAINS 'GOOD' DATA
;; CONTAINS 'BAD' DATA
;; RESERVED--NOT TO BE USED
;; AUTOMATIC MODE INDICATOR
;; INTERRUPT MODE INDICATOR
;; ADDRESS OF SWITCH REGISTER
;; ADDRESS OF DISPLAY REGISTER
;; TTY KBD STATUS
;; TTY KBD BUFFER
;; TTY PRINTER STATUS REG. ADDRESS
;; TTY PRINTER BUFFER REG. ADDRESS
;; CONTAINS NULL CHARACTER FOR FILLS
;; CONTAINS # OF FILLER CHARACTERS REQUIRED
;; INSERT FILL CHARS. AFTER A "LINE FEED"
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; MAX. NUMBER OF ITERATIONS
;; ESCAPE ON ERROR ADDRESS
;; CODE FOR BELL
;; QUESTION MARK
;; CARRIAGE RETURN
;; LINE FEED

000377

.SBTTL APT MAILBOX-ETABLE

.EVEN

F03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 9
 DZR6A8.P11 APT MAILBOX-ETABLE

SEQ 0010

374	001214		\$MAIL:		:: APT MAILBOX
375	001214	000000	\$MSGTY: .WORD	AMSGTY	:: MESSAGE TYPE CODE
376	001216	000000	\$FATAL: .WORD	AFATAL	:: FATAL ERROR NUMBER
377	001220	000000	\$TESTN: .WORD	ATESTN	:: TEST NUMBER
378	001222	000000	\$PASS: .WORD	APASS	:: PASS COUNT
379	001224	000000	\$DEVCT: .WORD	ADEVCT	:: DEVICE COUNT
380	001226	000000	\$UNIT: .WORD	AUNIT	:: I/O UNIT NUMBER
381	001230	000000	\$MSGAD: .WORD	AMSGAD	:: MESSAGE ADDRESS
382	001232	000000	\$MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
383	001234		\$ETABLE:		:: APT ENVIRONMENT TABLE
384	001234	000	\$ENV: .BYTE	AENV	:: ENVIRONMENT BYTE
385	001235	000	\$ENVM: .BYTE	AENVM	:: ENVIRONMENT MODE BITS
386	001236	000000	\$SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
387	001240	000000	\$USWR: .WORD	AUSWR	:: USER SWITCHES
388	001242	000000	\$CPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
389			*		BITS 15-11=CPU TYPE
390			*		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
391			*		11/70=06, PDQ=07, Q=10
392			*		BIT 10=REAL TIME CLOCK
393			*		BIT 9=FLOATING POINT PROCESSOR
394			*		BIT 8=MEMORY MANAGEMENT
395	001244	000	\$MAMS1: .BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
396	001245	000	\$MTYP1: .BYTE	AMTYP1	:: MEM. TYPE, BLK#1
397			*		MEM. TYPE BYTE -- (HIGH BYTE)
398			*		900 NSEC CORE=001
399			*		300 NSEC BIPOLAR=002
400			*		500 NSEC MOS=003
401	001246	000000	\$MADR1: .WORD	AMADR1	:: HIGH ADDRESS, BLK#1
402			*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
403	001250	000	\$MAMS2: .BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
404	001251	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
405	001252	000000	\$MADR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
406	001254	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
407	001255	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
408	001256	000000	\$MADR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
409	001260	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
410	001261	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
411	001262	000000	\$MADR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
412	001264	120210	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
413	001266	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
414	001270	177440	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
415	001272	000000	\$DEVH: .WORD	ADEVH	:: DEVICE MAP
416	001274	000000	\$CDW1: .WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
417	001276	000000	\$CDW2: .WORD	ACDW2	:: CONTROLLER DESCRIPTION WORD#2
418	001300		\$ETEND:		
419			.MEXIT		

420
421
422
423
424
425
426
427
428
429
430
431
432
433
434 001300
435
436 001300 063261
437 001302 000000
438 001304 061426
439 001306 061572
440
441 001310 000000
442 001312 000000
443 001314 061434
444 001316 061612
445
446 001320 000000
447 001322 000000
448 001324 061444
449 001326 061636
450
451 001330 000000
452 001332 000000
453 001334 061456
454 001336 061662
455
456 001340 064776
457 001342 000000
458 001344 061470
459 001346 061706
460
461 001350 065070
462 001352 000000
463 001354 061474
464 001356 061722
465
466 001360 065150
467 001362 000000
468 001364 061474
469 001366 061722
470
471 001370 065206
472 001372 000000
473 001374 061474
474 001376 061722
475

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:
 ; ERROR 1: NEM ACCESSING RK611 REGISTER
 ; EM1
 ; 0
 ; DT001
 ; DF001
 ; ERROR 2: ATTEMPTING TO INITIALIZE RK611
 ; EM2N:
 ; 0
 ; 0
 ; DT002
 ; DF002
 ; ERROR 3: ATTEMPTING TO WRITE REG
 ; EM3N:
 ; 0
 ; 0
 ; DT003
 ; DF003
 ; ERROR 4: ATTEMPTING TO CLEAR RK611
 ; EM4N:
 ; 0
 ; 0
 ; DT004
 ; DF004
 ; ERROR 5: INTERRUPT DID NOT OCCUR
 ; EM24
 ; 0
 ; DT005
 ; DF005
 ; ERROR 6: UNEXPECTED INTERRUPT WHEN LOWERING PROCESSOR PRIORITY
 ; EM25
 ; 0
 ; DT006
 ; DF006
 ; ERROR 7: VECTOR ADDRESS INCORRECT
 ; EM26
 ; 0
 ; DT006
 ; DF006
 ; ERROR 10: INTERRUPT DID NOT CLEAR IN RK611
 ; EM27
 ; 0
 ; DT006
 ; DF006
 ; ERROR 11: EXPECTED INTERRUPT DID NOT OCCUR AT PROCESSOR PRIORITY

H03

476	001400	065242	EM28	
477	001402	000000	0	
478	001404	061504	DT011	
479	001406	061746	DF011	
480			ERROR 12:	UNEXPECTED INTERRUPT OCCURRED AT PROCESSOR PRIORITY
481	001410	065324	EM29	
482	001412	000000	0	
483	001414	061504	DT011	
484	001416	061746	DF011	
485			ERROR 13:	INTERRUPT DID NOT OCCUR WHEN PRIORITY LOWERED
486	001420	065403	EM30	
487	001422	000000	0	
488	001424	061504	DT011	
489	001426	061746	DF011	
490			ERROR 14:	SETTENG INTERRUPT ENABLE CAUSED EXPECTED INTERRUPT
491	001430	065454	EM31	
492	001432	000000	0	
493	001434	061470	DT005	
494	001436	061706	DF005	
495			ERROR 15:	CONTROLLER CLEAR DID NOT CLEAR STORED INTERRUPT
496	001440	065525	EM32	
497	001442	000000	0	
498	001444	061470	DT005	
499	001446	061706	DF005	
500			ERROR 16:	ATTEMPTING TO READ SILO WHEN EMPTY - CS2 INCORRECT
501	001450	065642	EM34	
502	001452	067102	EM1009	
503	001454	061512	DT016	
504	001456	061766	DF016	
505			ERROR 17:	ATTEMPTING TO READ SILO WHEN EMPTY - CS1 INCORRECT
506	001460	065642	EM34	
507	001462	066604	EM1000	
508	001464	061512	DT016	
509	001466	061766	DF016	
510			ERROR 20:	ATTEMPTING TO CLEAR DATA LATE - CS1 INCORRECT
511	001470	065705	EM35	
512	001472	066604	EM1000	
513	001474	061512	DT016	
514	001476	061766	DF016	
515			ERROR 21:	ATTEMPTING TO CLEAR DATA LATE - CS2 INCORRECT
516	001500	065705	EM35	
517	001502	066714	EM1004	
518	001504	061512	DT016	
519	001506	061766	DF016	
520			ERROR 22:	ATTEMPTING TO READ SILO CONTAINING ONE WORD CS1 INCORRECT
521				
522	001510	065743	EM36	
523	001512	066604	EM1000	
524	001514	061512	DT016	
525	001516	061766	DF016	
526			ERROR 23:	ATTEMPTING TO READ SILO CONTAINING ONE WORD CS2 INCORRECT
527				
528	001520	065743	EM36	
529	001522	066714	EM1004	
530	001524	061512	DT016	
531	001526	061766	DF016	

532			;	ERROR 24:	ATTEMPTING TO LOAD SILO - CS1 INCORRECT
533	001530	066017		EM37	
534	001532	066604		EM1000	
535	001534	061526		DT024	
536	001536	062012		DF024	
537			;	ERROR 25:	ATTEMPTING TO LOAD SILO - CS2 INCORRECT
538	001540	066017		EM37	
539	001542	066714		EM1004	
540	001544	061526		DT024	
541	001546	062012		DF024	
542			;	ERROR 26:	ATTEMPTING TO READ SILO - CS1 INCORRECT
543	001550	066047		EM38	
544	001552	066604		EM1000	
545	001554	061544		DT026	
546				DF026	
547	001556	062036			
548			;	ERROR 27:	ATTEMPTING TO READ SILO - CS2 INCORRECT
549	001560	066047		EM38	
550	001562	066714		EM1004	
551	001564	061544		DT026	
552	001566	062036		DF026	
553			;	ERROR 30:	ATTEMPTING TO READ SILO - DATA INCORRECT
554	001570	066047		EM38	
555	001572	067054		EM1008	
556	001574	061544		DT026	
557	001576	062036		DF026	
558			;	ERROR 31:	ATTEMPTING TO LOAD FULL SILO - CS2 INCORRECT
559	001600	066077		EM39	
560	001602	066714		EM1004	
561	001604	061512		DT016	
562	001606	061766		DF016	
563			;	ERROR 32:	ATTEMPTING TO LOAD FULL SILO - CS1 INCORRECT
564	001610	065324		EM29	
565	001612	066604		EM1000	
566	001614	061512		DT016	
567	001616	061766		DF016	
568			;	ERROR 33:	DATA LATE DID NOT CAUSE EXPECTED INTERRUPT
569	001620	066147		EM40	
570	001622	000000		0	
571	001624	061470		DT005	
572	001626	061706		DF005	
573			;	ERROR 34:	UNEXPECTED INTERRUPT DUE TO UNCLEARED CONTROLLER ERROR
574	001630	066215		EM41	
575	001632	000000		0	
576	001634	061470		DT005	
577	001636	061706		DF005	
578			;	ERROR 35:	CONTROLLER CLEAR DID NOT CLEAR PENDING INTERRUPT DUE TO CONTROLLER ERROR
579			;		
580	001640	066277		EM42	
581	001642	000000		0	
582	001644	061470		DT005	
583	001646	061706		DF005	
584			;	ERROR 36:	CONTROLLER ERROR CAUSED INTERRUPT WITH INTERRUPT ENABLE RESET
585			;		
586	001650	066404		EM43	
587	001652	000000		0	

588	001654	061470	D1005	
589	001656	061706	DF005	
590			ERROR 37:	INTERRUPT DID NOT OCCUR WHEN INTERRUPT ENABLE
591				SET WITH INTERRUPT PENDING DUE TO DATA LATE
592	001660	066470	EM44	
593	001662	062261	DH000A	
594	001664	061474	DT006	
595	001666	061722	DF006	
596			ERROR 40:	UNEXPECTED MEMORY PARITY ENABLE TRAP
597	001670	063214	EM000	
598	001672	062312	DH000C	
599	001674	061422	DT000	
600	001676	061566	DF000	

```

601          .SBTTL  TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
602
603 001700 000000 T.CS1: .WORD 0 ;CONTROL AND STATUS REGISTER 1
604 001702 000000 T.WC: .WORD 0 ;WORD COUNT REGISTER
605 001704 000000 T.BA: .WORD 0 ;BUS ADDRESS REGISTER
606 001706 000000 T.DA: .WORD 0 ;DESIRED TRACK SECTOR REGISTER
607 001710 000000 T.CS2: .WORD 0 ;CONTROL AND STATUS REGISTER 2
608 001712 000000 T.DS: .WORD 0 ;DRIVE STATUS REGISTER
609 001714 000000 T.ER: .WORD 0 ;ERROR REGISTER
610 001716 000000 T.ASOF: .WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
611 001720 000000 T.DCYL: .WORD 0 ;DESIRED CYLINDER REGISTER
612 001722 000000 T.DB: .WORD 0 ;DATA BUFFER
613 001724 000000 T.MR1: .WORD 0 ;MAINTENANCE REGISTER 1
614 001726 000000 T.MR2: .WORD 0 ;MAINTENANCE REGISTER 2
615 001730 000000 T.MR3: .WORD 0 ;MAINTENANCE REGISTER 3
616 001732 000000 T.ECPS: .WORD 0 ;ECC POSITION INFORMATION
617 001734 000000 T.ECPT: .WORD 0 ;ECC PATTERN INFORMATION
618 001736 000000 T.SPAR: .WORD 0 ;SPARE REGISTER
619
620          .SBTTL  EXPECTED RK611 CONTROLLER REGISTERS
621
622 001740 000000 E.CS1: .WORD 0 ;CONTROL AND STATUS REGISTER 1
623 001742 000000 E.WC: .WORD 0 ;WORD COUNT REGISTER
624 001744 000000 E.BA: .WORD 0 ;BUS ADDRESS REGISTER
625 001746 000000 E.DA: .WORD 0 ;DESIRED TRACK SECTOR REGISTER
626 001750 000000 E.CS2: .WORD 0 ;CONTROL AND STATUS REGISTER 2
627 001752 000000 E.DS: .WORD 0 ;DRIVE STATUS REGISTER
628 001754 000000 E.ER: .WORD 0 ;ERROR REGISTER
629 001756 000000 E.ASOF: .WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
630 001760 000000 E.DCYL: .WORD 0 ;DESIRED CYLINDER REGISTER
631 001762 000000 E.DB: .WORD 0 ;DATA BUFFER
632 001764 000000 E.MR1: .WORD 0 ;MAINTENANCE REGISTER 1
633 001766 000000 E.MR2: .WORD 0 ;MAINTENANCE REGISTER 2
634 001770 000000 E.MR3: .WORD 0 ;MAINTENANCE REGISTER 3
635 001772 000000 E.ECPS: .WORD 0 ;ECC POSITION INFORMATION
636 001774 000000 E.ECPT: .WORD 0 ;ECC PATTERN INFORMATION
637 001776 000000 E.SPAR: .WORD 0 ;SPARE REGISTER
638
639          .SBTTL  PROGRAM DEFINED VARIABLES
640
641 002000 000210 RKVEC: .WORD 210 ;RK611 VECTOR ADDRESS
642 002002 000240 RKPRI: .WORD PR5 ;RK611 PRIORITY
643 002004 000000 SRTFLG: .WORD 0 ;START FLAG
644          ; 0 = 200
645          ; 1 = 214
646          ; -1 = 204
647 002006 000000 ERRCNT: .WORD 0 ;ERROR COUNT FOR SWITCH 12 ABORT
648 002010 000000 CONFIG: .WORD 0 ;DATA WRITTEN IN REGISTER
649 002012 000000 CONFG1: .WORD 0 ;DATA GENERATION WORD
650 002014 000000 PREREG: .WORD 0 ;PREVIOUS REGISTER CONTENTS
651 002016 000000 SAVFLG: .WORD 0 ;TRAP CATCHER SAVED
652 002020 000000 PRIOR: .WORD 0 ;PROCESSOR PRIORITY
653 002022 000000 SILCNT: .WORD 0 ;NUMBER OF WORDS READ OR WRITTEN IN SILO
654 002024 000000 TRAPPC: .WORD 0 ;TRAP PC FOR UNEXPECTED MAIN MEMORY CHECK
655 002026 000000 SAVSWR: .WORD 0 ;RESTORE SWITCH REGISTER

```

```

656          .SBTTL PROGRAM SETUP
657
658 002030 012737 000001 002004 PARM: MOV #1,SRTFLG ;LOAD START FLAG FOR PARMETER START
659 002036 000406                BR      START1
660
661 002040 012737 177777 002004 RESTRT: MOV #-1,SRTFLG ;LOAD START FLAG FOR RESTART
662 002046 000402                BR      START1
663
664 002050 005037 002004          START: CLR SRTFLG ;CLEAR START FLAG
665 002054 000005          START1: RESET ;RESET THE WHOLE SYSTEM
666 002056 012706 001100          MOV #STACK,SP ;INITIALIZE STACK POINTER
667 002062 012746 000340          MOV #PR7,-(SP) ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
668 002066 012746 002074          MOV #1$,-(SP) ;LOAD START OF PROGRAM
669 002072 000002                RTI ;LOAD PSW
670
671 002074          1$:
672          .SBTTL INITIALIZE THE COMMON TAGS
673          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
674 002074 012706 001100          MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
675 002100 005026          CLR (R6)+ ;;CLEAR MEMORY LOCATION
676 002102 022706 001140          CMP #SWR,R6 ;;DONE?
677 002106 001374          BNE -.6 ;;LOOP BACK IF NO
678 002110 012706 001100          MOV #STACK,SP ;;SETUP THE STACK POINTER
679          ;;INITIALIZE A FEW VECTORS
680 002114 012737 055426 000020          MOV #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
681 002122 012737 000340 000022          MOV #340,@IOTVEC+2 ;;LEVEL 7
682 002130 012737 056426 000030          MOV #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
683 002136 012737 000340 000032          MOV #340,@EMTVEC+2 ;;LEVEL 7
684 002144 012737 061332 000034          MOV #TRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
685 002152 012737 000340 000036          MOV #340,@TRAPVEC+2;LEVEL 7
686 002160 012737 061202 000024          MOV #SPWRDN,@PWRVEC ;;POWER FAILURE VECTOR
687 002166 012737 000340 000026          MOV #340,@PWRVEC+2 ;;LEVEL 7
688 002174 013737 055120 055112          MOV SENDCT,SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
689 002202 005037 001200          CLR STIMES ;;INITIALIZE NUMBER OF ITERATIONS
690 002206 005037 001202          CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
691 002212 112737 000001 001115          MOV #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
692 002220 012737 002220 001106          MOV #.,$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
693 002226 012737 002226 001110          MOV #.,$LPERR ;;SETUP THE ERROR LOOP ADDRESS
694          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
695          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
696 002234 013746 000004          MOV @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
697 002240 012737 002274 000004          MOV #64,$ERRVEC ;;SET UP ERROR VECTOR
698 002246 012737 177570 001140          MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
699 002254 012737 177570 001142          MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPIAY REGISTER
700 002262 022777 177777 176650          CMP #-1,$SWR ;;TRY TO REFERENCE HARDWARE SWR
701 002270 001012          BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
702          ;;AND THE HARDWARE SWR IS NOT = -1
703          BR 65$ ;;BRANCH IF NO TIMEOUT
704 002274 012716 002302          64$: MOV #65$, (SP) ;;SET UP FOR TRAP RETURN
705 002300 000002          RTI
706 002300 012737 000176 001140          65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
707 002310 012737 000174 001142          MOV #DISPREG,DISPLAY
708 002316 012637 000004          66$: MOV (SP)+,@ERRVEC ;;RESTORE ERROR VECTOR
709
710 002322 005037 001222          CLR $PASS ;;CLEAR PASS COUNT
711 002326 132737 000200 001235          BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
    
```

```

712 002334 001403          BEQ    67$      ;;YES,USE NON-APT SWITCH
713 002336 012737 001236 001140  MOV    #$$SWREG,SWR  ;;NO,USE APT SWITCH REGISTER
714 002344                67$:
715 002344 005037 002006          CLR    ERRCNT      ;CLEAR ERROR COUNT
716                .SBTTL  TYPE PROGRAM NAME
717                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
718 002350 005227 177777          INC    #-1         ;;FIRST TIME?
719 002354 001063          BNE    68$      ;;BRANCH IF NO
720 002356 022737 055254 000042  CMP    #SENDAD,2#42  ;;ACT-11?
721 002364 001457          BEQ    68$      ;;BRANCH IF YES
722 002366 104401 002434          TYPE   69$      ;;TYPE ASCIZ STRING
723                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
724 002372 005737 000042          TST    2#42        ;;ARE WE RUNNING UNDER XXDP/ACT?
725 002376 001012          BNE    70$      ;;BRANCH IF YES
726 002400 123727 001234 000001  CMPB   $ENV,#1      ;;ARE WE RUNNING UNDER APT?
727 002406 001406          BEQ    70$      ;;BRANCH IF YES
728 002410 023727 001140 000176  CMP    SWR,#SWREG   ;;SOFTWARE SWITCH REG SELECTED?
729 002416 001005          BNE    71$      ;;BRANCH IF NO
730 002420 104406          GTSWR                ;;GET SOFT-SWR SETTINGS
731 002422 000403          BR    71$
732 002424 112737 000001 001134 70$:  MOVB   #1,$AUTOB   ;;SET AUTO-MODE INDICATOR
733 002432                71$:
734 002432 000434          BR    68$      ;;GET OVER THE ASCIZ
735                ;;69$:
736 002524                68$:  .ASCIZ <CRLF>/RK611 DISKLESS DIAGNOSTIC: PART 1 MAINDEC-11-DZR6A-B/<CRLF>
737 002524 022737 000001 002004  CMP    #1,SRTFLG   ;CHECK IF PARAMETER START
738 002532 001122          BNE    15$      ;NO,CONTINUE SETUP
739 002534 104401 062062          TYPE  ,OPR001     ;TYPE "RK611 BUS ADDRESS ( ) ="
740 002540 013746 001270          MOV    $BASE,-(SP) ;SAVE $BASE FOR TYPEOUT
741 002544 104402                ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
742 002546 104401 062111          TYPE  ,OPR002
743 002552 104412          RDOCT                ;GET VALUE
744 002554 012637 001160          MOV    (SP)+,$TMPD
745 002560 001407          BEQ    7$      ;CHECK IF <CR>
746 002562 022737 160000 001160  CMP    #160000,$TMPD ;CHECK IF IN I/O PAGE
747 002570 101361          BHI    5$
748 002572 013737 001160 001270 7$:  MOV    $TMPD,$BASE ;LOAD NEW BUS ADDRESS
749 002600 104401 062117          TYPE  ,OPR003     ;TYPE "RK611 VECTOR ADDRESS ( ) ="
750 002604 013746 001264          MOV    $VECT1,-(SP) ;TYPE OUT CURRENT VECTOR ADDRESS
751 002610 042716 160000          BIC    #160000,(SP)
752 002614 104402          TYPOC
753 002616 104401 062111          TYPE  ,OPR002
754 002622 104412          RDOCT                ;GET VALUE
755 002624 012637 001160          MOV    (SP)+,$TMPD
756 002630 001412          BEQ    10$     ;CHECK IF <CR>
757 002632 022737 001000 001160  CMP    #1000,$TMPD ;CHECK IF LEGAL
758 002640 101757          BLOS   7$
759 002642 042737 017777 001264  BIC    #17777,$VECT1 ;LOAD NEW VECTOR ADDRESS
760 002650 053737 001160 001264  BIS    $TMPD,$VECT1
761 002656 104401 062147          10$: TYPE  ,OPR004     ;TYPE "RK611 PRIORITY ( ) ="
762 002662 005046          CLR    -(SP)     ;MAKE ROOM ON THE STACK
763 002664 113716 001265          MOVB   $VECT1+1,(SP)
764 002670 006216          ASR    (SP)      ;SHIFT 5 BITS RIGHT
765 002672 006216          ASR    (SP)
766 002674 006216          ASR    (SP)
767 002676 006216          ASR    (SP)
    
```

768	002700	006216		ASR	(SP)	
769	002702	104402		TYPOC		
770	002704	104401	062111	TYPE	,OPR002	
771	002710	104412		RDOCT		;GET VALUE
772	002712	012637	001160	MOV	(SP)+,\$TMPO	
773	002716	001430		BEQ	15\$;CHECK FOR DEFAULT
774	002720	022737	000007 001160	CMP	#7,\$TMPO	;CHECK IF LEGAL
775	002726	103753		BLO	10\$	
776	002730	022737	000004 001160	CMP	#4,\$TMPO	
777	002736	101347		BHI	10\$	
778	002740	006337	001160	ASL	\$TMPO	;SHIFT 5 BITS LEFT
779	002744	006337	001160	ASL	\$TMPO	
780	002750	006337	001160	ASL	\$TMPO	
781	002754	006337	001160	ASL	\$TMPO	
782	002760	006337	001160	ASL	\$TMPO	
783	002764	042737	160000 001264	BIC	#160000,\$VECT1	;STORE NEW PRIORITY
784	002772	153737	001160 001265	BISB	\$TMPO,\$VECT1+1	
785	003000	013737	001264 002000 15\$:	MOV	\$VECT1,RKVEC	;STORE RK611 VECTOR
786	003006	042737	160000 002000	BIC	#160000,RKVEC	;CLEAR PRIORITY BITS
787	003014	113737	001265 002002	MOV	\$VECT1+1,RKPRI	;STORE PRIORITY
788	003022	005737	002016	TST	SAVFLG	;CHECK IF TRAP CATCHER IS TO BE RESTORED
789	003026	001412		BEQ	NEWPAS	;NO, GO TO FIRST TEST
790	003030	012701	067764	MOV	#SAVVEC,R1	;RESTORE TRAP CATCHER
791	003034	005000		CLR	R0	
792	003036	012703	000400	MOV	#400,R3	;STORE COUNT
793	003042	012120		MOV	(R1)+,(R0)+	
794	003044	005303		DEC	R3	
795	003046	001375		BNE	16\$	
796	003050	005037	002016	CLR	SAVFLG	;INDICATE THAT TRAP CATCHER HAS BEEN RESTORED
797						
798	003054	004737	055274	NEWPAS: JSR	PC,CHKPAR	;CHECK FOR MEMORY CHECK ENABLE
799	003060	012746	000340	MOV	#PR7,-(SP)	;LOCK OUT INTERRUPTS
800	003064	012746	003072	MOV	#TST1,-(SP)	
801	003070	000002		RTI		


```

802
803
804
805
806
807
808
809
810
811
812 003072 000004
813 003074 012737 000764 001200
814 003102 013702 001270
815 003106 012737 003140 000004
816 003114 012737 000340 000006
817 003122 012703 000020
818 003126 005712
819 003130 005722
820 003132 005303
821 003134 001374
822 003136 000405
823
824 003140 062706 000004
825 003144 010237 001122
826 003150 104001
827 003152 012737 000006 000004
828 003160 005037 000006
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847 003164 000004
848 003166 012737 000012 001200
849 003174 013702 001270
850 003200 012737 063336 001310
851 003206 000005
852 003210 004737 055274
853 003214 016237 000000 001126
854 003222 022737 000200 001126
855 003230 001407
856 003232 012737 000200 001124
857 003240 012737 066604 001312

```

```

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

```

```

*****
*ST1: SCOPE
MOV #500, $TIMES ;; DO 500. ITERATIONS
MOV $BASE, R2 ;; LOAD RK611 BASE
MOV #105, $ERRVEC ;; LOAD VECTOR FOR NEM
MOV #PR7, $ERRVEC+2
MOV #16, R3 ;; LOAD NUMBER OF REGISTER
SS: TST (R2) ;; ADDRESS RK611 REGISTER
TST (R2)+ ;; INCREMENT TO NEXT REGISTER
DEC R3 ;; CHECK IF ALL REGISTERS ADDRESS
BNE SS ;; NO CONTINUE
BR 15$ ;; RESTORE TRAP CATCHER

10$: ADD #4, SP ;; ADJUST STACK
MOV R2, $BDADR ;; LOAD ADDRESS PRINT OUT
ERROR 1
15$: MOV $ERRVEC+2, $ERRVEC ;; RESTORE TRAP CATCHER
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.
*****

```

```

*****
*ST2: SCOPE
MOV #10, $TIMES ;; DO 10. ITERATIONS
MOV $BASE, R2 ;; LOAD RK611 BASE REGISTER
MOV #EM2, EM2N ;; LOAD ERROR MESSAGE FOR PRINT OUT
RESET ;; CLEAR RK611 WITH UNIBUS INIT
JSR PC, $CHKPAR ;; CHECK FOR MEMORY CHECK ENABLE
MOV $RCS1(R2), $BDADR ;; SAVE COMMAND AND STATUS REG.7
CMP #RDY, $BDADR ;; CHECK IF CS1 CORRECT
BEQ 1$ ;; YES, CHECK OTHER REGISTERS
MOV #RDY, $GDDAT ;; LOAD EXPECTED CONTENTS
MOV #EM100, EM2N+2 ;; LOAD ERROR MESSAGE

```

C04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 19
 DZR6AB.P11 T2 RESET RK611 AND VERIFY REGISTERS

SEQ 0020

858	003246	104002				ERROR	2	
859	003250	016237	000004	001126	15:	MOV	RKBA(R2), \$BDDAT	;SAVE BUS ADD REG
860	003256	001406				BEQ	25	;CHECK IF ZERO
861	003260	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
862	003264	012737	066647	001312		MOV	#EM1002, EM2N+2	;LOAD ERROR MESSAGE
863	003272	104002				ERROR	2	
864	003274	016237	000006	001126	25:	MOV	RKDA(R2), \$BDDAT	;SAVE DISK ADDRESS ERROR
865	003302	001406				BEQ	35	;CHECK IF ZERO
866	003304	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
867	003310	012737	066671	001312		MOV	#EM1003, EM2N+2	;LOAD ERROR MESSAGE
868	003316	104002				ERROR	2	
869	003320	016237	000016	001126	35:	MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION/OFFSET REG.
870	003326	001406				BEQ	45	;CHECK IF ZERO
871	003330	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
872	003334	012737	067011	001312		MOV	#EM1007, EM2N+2	;LOAD ERROR MESSAGE
873	003342	104002				ERROR	2	
874	003344	016237	000010	001126	45:	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG 2
875	003352	022737	000100	001126		CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
876	003360	001407				BEQ	55	;YES, CONTINUE
877	003362	012737	000100	001124		MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
878	003370	012737	066714	001312		MOV	#EM1004, EM2N+2	;LOAD ERROR MESSAGE
879	003376	104002				ERROR	2	
880	003400	016237	000012	001126	55:	MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
881	003406	001406				BEQ	65	;CHECK IF ZERO
882	003410	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
883	003414	012737	066732	001312		MOV	#EM1005, EM2N+2	;LOAD ERROR MESSAGE
884	003422	104002				ERROR	2	
885	003424	016237	000014	001126	65:	MOV	RKER(2), \$BDDAT	;STORE ERROR STATUS REG
886	003432	001406				BEQ	75	;CHECK IF ZERO
887	003434	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
888	003440	012737	066765	001312		MOV	#EM1006, EM2N+2	;LOAD ERROR MESSAGE
889	003446	104002				ERROR	2	
890	003450	016237	000020	001126	75:	MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
891	003456	001406				BEQ	95	;CHECK IF EQUAL ZERO
892	003460	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
893	003464	012737	067302	001312		MOV	#EM1016, EM2N+2	;LOAD ERROR MESSAGE
894	003472	104002				ERROR	2	
895	003474	016237	000026	001126	95:	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG 1
896	003502	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MR1
897	003510	032737	020000	001126		BIT	#ECCW, \$BDDAT	
898	003516	001403				BEQ	105	
899	003520	052737	020000	001124		BIS	#ECCW, \$GDDAT	
900	003526	023737	001124	001126	105:	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
901	003534	001407				BEQ	115	;YES, CONTINUE TEST
902	003536	012737	022000	001124		MOV	#ECCW!MEWD, \$GDDAT	;LOAD EXPECTED CONTENTS
903	003544	012737	067102	001312		MOV	#EM1009, EM2N+2	;LOAD ERROR MESSAGE
904	003552	104002				ERROR	2	
905	003554	016237	000032	001126	115:	MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
906	003562	001406				BEQ	125	;CHECK IF ZERO
907	003564	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED RESULTS
908	003570	012737	067152	001312		MOV	#EM1013, EM2N+2	;LOAD ERROR MESSAGE
909	003576	104002				ERROR	2	
910	003600	016237	000030	001126	125:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
911	003606	022737	004066	001126		CMP	#4066, \$BDDAT	;CHECK IF ECC POSITION CORRECT
912	003614	001407				BEQ	135	;YES, CONTINUE
913	003616	012737	004066	001124		MOV	#4066, \$GDDAT	;LOAD EXPECTED RESULTS

```

914 003624 012737 067130 001312      MOV      #EM1012,EM2N+2 ;LOAD ERROR MESSAGE
915 003632 104002      ERROR 2
915 003634 016237 000000 001126 13$:  MOV      RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG 1
917 003642 022737 000200 001126      CMP      #RDY, $BDDAT ;CHECK IF CS1 CORRECT
918 003650 001407      BEQ      14$ ;YES, CONTINUE
919 003652 012737 000200 001124      MOV      #RDY, $GDDAT ;LOAD EXPECTED RESULTS
920 003660 012737 067174 001312      MOV      #EM1014,EM2N+2 ;LOAD ERROR MESSAGE
921 003666 104002      ERROR 2
922 003670 016237 000010 001126 14$:  MOV      RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG 2.
923 003676 022737 000100 001126      CMP      #IR, $BDDAT ;CHECK IF CS2 CORRECT
924 003704 001407      BEQ      TST3 ;GO ON TO NEXT TEXT
925 003706 012737 000100 001124      MOV      #IR, $GDDAT ;LOAD EXPECTED RESULTS
926 003714 012737 067237 001312      MOV      #EM1015,EM2N+2 ;LOAD ERROR MESSAGE
927 003722 104002      ERROR 2

```

```

928
929
930 *****
931 *TEST 3          CONTROLLER CLEAR AND VERIFY REGISTERS
932 *
933 *      INITIALIZE THE RK611 CONTROLLER WITH A CONTROLLER
934 *      CLEAR AND READ ALL REGISTER OF THE RK611 REGISTERS
935 *      EXCEPT THE DATA BUFFER AND VERIFY THAT THEY ARE
936 *      CORRECT.  REEXAMINE COMMAND AND STATUS REGISTER 1
937 *      TO MAKE SURE CONTROLLER ERROR DID NOT SET.  REEXAMINE
938 *      COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE
939 *      DID NOT SET.
940 *****

```

```

941 003724 000004      TST3:  SCOPE
942 003726 012737 000144 001200      MOV      #100, $TIMES ;DO 100. ITERATIONS
943 003734 013702 001270      MOV      $BASE, R2 ;LOAD RK611 BASE REGISTER
944 003740 012737 063413 001310      MOV      #EM3, EM2N ;LOAD ERROR MESSAGE FOR PRINT OUT
945 003746 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
946 003754 016237 000000 001126      MOV      RKCS1(R2), $BDDAT ;SAVE COMMAND AND STATUS REG.7
947 003762 022737 000200 001126      CMP      #RDY, $BDDAT ;CHECK IF CS1 CORRECT
948 003770 001407      BEQ      1$ ;YES, CHECK OTHER REGISTERS
949 003772 012737 000200 001124      MOV      #RDY, $GDDAT ;LOAD EXPECTED CONTENTS
950 004000 012737 066604 001312      MOV      #EM1000,EM2N+2 ;LOAD ERROR MESSAGE
951 004006 104002      ERROR 2
952 004010 016237 000004 001126 1$:  MOV      RKBA(R2), $BDDAT ;SAVE BUS ADD REG
953 004016 001406      BEQ      2$ ;CHECK IF ZERO
954 004020 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
955 004024 012737 066647 001312      MOV      #EM1002,EM2N+2 ;LOAD ERROR MESSAGE
956 004032 104002      ERROR 2
957 004034 016237 000006 001126 2$:  MOV      RKDA(R2), $BDDAT ;SAVE DISK ADDRESS ERROR
958 004042 001406      BEQ      3$ ;CHECK IF ZERO
959 004044 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
960 004050 012737 066671 001312      MOV      #EM1003,EM2N+2 ;LOAD ERROR MESSAGE
961 004056 104002      ERROR 2
962 004060 016237 000016 001126 3$:  MOV      RKASOF(R2), $BDDAT ;STORE ATTENTION/OFFSET REG.
963 004066 001406      BEQ      4$ ;CHECK IF ZERO
964 004070 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
965 004074 012737 067011 001312      MOV      #EM1007,EM2N+2 ;LOAD ERROR MESSAGE
966 004102 104002      ERROR 2
967 004104 016237 000010 001126 4$:  MOV      RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG 2
968 004112 022737 000100 001126      CMP      #IR, $BDDAT ;CHECK IF CS2 CORRECT
969 004120 001407      BEQ      5$ ;YES, CONTINUE

```

E04

```

970 004122 012737 000100 001124      MOV      #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
971 004130 012737 066714 001312      MOV      #EM1004,EM2N+2 ;LOAD ERROR MESSAGE
972 004136 104002                ERROR    2
973 004140 016237 000012 001126 5$:      MOV      RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
974 004146 001406                BEQ      6$              ;CHECK IF ZERO
975 004150 005037 001124                CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
976 004154 012737 066732 001312      MOV      #EM1005,EM2N+2 ;LOAD ERROR MESSAGE
977 004162 104002                ERROR    2
978 004164 016237 000014 001126 6$:      MOV      RKER(2),$BDDAT  ;STORE ERROR STATUS REG
979 004172 001406                BEQ      7$              ;CHECK IF ZERO
980 004174 005037 001124                CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
981 004200 012737 066765 001312      MOV      #EM1006,EM2N+2 ;LOAD ERROR MESSAGE
982 004206 104002                ERROR    2
983 004210 016237 000020 001126 7$:      MOV      RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
984 004216 001406                BEQ      9$              ;CHECK IF EQUAL ZERO
985 004220 005037 001124                CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
986 004224 012737 067302 001312      MOV      #EM1016,EM2N+2 ;LOAD ERROR MESSAGE
987 004232 104002                ERROR    2
988 004234 016237 000026 001126 9$:      MOV      RKMRI(R2),$BDDAT ;STORE MAINTENANCE REG 1
989 004242 012737 002000 001124      MOV      #MEWD,$GDDAT   ;LOAD EXPECTED MRI
990 004250 032737 020000 001126      BIT      #ECCW,$BDDAT
991 004256 001403                BEQ      10$
992 004260 052737 020000 001124      BIS      #ECCW,$GDDAT
993 004266 023737 001124 001126 10$:     CMP      $GDDAT,$BDDAT  ;CHECK IF MRI CORRECT
994 004274 001407                BEQ      11$              ;YES,CONTINUE TEST
995 004276 012737 022000 001124      MOV      #ECCW!MEWD,$GDDAT ;LOAD EXPECTED CONTENTS
996 004304 012737 067102 001312      MOV      #EM1009,EM2N+2 ;LOAD ERROR MESSAGE
997 004312 104002                ERROR    2
998 004314 016237 000032 001126 11$:     MOV      RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
999 004322 001406                BEQ      12$              ;CHECK IF ZERO
1000 004324 005037 001124                CLR      $GDDAT          ;LOAD EXPECTED RESULTS
1001 004330 012737 067152 001312      MOV      #EM1013,EM2N+2 ;LOAD ERROR MESSAGE
1002 004336 104002                ERROR    2
1003 004340 016237 000030 001126 12$:     MOV      RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
1004 004346 022737 004066 001126      CMP      #4066,$BDDAT   ;CHECK IF ECC POSITION CORRECT
1005 004354 001407                BEQ      13$              ;YES,CONTINUE
1006 004356 012737 004066 001124      MOV      #4066,$GDDAT   ;LOAD EXPECTED RESULTS
1007 004364 012737 067130 001312      MOV      #EM1012,EM2N+2 ;LOAD ERROR MESSAGE
1008 004372 104002                ERROR    2
1009 004374 016237 000000 001126 13$:     MOV      RKCSI(R2),$BDDAT ;STORE COMMAND AND STATUS REG 1
1010 004402 022737 000200 001126      CMP      #RDY,$BDDAT    ;CHECK IF CS1 CORRECT
1011 004410 001407                BEQ      14$              ;YES,CONTINUE
1012 004412 012737 000200 001124      MOV      #RDY,$GDDAT    ;LOAD EXPECTED RESULTS
1013 004420 012737 067174 001312      MOV      #EM1014,EM2N+2 ;LOAD ERROR MESSAGE
1014 004426 104002                ERROR    2
1015 004430 016237 000010 001126 14$:     MOV      RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG 2.
1016 004436 022737 000100 001126      CMP      #IR,$BDDAT     ;CHECK IF CS2 CORRECT
1017 004444 001407                BEQ      TST4            ;GO ON TO NEXT TEXT
1018 004446 012737 000100 001124      MOV      #IR,$GDDAT     ;LOAD EXPECTED RESULTS
1019 004454 012737 067237 001312      MOV      #EM1015,EM2N+2 ;LOAD ERROR MESSAGE
1020 004462 104002                ERROR    2
1021
1022
1023
1024
1025
*****
*TEST 4          WRITE BUS ADDRESS WITH 177777 (PART 1)
*
*              THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777

```

F04

```

1026          : *      AND CHECK IF EQUAL TO 177776 AND THAT NO REGISTER INTERACTION
1027          : *      TAKES PLACE. A RESET IS DONE AT THE END OF THE TEST TO
1028          : *      MAKE SURE THE BUS ADDRESS CLEARS AND ALL RK611 REGISTERS
1029          : *      ARE IN THEIR INITIALIZED STATE.
1030          : *
1031          : *****
1032 004464 000004          *ST4: SCOPE
1033 004466 012737 000012 001200  MOV      #10, $TIMES      ; DO 10 ITERATIONS
1034 004474 013702 001270 001200  MOV      $BASE, R2      ; LOAD RK611 BASE
1035 004500 012737 177777 002010  MOV      #177777, CONFIG ; LOAD VALUE WRITTEN FOR PRINT OUT
1036 004506 012737 063475 001320  MOV      #EM4, EM3N     ; LOAD ERROR MESSAGE
1037 004514 012762 100000 000000  MOV      #CLR, RKCS1(R2) ; INITIALIZE RK611 CONTROLLER
1038 004522 012762 177777 000004  MOV      #177777, RKBA(R2) ; WRITE BUS ADDRESS WITH 177777
1039 004530 016237 000004 001126  MOV      RKBA(R2), $BDDAT ; STORE BUS ADDRESS REG
1040 004536 022737 177776 001126  CMP      #177776, $BDDAT ; CHECK IF BUS ADDRESS CORRECT
1041 004544 001407          BEQ      1$             ; YES, CHECK IF ANY REGISTER MODIFIED
1042 004546 012737 177776 001124  MOV      #177776, $GDDAT ; LOAD EXPECTED RESULTS
1043 004554 012737 066647 001322  MOV      #EM1002, EM3N+2 ; LOAD ERROR MESSAGE
1044 004562 104003          ERROR      3
1045 004564 016237 000000 001126 1$: MOV      RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG.1
1046 004572 022737 000200 001126  CMP      #RDY, $BDDAT    ; CHECK IF CS1 CORRECT
1047 004600 001407          BEQ      2$             ; YES, CHECK OTHER REGISTERS
1048 004602 012737 000200 001124  MOV      #RDY, $GDDAT    ; LOAD EXPECTED RESULTS
1049 004610 012737 067324 001322  MOV      #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
1050 004616 104003          ERROR      3
1051 004620 016237 000006 001126 2$: MOV      RKDA(R2), $BDDAT ; STORE DISK ADD REG.
1052 004626 001406          BEQ      3$             ; CHECK IF ZERO
1053 004630 005037 001124          CLR      $GDDAT        ; LOAD EXPECTED CONTENTS
1054 004634 012737 067422 001322  MOV      #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
1055 004642 104003          ERROR      3
1056 004644 016237 000016 001126 3$: MOV      RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY/OFFSET REG.
1057 004652 001406          BEQ      4$             ; CHECK IF ZERO
1058 004654 005037 001124          CLR      $GDDAT        ; LOAD EXPECTED CONTENTS
1059 004660 012737 067566 001322  MOV      #EM1024, EM3N+2 ; LOAD ERROR MESSAGE
1060 004666 104003          ERROR      3
1061 004670 016237 000010 001126 4$: MOV      RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG 2
1062 004676 022737 000100 001126  CMP      #IR, $BDDAT    ; CHECK IF CS2 CORRECT
1063 004704 001407          BEQ      5$             ; YES, CONTINUE
1064 004706 012737 000100 001124  MOV      #IR, $GDDAT    ; LOAD EXPECTED CONTENTS
1065 004714 012737 067460 001322  MOV      #EM1021, EM3N+2 ; LOAD ERROR MESSAGE
1066 004722 104003          ERROR      3
1067 004724 016237 000012 001126 5$: MOV      RKDS(R2), $BDDAT ; STORE DRIVE STATUS REG.
1068 004732 001406          BEQ      6$             ; CHECK IF ZERO
1069 004734 005037 001124          CLR      $GDDAT        ; LOAD EXPECTED CONTENTS
1070 004740 012737 067501 001322  MOV      #EM1022, EM3N+2 ; LOAD ERROR MESSAGE
1071 004746 104003          ERROR      3
1072 004750 016237 000014 001126 6$: MOV      RKER(R2), $BDDAT ; STORE ERROR REG.
1073 004756 001406          BEQ      7$             ; CHECK IF ZERO
1074 004760 005037 001124          CLR      $GDDAT        ; LOAD EXPECTED CONTENTS
1075 004764 012737 067537 001322  MOV      #EM1023, EM3N+2 ; LOAD ERROR MESSAGE
1076 004772 104003          ERROR      3
1077 004774 016237 000020 001126 7$: MOV      RKCYL(R2), $BDDAT ; STORE CYLINDER ADD. REG..
1078 005002 001406          BEQ      9$             ; CHECK IF EQUAL ZERO
1079 005004 005037 001124          CLR      $GDDAT        ; LOAD EXPECTED RESULTS
1080 005010 012737 067634 001322  MOV      #EM1025, EM3N+2 ; LOAD ERROR MESSAGE
1081 005016 104003          ERROR      3

```

```

1082 005020 016237 000026 001126 95:  MOV      RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG. 1
1083 005026 012737 002000 001124      MOV      #MEWD, $GDDAT ;LOAD EXPECTED MR1
1084 005034 032737 020000 001126      BIT      #ECCW, $BDDAT
1085 005042 001403      BEQ      105
1086 005044 052737 020000 001124      BIS      #ECCW, $GDDAT
1087 005052 023737 001124 001126 105:  CMP      $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
1088 005060 001404      BEQ      115 ;YES, CONTINUE TEST
1089 005062 012737 067661 001322      MOV      #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
1090 005070 104003      ERROR   3
1091 005072 016237 000032 001126 115:  MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
1092 005100 001406      BEQ      125 ;CHECK IF ZERO
1093 005102 005037 001124      CLR      $GDDAT ;LOAD EXPECTED RESULTS
1094 005106 012737 067737 001322      MOV      #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
1095 005114 104003      ERROR   3
1096 005116 016237 000030 001126 125:  MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REGC
1097 005124 022737 004066 001126      CMP      #4066, $BDDAT ;CHECK IF ECC POSITION CRRECT
1098 005132 001407      BEQ      135 ;YES, CLEAR RK611
1099 005134 012737 004066 001124      MOV      #4066, $GDDAT ;LOAD EXPECTED RESULTS
1100 005142 012737 067712 001322      MOV      #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
1101 005150 104003      ERROR   3
1102 005152 012737 177776 002014 135:  MOV      #177776, PREREG ;LOAD PREVIOUS CONTENTS
1103 005160 012737 063531 001330      MOV      #EM5, EM4N ;LOAD ERROR MESSAGE
1104 005166 000005      RESET   ;CLEAR RK611 WITH UNIBUS INIT
1105 005170 004737 055274      JSR      PC, CHKPAR ;CHECK FOR MEMORY CHECK ENABLE
1106 005174 016237 000004 001126      MOV      RKBA(R2), $BDDAT ;STORE BUS ADDRESS REG.
1107 005202 001406      BEQ      TST5 ;CHECK IF CLEARED
1108 005204 005037 001124      CLR      $GDDAT ;LOAD EXPECTED RESULTS
1109 005210 012737 066647 001332      MOV      #EM1002, EM4N+2 ;LOAD ERROR MESSAGE
1110 005216 104004      ERROR   4

```

```

1111
1112 ;*****
1113 ;*TEST 5 WRITE BUS ADDRESS WITH 177777 (PART 2)
1114 ;*
1115 ;* THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777.
1116 ;* A CONTROLLER CLEAR IS DONE. MAKE SURE THE BUS ADDRESS CLEARS.
1117 ;*

```

```

1118 ;*****
1119 TST5: SCOPE
1120 005220 000004      MOV      #100, $TIMES ;DO 100. ITERATIONS
1121 005222 012737 000144 001200      MOV      $BASE, R2 ;LOAD RK611 BASE
1122 005230 013702 001270      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
1123 005234 012762 100000 000000      MOV      #177777, RKBA(R2) ;LOAD BUS ADDRESS WITH ALL ONES
1124 005242 012762 177777 000004      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
1125 005250 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
1126 005256 016237 000004 001126      MOV      RKBA(R2), $BDDAT ;STORE BUS ADDRESS
1127 005264 001414      BEQ      TST6 ;CHECK IF ZERO
1128 005266 012737 177776 002014      MOV      #177776, PREREG ;LOAD PREVIOUS CONTENTS
1129 005274 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
1130 005300 012737 063610 001330      MOV      #EM6, EM4N ;LOAD ERROR MESSAGE
1131 005306 012737 066647 001332      MOV      #EM1002, EM4N+2
1132 005314 104004      ERROR   4

```

```

1133 ;*****
1134 ;*TEST 6 WRITE WORD COUNT REG. WITH 177777
1135 ;*
1136 ;* THIS TEST WILL WRITE THE WORD COUNT REGISTER
1137 ;* TO 0 AND 177777 AND CHECK IF EQUAL TO 0 AND 177777

```


H04

```

1138      ;*      RESPECTIVELY AND THAT NO REGISTER INTERACTION TAKES PLACE.
1139      ;*
1140      ;*      ISSUE A CONTROLLER CLEAR AND MADE SURE THAT THE WORD
1141      ;*      COUNT REGISTER DOES NOT CHANGE.
1142      ;*
1143      ;*****
1144      005316 000004      ST6:  SCOPE
1145      005320 012737 000764 001200      MOV      #500.,$TIMES      ;;DO 500. ITERATIONS
1146      005326 013702 001270      MOV      $BASE,R2      ;;LOAD RK611 BASE
1147      005332 005037 002010      CLR      CONFIG      ;;CLEAR CONFIGURATION WORD
1148      005336 012737 063673 001320      MOV      #EM7,EM3N      ;;LOAD ERROR MESSAGE
1149      005344 012737 005352 001110      MOV      #1$, $LPERR      ;;LOAD LOOP ON ERROR LOCATION FOR
1150      ;          SUBTEST LOOP
1151
1152      005352      15:
1153      005352 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
1154      005360 013762 002010 000002      MOV      CONFIG,RKWC(R2) ;WRITE WORD COUNT REG.
1155      005366 016237 000002 001126      MOV      RKWC(R2), $BDDAT ;STORE WORD COUNT REG
1156      005374 023737 002010 001126      CMP      CONFIG,$BDDAT   ;CHECK IF WORD COUNT CORRECT
1157      005402 001407      BEQ      2$              ;YES, CHECK IF ANY REGISTER MODIFIED
1158      005404 013737 002010 001124      MOV      CONFIG,$GDDAT   ;LOAD EXPECTED RESULTS
1159      005412 012737 066622 001322      MOV      #EM1001,EM3N+2 ;LOAD ERROR MESSAGE
1160      005420 104003      ERROR    3
1161      005422 016237 000000 001126      2$:  MOV      RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG. 1
1162      005430 022737 000200 001126      CMP      #RDY,$BDDAT     ;CHECK IF CS1 CORRECT
1163      005436 001407      BEQ      3$              ;YES, CONTINUE TEST
1164      005440 012737 000200 001124      MOV      #RDY,$GDDAT     ;LOAD EXPECTED RESULTS
1165      005446 012737 067324 001322      MOV      #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
1166      005454 104003      ERROR    3
1167      005456 016237 000004 001126      3$:  MOV      RKBA(R2), $BDDAT ;STORE BUS ADD REG.
1168      005464 001406      BEQ      4$              ;CHECK IF ZERO
1169      005466 005037 001124      CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
1170      005472 012737 067375 001322      MOV      #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
1171      005500 104003      ERROR    3
1172      005502 016237 000006 001126      4$:  MOV      RKDA(R2), $BDDAT ;STORE DISK ADD REG.
1173      005510 001406      BEQ      5$              ;CHECK IF ZERO
1174      005512 005037 001124      CLR      $GDDAT          ;LOAD EXPECTD CONTENTS
1175      005516 012737 067422 001322      MOV      #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
1176      005524 104003      ERROR    3
1177      005526 016237 000016 001126      5$:  MOV      RKASOF(R2), $BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
1178      005534 001406      BEQ      6$              ;CHECK IF ZERO
1179      005536 005037 001124      CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
1180      005542 012737 067566 001322      MOV      #EM1024,EM3N+2 ;LOAD ERROR MESSAGE
1181      005550 104003      ERROR    3
1182      005552 016237 000010 001126      6$:  MOV      RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG 2
1183      005560 022737 000100 001126      CMP      #IR,$BDDAT      ;CHECK IF CS2 CORRECT
1184      005566 001407      BEQ      7$              ;YES, CONTINUE
1185      005570 012737 000100 001124      MOV      #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
1186      005576 012737 067460 001322      MOV      #EM1021,EM3N+2 ;LOAD ERROR MESSAGE
1187      005604 104003      ERROR    3
1188      005606 016237 000012 001126      7$:  MOV      RKDS(R2), $BDDAT ;STORE DRIVE STATUS REG.
1189      005614 001406      BEQ      8$              ;CHECK IF ZERO
1190      005616 005037 001124      CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
1191      005622 012737 067501 001322      MOV      #EM1022, EM3N+2 ;LOAD ERROR MESSAGE
1192      005630 104003      ERROR    3
1193      005632 016237 000014 001126      8$:  MOV      RKER(R2), $BDDAT ;STORE ERROR REG

```



```

1194 005640 001406          BEQ      9$          ;CHECK IF ZERO
1195 005642 005037 001124    CLR      $GDDAT     ;LOAD EXPECTED CONTENTS
1196 005646 012737 067537 001322  MOV     #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
1197 005654 104003          ERROR    3
1198 005656 016237 000020 001126 9$:  MOV     RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD. REG
1199 005664 001406          BEQ      11$         ;CHECK IF EQUAL ZERO
1200 005666 005037 001124    CLR      $GDDAT     ;LOAD EXPECTED RESULTS
1201 005672 012737 067634 001322  MOV     #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
1202 005700 104003          ERROR    3
1203 005702 016237 000026 001126 11$: MOV     RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG 1
1204 005710 012737 002000 001124    MOV     #MEWD, $GDDAT ;LOAD EXPECTED MRI
1205 005716 032737 020000 001126    BIT     #ECCW, $BDDAT
1206 005724 001403          BEQ      12$         ;CHECK IF MRI CORRECT
1207 005726 052737 020000 001124    BIS     #ECCW, $GDDAT ;YES, CONTINUE TEST
1208 005734 023737 001124 001126 12$: CMP     $GDDAT, $BDDAT
1209 005742 001404          BEQ      13$         ;LOAD ERROR MESSAGE
1210 005744 012737 067661 001322  MOV     #EM1026,EM3N+2
1211 005752 104003          ERROR    3
1212 005754 016237 000032 001126 13$: MOV     RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
1213 005762 001406          BEQ      14$         ;CHECK IF ZERO
1214 005764 005037 001124    CLR      $GDDAT     ;LOAD EXPECTED RESULTS
1215 005770 012737 067737 001322  MOV     #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
1216 005776 104003          ERROR    3
1217 006000 016237 000030 001126 14$: MOV     RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
1218 006006 022737 004066 001126    CMP     #4066, $BDDAT ;CHECK IF ECC POSITION CORRECT
1219 006014 001407          BEQ      15$         ;YES, CLEAR RK611
1220 006016 012737 004066 001124    MOV     #4066, $GDDAT ;LOAD EXPECTED RESULTS
1221 006024 012737 067712 001322  MOV     #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
1222 006032 104003          ERROR    3
1223 006034 012762 100000 000000 15$: MOV     #CCLR, RKCS1(R2) ;CLEAR RK611
1224 006042 016237 000002 001126    MOV     RKWC(R2), $BDDAT ;STORE WORD COUNT REG.
1225 006050 023737 002010 001126    CMP     CONFIG, $BDDAT ;CHECK IF WORD COUNT NOT CHANGED
1226                                     ; BY CONTROLLER CLEAR
1227 006056 001412          BEQ      20$         ;YES, CONTINUE
1228 006060 013737 002010 001124    MOV     CONFIG, $GDDAT ;LOAD EXPECTED DATA
1229 006066 012737 063413 001310    MOV     #EM3, EM2N    ;LOAD ERROR MESSAGE
1230 006074 012737 067345 001312    MOV     #EM1018,EM2N+2
1231 006102 104002          ERROR    2
1232 006104 104415          SCOPI    20$:      ;TEST IF LOOP ON ERROR
1233 006106 005737 002010    TST     CONFIG      ;CHECK IF FINISHED
1234 006112 001005          BNE     TST7        ;;YES, GO TO NEXT TEST
1235 006114 012737 177777 002010    MOV     #177777, CONFIG ;USE 177777 FOR SECOND PASS
1236 006122 000137 005352          JMP     1$

```

```

*****
*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 INTERACTION
*      TAKES PLACE. A CONTROLLER CLEAR IS DONE AT THE END OF
*      THE TEST TO MAKE SURE THE DISK ADDRESS CLEARS.
*
*****

```

```

1246
1247 006126 000004          TST7:  SCOPE
1248 006130 012737 000144 001200    MOV     #100, $TIMES ;;DO 100. ITERATIONS
1249 006136 013702 001270    MOV     $BASE, R2    ;LOAD RK611 BASE

```

1250	006142	012737	177777	002010		MOV	#177777,CONFIG	;LOAD CONFIGURAIION WORD
1251	006150	012737	063732	001320		MOV	#EM8,EM3N	;LOAD ERROR MESSAGE
1252	006156	012762	100000	000000		MOV	#CLR,RKCS1(R2)	;CLEAR RK611 WITH CONTROLLER CLEAR
1253	006164	005062	000002			CLR	RKWC(R2)	;CLEAR WORD COUNT REG
1254	006170	012762	177777	000006		MOV	#177777,RKDA(R2)	;WRITE DISK ADDRESS WITH
1255								177777
1256	006176	016237	000006	001126		MOV	RKDA(R2),SBDDAT	;STORE DISK ADDRESS REG.
1257	006204	022737	003437	001126		CMP	#3437,SBDDAT	;CHECK IF DISK ADDRESS REG CORRECT
1258	006212	001407				BEQ	1\$;YES, CHECK IF AY REGISTER MODIFIED
1259	006214	012737	003437	001124		MOV	#3437,\$GDDAT	;LOAD EXPECTED RESULTS
1260	006222	012737	066671	001322		MOV	#EM1003,EM3N+2	;LOAD ERROR MESSAGE
1261	006230	104003				ERROR	3	
1262	006232	016237	000000	001126	1\$:	MOV	RKCS1(R2),SBDDAT	;STORE COMMAND AND STATUS REG.1
1263	006240	022737	000200	001126		CMP	#RDY,SBDDAT	;CHECK IF CS1 CORRECT
1264	006246	001407				BEQ	2\$;YES, CONTINUE
1265	006250	012737	000200	001124		MOV	#RDY,\$GDDAT	;LOAD EXPECTED RESULTS
1266	006256	012737	067324	001322		MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
1267	006264	104003				ERROR	3	
1268	006266	016237	000004	001126	2\$:	MOV	RKBA(R2),SBDDAT	;STORE BUS ADD REG.
1269	006274	001406				BEQ	3\$;CHECK IF ZERO
1270	006276	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1271	006302	012737	067375	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
1272	006310	104003				ERROR	3	
1273	006312	016237	000002	001126	3\$:	MOV	RKWC(R2),SBDDAT	;STORE WORD COUNT REG.
1274	006320	001406				BEQ	4\$;CHECK IF ZERO
1275	006322	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1276	006326	012737	067345	001322		MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
1277	006334	104003				ERROR	3	
1278	006336	016237	000016	001126	4\$:	MOV	RKASOF(R2),SBDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
1279	006344	001406				BEQ	5\$;CHECK IF ZERO
1280	006346	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1281	006352	012737	067566	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
1282	006360	104003				ERROR	3	
1283	006362	016237	000010	001126	5\$:	MOV	RKCS2(R2),SBDDAT	;STORE COMMAND AND STATUS REG.2
1284	006370	022737	000100	001126		CMP	#IR,SBDDAT	;CHECK IF CS2 CORRECT
1285	006376	001407				BEQ	6\$;YES, CONTINUE
1286	006400	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
1287	006406	012737	067460	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
1288	006414	104003				ERROR	3	
1289	006416	016237	000012	001126	6\$:	MOV	RKDS(R2),SBDDAT	;STORE DRIVE STAUUS REG.
1290	006424	001406				BEQ	7\$;CHECK IF ZERO
1291	006426	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1292	006432	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
1293	006440	104003				ERROR	3	
1294	006442	016237	000014	001126	7\$:	MOV	RKER(R2),SBDDAT	;STORE ERRPR REG.
1295	006450	001406				BEQ	8\$;CHECK IF ZERO
1296	006452	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1297	006456	012737	067537	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
1298	006464	104003				ERROR	3	
1299	006466	016237	000020	001126	8\$:	MOV	RKDCYL(R2),SBDDAT	;STORE CYLINDER ADD REG
1300	006474	001406				BEQ	10\$;CHECK IF EQUAL ZERO
1301	006476	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED RESULTS
1302	006502	012737	067634	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
1303	006510	104003				ERROR	3	
1304	006512	016237	000026	001126	10\$:	MOV	RKMR1(R2),SBDDAT	;STORE MAINTENANCE REG 1
1305	006520	012737	002000	001124		MOV	#MEWD,\$GDDAT	;LOAD EXPECTED MRI

```

1306 006526 032737 020000 001126 BIT #ECCW,$BDDAT
1307 006534 001403 BEQ 11$
1308 006536 052737 020000 001124 BIS #ECCW,$GDDAT
1309 006544 023737 001124 001126 11$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
1310 006552 001404 BEQ 12$ ;YES, CONTINUE TEST
1311 006554 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
1312 006562 104003 ERROR 3
1313 006564 016237 000032 001126 12$: MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
1314 006572 001406 BEQ 13$ ;CHECK IF ZERO
1315 006574 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
1316 006600 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
1317 006606 104003 ERROR 3
1318 006610 016237 000030 001126 13$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
1319 006616 022737 004066 001126 CMP #4066,$BDDAT ;CHECK IF ECC POSITION CORRECT
1320 006624 001407 BEQ 14$ ;YES, ISSUE CONTROLLER CLEAR
1321 006626 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED RESULTS
1322 006634 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
1323 006642 104003 ERROR 3
1324 006644 012762 100000 000000 14$: MOV #CLR,RKCS1(R2) ;CLEAR RK611
1325 006652 016237 000006 001126 MOV RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG.
1326 006660 001414 BEQ TST10 ;;YES, GO ON TO NEXT TEST
1327 006662 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
1328 006666 012737 063413 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
1329 006674 012737 066671 001332 MOV #EM1003,EM4N+2
1330 006702 012737 003437 002014 MOV #3437,PREREG ;LOAD PREVIOUS VALUE
1331 006710 104004 ERROR 4

```

```

1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361

```

```

;*****
;SBTTL **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 CONTROLLER
; CLEAR TO HE 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.
;
; 000000 000010 000200 004000 100000
; 000001 000020 000400 010000
; 000002 000040 001000 020000
; 000004 000100 002000 040000
;

```

```

1362
1363 006712 000004
1364 006714 012737 000144 001200
1365 006722 012701 000021
1366 006726 012737 000001 002010
1367 006734 012737 063475 001320
1368 006742 012737 100000 000000
1369 006750 012737 006756 00111C
1370
1371
1372 006756
1373 006756 005062 000002
1374 006762 013762 002010 000004
1375 006770 016237 000004 001126
1376 006776 013737 002010 001124
1377 007004 042737 000001 001124
1378 007012 023737 001124 001126
1379 007020 001404
1380 007022 012737 066647 001322
1381 007030 104003
1382 007032
1383 007032 016237 000000 001126
1384 007040 022737 000200 001126
1385 007046 001407
1386 007050 012737 000100 001124
1387 007056 012737 067324 001322
1388 007064 104003
1389 007066
1390 007066 016237 000002 001126
1391 007074 001406
1392 007076 005037 001124
1393 007102 012737 067345 001322
1394 007110 104003
1395 007112
1396 007112 016237 000006 001126
1397 007120 001406
1398 007122 005037 001124
1399 007126 012737 067422 001322
1400 007134 104003
1401 007136
1402 007136 016237 000016 001126
1403 007144 001406
1404 007146 005037 001124
1405 007152 012737 067566 001322
1406 007160 104003
1407 007162
1408 007162 016237 000010 001126
1409 007170 022737 000100 001126
1410 007176 001407
1411 007200 012737 000100 001124
1412 007206 012737 067460 001322
1413 007214 104003
1414 007216
1415 007216 016237 000012 001126
1416 007224 001406
1417 007226 005037 001124

```

```

*****
TST10: SCOPE
MOV #100,$TIMES ;DO 100. ITERATIONS
MOV #17,R1 ;LOAD NUMBER OF PATTERNS
MOV #000001,CONFIG ;LOAD INITIAL CONFIGURATION
MOV #EM4,EM3N ;LOAD ERROR MESSAGE
MOV #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #15,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1$:
CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV CONFIG,RKBA(R2) ;WRITE RKBA
MOV RKBA(R2),$BDDAT ;STORE RKBA
MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
BIC #1,$GDDAT ;INITIALIZE READ ONLY BITS
CMP $GDDAT,$BDDAT ;CHECK IF RKBA CORRECT
BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1002,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 #IR,$GDDAT ;LOAD EXPECTED RESULTS
MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

3$:
MOV RKWC(R2),$BDDAT ;STORE WORD COUNT REG
BEQ 5$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

5$:
MOV RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
BEQ 6$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

6$:
MOV RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
BEQ 7$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1024,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

7$:
MOV RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
CMP #IR,$BDDAT ;CHECK IF CS2 CORRECT
BEQ 8$ ;YES, CONTINUE
MOV #IR,$GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1021,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

8$:
MOV RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
BEQ 9$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS

```

M04

```

1418 007232 012737 067501 001322      MOV      #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
1419 007240 104003      ERROR    3
1420 007242 016237 000014 001126 9$:     MOV      RKER(R2),SBDDAT ;STORE ERROR REG
1421 007250 001406      BEQ      10$           ;CHECK IF ZERO
1422 007252 005037 001124      CLR      $GDDAT        ;LOAD EXPECTED CONTENTS
1423 007256 012737 067537 001322      MOV      #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
1424 007264 104003      ERROR    3
1425 007266      10$:
1426 007266 016237 000020 001126      MOV      RKDCYL(R2),SBDDAT ;STORE CYLINDER ADD REG
1427 007274 001406      BEQ      12$           ;CHECK IF ZERO
1428 007276 005037 001124      CLR      $GDDAT        ;LOAD EXPECTED CONTENTS
1429 007302 012737 067634 001322      MOV      #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
1430 007310 104003      ERROR    3
1431 007312      12$:
1432 007312 016237 000026 001126      MOV      RKMRI(R2),SBDDAT ;STORE MAINTENANCE REG.1
1433 007320 012737 002000 001124      MOV      #MEWD,$GDDAT    ;LOAD EXPECTED MRI
1434 007326 032737 020000 001126      BIT      #ECCW,$BDDAT
1435 007334 001403      BEQ      13$
1436 007336 052737 020000 001124      BIS      #ECCW,$GDDAT
1437 007344 023737 001124 001126 13$:     CMP      $GDDAT,$BDDAT ;CHECK IF MR: CORRECT
1438 007352 001404      BEQ      14$           ;YES,ISSUE CONTROLLER CLEAR
1439 007354 012737 067661 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
1440 007362 104003      ERROR    3
1441 007364      14$:
1442 007364 016237 000032 001126      MOV      RKECPT(R2),SBDDAT ;STORE ECC PATTERN REG.
1443 007372 001406      BEQ      15$           ;CHECK IF ZERO
1444 007374 005037 001124      CLR      $GDDAT        ;LOAD EXPECTED CONTENTS
1445 007400 012737 067737 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
1446 007406 104003      ERROR    3
1447 007410 016237 000030 001126 15$:     MOV      RKECPS(R2),SBDDAT ;STORE ECC POSITION REG.
1448 007416 012737 004066 001124 16$:     MOV      #4066,$GDDAT    ;USE 4066
1449 007424 023737 001124 001126 17$:     CMP      $GDDAT,$BDDAT ;CHECK IF ECC POSITION CORRECT
1450 007432 001404      BEQ      18$           ;YES,INITIALIZE RK611
1451 007434 012737 067712 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
1452 007442 104003      ERROR    3
1453 007444 016237 000004 002014 18$:     MOV      RKBA(R2),PREREG ;GET PREVIOUS CONTENTS
1454 007452 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
1455 007460 016237 000004 001126      MOV      RKBA(R2),SBDDAT ;GET CURRENT VALUE
1456 007466 005037 001124      CLR      $GDDAT        ;LOAD EXPECTED CONTENTS
1457 007472 023737 001124 001126      CMP      $GDDAT,$BDDAT ;CHECK IF RKBA CORRECT
1458 007500 001407      BEQ      19$           ;YES,CHECK IF FINISHED
1459 007502 012737 063413 001330      MOV      #EM3,EM4N      ;LOAD ERROR MESSAGE
1460 007510 012737 066647 001332      MOV      #EM1002,EM4N+2
1461 007516 104004      ERROR    4
1462 007520 104415      19$:     SCOP1          ;CHECK IF LOOP ON ERROR
1463 007522 000241      CLC          ;SHIFT IN ZERO
1464 007524 006137 002010      ROL      CONFIG
1465 007530 005301      DEC      R1           ;CHECK IF FINISHED
1466 007532 001402      BEQ      TST11        ;;YES, GO ON TO NEXT TEST
1467 007534 000137 006756      JMP      1$

```

```

1468
1469 *****
1470 *TEST 11 REGISTER INTERACTION USING BUS ADDRESS (PART 2)
1471 *
1472 * THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
1473 * CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD

```

```

1474          : *      COUNT REGISTER TO 0.
1475          : *
1476          : *      THE TEST ITSELF WILL CONSIST OF WRITING THE
1477          : *      BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
1478          : *      TEST IF BUS ADDRESS IS CORRECT AND THAT NO
1479          : *      REGISTER INTERACTION TAKES PLACE.
1480          : *
1481          : *      177777 177767 177577 173777 077777
1482          : *      177776 177757 177377 167777
1483          : *      177775 177737 176777 157777
1484          : *      177773 177677 175777 137777
1485          : *
1486          : * *****
1487          : * ST11: SCOPE
1488          : *      MOV      #100, $TIMES      ; DO 100. ITERATIONS
1489          : *      MOV      #17, R1          ; LOAD NUMBER OF PATTERNS
1490          : *      MOV      #177776, CONFIG ; LOAD INITIAL CONFIGURATION
1491          : *      MOV      #EM4, EM3N       ; LOAD ERROR MESSAGE
1492          : *      MOV      #CCLR, RKCS1    ; CLEAR RK611 WITH CONTROLLER CLEAR
1493          : *      MOV      #1$, $LPERR   ; LOAD LOOP ON ERROR LOCATION FOR
1494          : *                               ; SUBTEST LOOP
1495
1496          : *
1497          : * 1$:
1498          : *      CLR      RKWC(R2)        ; CLEAR WORD COUNT REG.
1499          : *      MOV      CONFIG, RKBA(R2) ; WRITE RKBA
1500          : *      MOV      RKBA(R2), $BDDAT ; STORE RKBA
1501          : *      MOV      CONFIG, $GDDAT   ; PREPARE EXPECTED RESULTS
1502          : *      BIC      #1, $GDDAT       ; INITIALIZE READ ONLY BITS
1503          : *      CMP      $GDDAT, $BDDAT    ; CHECK IF RKBA CORRECT
1504          : *      BEQ      2$                ; YES, TEST IF ANY OTHER REG MODIFIED
1505          : *      MOV      #EM1002, EM3N+2 ; LOAD ERROR MESSAGE
1506          : *      ERROR   3
1507          : *
1508          : * 2$:
1509          : *      MOV      RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
1510          : *      CMP      #RDY, $BDDAT     ; CHECK IF CS1 CORRECT
1511          : *      BEQ      3$                ; YES, CONTINUE
1512          : *      MOV      #IR, $GDDAT      ; LOAD EXPECTED RESULTS
1513          : *      MOV      #EM1017, EM3N+2   ; LOAD ERROR MESSAGE
1514          : *      ERROR   3
1515          : *
1516          : * 3$:
1517          : *      MOV      RKWC(R2), $BDDAT   ; STORE WORD COUNT REG
1518          : *      BEQ      5$                ; CHECK IF ZERO
1519          : *      CLR      $GDDAT           ; LOAD EXPECTED CONTENTS
1520          : *      MOV      #EM1018, EM3N+2   ; LOAD ERROR MESSAGE
1521          : *      ERROR   3
1522          : *
1523          : * 5$:
1524          : *      MOV      RKDA(R2), $BDDAT   ; STORE DISK AVERAGE REG
1525          : *      BEQ      6$                ; CHECK IF ZERO
1526          : *      CLR      $GDDAT           ; LOAD EXPECTED CONTENTS
1527          : *      MOV      #EM1020, EM3N+2   ; LOAD ERROR MESSAGE
1528          : *      ERROR   3
1529          : *
1530          : * 6$:
1531          : *      MOV      RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY/OFFSET REG.
1532          : *      BEQ      7$                ; CHECK IF ZERO
1533          : *      CLR      $GDDAT           ; LOAD EXPECTED CONTENTS
1534          : *      MOV      #EM1024, EM3N+2   ; LOAD ERROR MESSAGE

```


B05

1530	010006	104003			ERROR	3	
1531	010010				75:		
1532	010010	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	: STORE COMMAND AND STATUS REG.2
1533	010016	022737	000100	001126	CMP	#IR, \$BDDAT	: CHECK IF CS2 CORRECT
1534	010024	001407			BEQ	85	: YES, CONTINUE
1535	010026	012737	000100	001124	MOV	#IR, \$GDDAT	: LOAD EXPECTED CONTENTS
1536	010034	012737	067460	001322	MOV	#EM1021, EM3N+2	: LOAD ERROR MESSAGE
1537	010042	104003			ERROR	3	
1538	010044				85:		
1539	010044	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	: STORE DRIVE STATUS REG
1540	010052	001406			BEQ	95	: CHECK IF ZERO
1541	010054	005037	001124		CLR	\$GDDAT	: LOAD EXPECTED CONTENTS
1542	010060	012737	067501	001322	MOV	#EM1022, EM3N+2	: LOAD ERROR MESSAGE
1543	010066	104003			ERROR	3	
1544	010070	16237	000014	001126	95:		
1545	010076	01406			MOV	RKER(R2), \$BDDAT	: STORE ERROR REG
1546	010100	005037	001124		BEQ	105	: CHECK IF ZERO
1547	010104	012737	067537	001322	CLR	\$GDDAT	: LOAD EXPECTED CONTENTS
1548	010112	012737	067537	001322	MOV	#EM1023, EM3N+2	: LOAD ERROR MESSAGE
1549	010112	104003			ERROR	3	
1550	010114				105:		
1551	010114	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	: STORE CYLINDER ADD REG
1552	010122	001406			BEQ	125	: CHECK IF ZERO
1553	010124	005037	001124		CLR	\$GDDAT	: LOAD EXPECTED CONTENTS
1554	010130	012737	067634	001322	MOV	#EM1025, EM3N+2	: LOAD ERROR MESSAGE
1555	010136	104003			ERROR	3	
1556	010140				125:		
1557	010140	016237	000026	001126	MOV	RKMRI(R2), \$BDDAT	: STORE MAINTENANCE REG.1
1558	010146	012737	002000	001124	MOV	#MEWD, \$GDDAT	: LOAD EXPECTED MRI
1559	010154	032737	020000	001126	BIT	#ECCW, \$BDDAT	
1560	010162	001403			BEQ	135	
1561	010164	052737	020000	001124	BIS	#ECCW, \$GDDAT	
1562	010172	023737	001124	001126	135:		
1563	010200	001404			CMP	\$GDDAT, \$BDDAT	: CHECK IF MRI CORRECT
1564	010202	012737	067661	001322	BEQ	145	: YES, ISSUE CONTROLLER CLEAR
1565	010210	104003			MOV	#EM1026, EM3N+2	: LOAD ERROR MESSAGE
1566	010212				ERROR	3	
1567	010212	016237	000032	001126	145:		
1568	010220	001406			MOV	RKECPT(R2), \$BDDAT	: STORE ECC PATTERN REG.
1569	010222	005037	001124		BEQ	155	: CHECK IF ZERO
1570	010226	012737	067737	001322	CLR	\$GDDAT	: LOAD EXPECTED CONTENTS
1571	010234	104003			MOV	#EM1030, EM3N+2	: LOAD ERROR MESSAGE
1572	010236	016237	000030	001126	155:		
1573	010244	012737	004066	001124	165:		
1574	010252	023737	001124	001126	175:		
1575	010260	001404			MOV	RKECPS(R2), \$BDDAT	: STORE ECC POSITION REG.
1576	010262	012737	067712	001322	MOV	#4066, \$GDDAT	: USE 4066
1577	010270	104003			CMP	\$GDDAT, \$BDDAT	: CHECK IF ECC POSITION CORRECT
1578	010272	016237	000004	002014	185:		
1579	010300	012762	!00000	000000	BEQ	185	: YES, INITIALIZE RK611
1580	010306	016237	000004	001126	MOV	#EM1029, EM3N+2	: LOAD ERROR MESSAGE
1581	010314	005037	001124		ERROR	3	
1582	010320	023737	001124	001126	185:		
1583	010326	001407			MOV	RKBA(R2), PREREG	: GET PREVIOUS CONTENTS
1584	010330	012737	063413	001330	MOV	#CCLR, RKCS1(R2)	: CLEAR RK611 CONTROLLER
1585	010336	012737	066647	001332	MOV	RKBA(R2), \$BDDAT	: GET CURRENT VALUE
1586	010344	!04004			CLR	\$GDDAT	: LOAD EXPECTED CONTENTS
					CMP	\$GDDAT, \$BDDAT	: CHECK IF RKBA CORRECT
					BEQ	195	: YES, CHECK IF FINISHED
					MOV	#EM3, EM4N	: LOAD ERROR MESSAGE
					MOV	#EM1002, EM4N+2	
					ERROR	4	

C05

```

1586 010346 104415 :95: SLOPI ;CHECK IF LOOP ON ERROR
1587 010350 000261 SEC ;SHIFT IN ONE
1588 010352 006137 002010 ROL CONFIG
1589 010356 005301 DEC R1 ;CHECK IF FINISHED
1590 010360 001402 BEQ *ST12 ;:YES, GO ON TO NEXT TEST
1591 010362 000137 007604 JMP IS

```

```

*****
:TEST 12 REGISTER INTERACTION USING BUS ADDRESS (PART 3)

```

```

: THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
: CLEAR TO THE PK611 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.

```

```

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

```

```

*****
:ST12: SCOPE

```

```

1611 010366 000004 : DO 100. ITERATIONS
1612 010370 012737 000144 001200 MOV #100, $TIMES ;DO 100. ITERATIONS
1613 010376 012701 000021 MOV #17, R1 ;LOAD NUMBER OF PATTERNS
1614 010402 005037 002010 CLR CONFIG ;LOAD INITIAL CONFIGURATION
1615 010406 012737 063475 001320 MOV #EM4, EM3N ;LOAD ERROR MESSAGE
1616 010414 012737 100000 000000 MOV #CCLA, RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
1617 010422 012737 010430 001110 MOV #IS, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

```

```

1620 010430 : IS:
1621 010430 005062 000002 CLR RKWC(R2) ;CLEAR WORD COUNT REG.
1622 010434 013762 002010 000004 MOV CONFIG, RKBA(R2) ;WRITE RKBA
1623 010442 016237 000004 001126 MOV RKBA(R2), $BDDAT ;STORE RKBA
1624 010450 013737 002010 001124 MOV CONFIG, $GDDAT ;PREPARE EXPECTED RESULTS
1625 010456 042737 000001 001124 BIC #1, $GDDAT ;INITIALIZE READ ONLY BITS
1626 010464 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKBA CORRECT
1627 010472 001404 BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
1628 010474 012737 066647 001322 MOV #EM1002, EM3N+2 ;LOAD ERROR MESSAGE
1629 010502 104003 ERROR 3

```

```

1630 010504 : 2$:
1631 010504 016237 000000 001126 MOV RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG. 1
1632 010512 022737 000200 001126 CMP #RDY, $BDDAT ;CHECK IF CS1 CORRECT
1633 010520 001407 BEQ 3$ ;YES, CONTINUE
1634 010522 012737 000100 001124 MOV #IR, $GDDAT ;LOAD EXPECTED RESULTS
1635 010530 012737 067324 001322 MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE
1636 010536 104003 ERROR 3

```

```

1637 010540 : 3$:
1638 010540 016237 000002 001126 MOV RKWC(R2), $BDDAT ;STORE WORD COUNT REG
1639 010546 001406 BEQ 5$ ;CHECK IF ZERO
1640 010550 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
1641 010554 012737 067345 001322 MOV #EM1018, EM3N+2 ;LOAD ERROR MESSAGE

```

1642	010562	104003					ERROR	3	
1643	010564				55:				
1644	010564	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	; STORE DISK AVERAGE REG	
1645	010572	001406				BEQ	65	; CHECK IF ZERO	
1646	010574	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
1647	010600	012737	067422	001322		MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE	
1648	010606	104003				ERROR	3		
1649	010610				65:				
1650	010610	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG.	
1651	010616	001406				BEQ	75	; CHECK IF ZERO	
1652	010620	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
1653	010624	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE	
1654	010632	104003				ERROR	3		
1655	010634				75:				
1656	010634	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG.2	
1657	010642	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT	
1658	010650	001407				BEQ	85	; YES, CONTINUE	
1659	010652	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS	
1660	010660	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE	
1661	010666	104003				ERROR	3		
1662	010670				85:				
1663	010670	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG	
1664	010676	001406				BEQ	95	; CHECK IF ZERO	
1665	010700	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
1666	010704	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE	
1667	010712	104003				ERROR	3		
1668	010714	016237	000014	001126	95:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG	
1669	010722	001406				BEQ	105	; CHECK IF ZERO	
1670	010724	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
1671	010730	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE	
1672	010736	104003				ERROR	3		
1673	010740				105:				
1674	010740	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG	
1675	010746	001406				BEQ	125	; CHECK IF ZERO	
1676	010750	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
1677	010754	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE	
1678	010762	104003				ERROR	3		
1679	010764				125:				
1680	010764	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG.1	
1681	010772	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MR1	
1682	011000	032737	020000	001126		BIT	#ECCW, \$BDDAT		
1683	011006	001403				BEQ	135		
1684	011010	052737	020000	001124		BIS	#ECCW, \$GDDAT		
1685	011016	023737	001124	001126	135:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT	
1686	011024	001404				BEQ	145	; YES, ISSUE CONTROLLER CLEAR	
1687	011026	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE	
1688	011034	104003				ERROR	3		
1689	011036				145:				
1690	011036	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.	
1691	011044	001406				BEQ	155	; CHECK IF ZERO	
1692	011046	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
1693	011052	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE	
1694	011060	104003				ERROR	3		
1695	011062	016237	000030	001126	155:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG.	
1696	011070	012737	004066	001124	165:	MOV	#4066, \$GDDAT	; USE 4066	
1697	011076	023737	001124	001126	175:	CMP	\$GDDAT, \$BDDAT	; CHECK IF ECC POSITION CORRECT	

E05

```

1698 011104 001404          BEQ      18$          ;YES, INITIALIZE RK611
1699 011106 012737 067712 001322  MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
1700 011114 104003          ERROR    3
1701 011116 016237 000004 002014 18$:  MOV      RKBA(R2),PREREG ;GET PREVIOUS CONTENTS
1702 011124 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
1703 011132 016237 000004 001126  MOV      RKBA(R2),SBODAT ;GET CURRENT VALUE
1704 011140 005037 001124          CLR      $GDDAT        ;LOAD EXPECTED CONTENTS
1705 011144 023737 001124 001126  CMP      $GDDAT,$BODAT ;CHECK IF RKBA CORRECT
1706 011152 001407          BEQ      19$          ;YES, CHECK IF FINISHED
1707 011154 012737 063413 001330  MOV      #EM3,EM4N      ;LOAD ERROR MESSAGE
1708 011162 012737 066647 001332  MOV      #EM1002,EM4N+2
1709 011170 104004          ERROR    4
1710 011172 104415          19$:  SCOPE1          ;CHECK IF LOOP ON ERROR
1711 011174 000261          SEC          ;SHIFT IN ONE
1712 011176 006137 002010  ROL      CONFIG
1713 011202 005301          DEC      R1          ;CHECK IF FINISHED
1714 011204 001402          BEQ      TST13       ;;YES, GO ON TO NEXT TEST
1715 011206 000137 010430  JMP      1$
1716
1717
1718 .....*****
1719 *TEST 13 REGISTER INTERACTION USING BUS ADDRESS (PART 4)
1720 *
1721 * THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
1722 * CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
1723 * COUNT REGISTER TO 0.
1724 *
1725 * THE TEST ITSELF WILL CONSIST OF WRITING THE
1726 * BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
1727 * TEST IF BUS ADDRESS IS CORRECT AND THAT NO
1728 * REGISTER INTERACTION TAKES PLACE.
1729 *
1730 * 100000 174000 177600 177770 000000
1731 * 140000 176000 177700 177774
1732 * 160000 177000 177740 177776
1733 * 170000 177400 177760 177777
1734 * .....*****
1735 *TST13: SCOPE
1736 011212 000004          MOV      #100,$TIMES ;DO 100. ITERATIONS
1737 011214 012737 000144 001200  MOV      #17,R1      ;LOAD NUMBER OF PATTERNS
1738 011222 012701 000021          MOV      #17,R1      ;LOAD INITIAL CONFIGURATION
1739 011226 005037 002010          CLR      CONFIG      ;LOAD ERROR MESSAGE
1740 011232 012737 063475 001320  MOV      #EM4,EM3N    ;CLEAR RK611 WITH CONTROLLER CLEAR
1741 011240 012737 100000 000000  MOV      #CCLR,RKCS1 ;LOOP ON ERROR LOCATION FOR
1742 011246 012737 011254 001110  MOV      #1$,SLPERR  ;SUBTEST LOOP
1743
1744
1745 011254          1$:  CLR      RKWC(R2)     ;CLEAR WORD COUNT REG.
1746 011260 013762 002010 000004  MOV      CONFIG,RKBA(R2) ;WRITE RKBA
1747 011266 016237 000004 001126  MOV      RKBA(R2),SBODAT ;STORE RKBA
1748 011274 013737 002010 001124  MOV      CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
1749 011302 042737 000001 001124  BIC      #1,$GDDAT    ;INITIALIZE READ ONLY BITS
1750 011310 023737 001124 001126  CMP      $GDDAT,$BODAT ;CHECK IF RKBA CORRECT
1751 011316 001404          BEQ      2$          ;YES, TEST IF ANY OTHER REG MODIFIED
1752 011320 012737 066647 001322  MOV      #EM1002,EM3N+2 ;LOAD ERROR MESSAGE
1753 011326 104003          ERROR    3

```

F05

1754	011330				25:			
1755	011330	016237	000000	001126		MOV	RKCS1(R2), \$BDDAT	; STORE COMMAND AND STATUS REG. 1
1756	011336	022737	000200	001126		CMP	#RDY, \$BDDAT	; CHECK IF CS1 CORRECT
1757	011344	001407				BEQ	35	; YES, CONTINUE
1758	011346	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED RESULTS
1759	011354	012737	067324	001322		MOV	#EM1017, EM3N+2	; LOAD ERROR MESSAGE
1760	011362	104003				ERROR	3	
1761	011364				35:			
1762	011364	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	; STORE WORD COUNT REG
1763	011372	001406				BEQ	55	; CHECK IF ZERO
1764	011374	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1765	011400	012737	067345	001322		MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
1766	011406	104003				ERROR	3	
1767	011410				55:			
1768	011410	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	; STORE DISK AVERAGE REG
1769	011416	001406				BEQ	65	; CHECK IF ZERO
1770	011420	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1771	011424	012737	067422	001322		MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
1772	011432	104003				ERROR	3	
1773	011434				65:			
1774	011434	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG.
1775	011442	001406				BEQ	75	; CHECK IF ZERO
1776	011444	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1777	011450	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
1778	011456	104003				ERROR	3	
1779	011460				75:			
1780	011460	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG.2
1781	011466	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
1782	011474	001407				BEQ	85	; YES, CONTINUE
1783	011476	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
1784	011504	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
1785	011512	104003				ERROR	3	
1786	011514				85:			
1787	011514	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
1788	011522	001406				BEQ	95	; CHECK IF ZERO
1789	011524	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1790	011530	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
1791	011536	104003				ERROR	3	
1792	011540	016237	000014	001126	95:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
1793	011546	001406				BEQ	105	; CHECK IF ZERO
1794	011550	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1795	011554	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
1796	011562	104003				ERROR	3	
1797	011564				105:			
1798	011564	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
1799	011572	001406				BEQ	125	; CHECK IF ZERO
1800	011574	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1801	011600	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
1802	011606	104003				ERROR	3	
1803	011610				125:			
1804	011610	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG.1
1805	011616	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MR1
1806	011624	032737	020000	001126		BIT	#ECCW, \$BDDAT	
1807	011632	001403				BEQ	135	
1808	011634	052737	020000	001124		BIS	#ECCW, \$GDDAT	
1809	011642	023737	001124	001126	135:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT

G05

1810	011650	001404			BEQ	14\$;YES,ISSUE CONTROLLER CLEAR
1811	011652	012737	067661	001322	MOV	#EM1026,EM3N+2		;LOAD ERROR MESSAGE
1812	011660	104003			ERROR	3		
1813	011662						14\$:	
1814	011662	016237	000032	001126	MOV	RKECPT(R2),SBDDAT		;STORE ECC PATTERN REG.
1815	011670	001406			BEQ	15\$;CHECK IF ZERO
1816	011672	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
1817	011676	012737	067737	001322	MOV	#EM1030,EM3N+2		;LOAD ERROR MESSAGE
1818	011704	104003			ERROR	3		
1819	011706	016237	000030	001126	MOV	RKECPS(R2),SBDDAT		;STORE ECC POSITION REG.
1820	011714	012737	004066	001124	MOV	#4066,\$GDDAT		;USE 4066
1821	011722	023737	001124	001126	CMP	\$GDDAT,\$BDDAT		;CHECK IF ECC POSITION CORRECT
1822	011730	001404			BEQ	18\$;YES,INITIALIZE RK611
1823	011732	012737	067712	001322	MOV	#EM1029,EM3N+2		;LOAD ERROR MESSAGE
1824	011740	104003			ERROR	3		
1825	011742	016237	000004	002014	MOV	RKBA(R2),PREREG		;GET PREVIOUS CONTENTS
1826	011750	012762	100000	000000	MOV	#CCLR,RKCS1(R2)		;CLEAR RK611 CONTROLLER
1827	011756	016237	000004	001126	MOV	RKBA(R2),SBDDAT		;GET CURRENT VALUE
1828	011764	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
1829	011770	023737	001124	001126	CMP	\$GDDAT,\$BDDAT		;CHECK IF RKBA CORRECT
1830	011776	001407			BEQ	19\$;YES,CHECK IF FINISHED
1831	012000	012737	063413	001330	MOV	#EM3,EM4N		;LOAD ERROR MESSAGE
1832	012006	012737	066647	001332	MOV	#EM1002,EM4N+2		
1833	012014	104004			ERROR	4		
1834	012016	104415					19\$:	
1835	012020	000261			SCOPI			;CHECK IF LOOP ON ERROR
1836	012022	006037	002010		SEC			;SHIFT IN ONE
1837	012026	005301			ROR	CONFIG		
1838	012030	001402			DEC	R1		;CHECK IF FINISHED
1839	012032	000137	011254		BEQ	TST14		;YES,GO ON TO NEXT TEST
1840					JMP	1\$		

```

*****
*TEST 14 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)
*
* THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
* 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
*
*****

```

1859	012036	000004			TST14:	SCOPE		
1860	012040	012737	000144	001200	MOV	#100,\$TIMES		;DO 100. ITERATIONS
1861	012046	012701	000021		MOV	#17,R1		;LOAD NUMBER OF PATTERNS
1862	012052	005037	002010		CLR	CONFIG		;LOAD INITIAL CONFIGURATION
1863	012056	012737	000001	002012	MOV	#000001,CONFIG1		
1864	012064	012737	064331	001320	MOV	#EM16,EM3N		;LOAD ERROR MESSAGE
1865	012072	012737	100000	000000	MOV	#CCLR,RKCS1		;CLEAR RK611 WITH CONTROLLER CLEAR

H05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 37
 DZR6A8.P11 T14 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)

SEQ 0038

1866	012100	012737	012106	001110		MOV	#15,\$LPERR	;LOAD LOOP ON ERROR LOCATION FOR ; SUBTEST LOOP
1867								
1868								
1869	012106				15:			
1870	012106	005062	000002			CLR	RKWC(R2)	;CLEAR WORD COUNT REG.
1871	012112	013762	002010	000000		MOV	CONFIG,RKCS1(R2)	;WRITE RKCS1
1872	012120	016237	000000	001126		MOV	RKCS1(R2),\$BDDAT	;STORE RKCS1
1873	012126	013737	002010	001124		MOV	CONFIG,\$GDDAT	;PREPARE EXPECTED RESULTS
1874	012134	042737	064000	001124		BIC	#DI!SPAR!CTO,\$GDDAT	;INITIALIZE READ ONLY BITS
1875	012142	052737	000200	001124		BIS	#RDY,\$GDDAT	
1876	012150	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;CHECK IF RKCS1 CORRECT
1877	012156	001404				BEQ	2\$;YES,TEST IF ANY OTHER REG MODIFIED
1878	012160	012737	066604	001322		MOV	#EM1000,EM3N+2	;LOAD ERROR MESSAGE
1879	012166	104003				ERROR	3	
1880	012170				25:			
1881	012170	016237	000004	001126		MOV	RKBA(R2),\$BDDAT	;STORE BUS AND REG
1882	012176	001406				BEQ	4\$;CHECK IF ZERO
1883	012200	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1884	012204	012737	067375	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
1885	012212	104003				ERROR	3	
1886	012214				45:			
1887	012214	016237	000002	001126		MOV	RKWC(R2),\$BDDAT	;STORE WORD COUNT REG
1888	012222	001406				BEQ	5\$;CHECK IF ZERO
1889	012224	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1890	012230	012737	067345	001322		MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
1891	012236	104003				ERROR	3	
1892	012240				55:			
1893	012240	016237	000006	001126		MOV	RKDA(R2),\$BDDAT	;STORE DISK AVERAGE REG
1894	012246	001406				BEQ	6\$;CHECK IF ZERO
1895	012250	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1896	012254	012737	067422	001322		MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
1897	012262	104003				ERROR	3	
1898	012264				65:			
1899	012264	016237	000016	001126		MOV	RKASOF(R2),\$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
1900	012272	001406				BEQ	7\$;CHECK IF ZERO
1901	012274	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1902	012300	012737	067566	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
1903	012306	104003				ERROR	3	
1904	012310				75:			
1905	012310	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT	;STORE COMMAND AND STATUS REG.2
1906	012316	022737	000100	001126		CMP	#IR,\$BDDAT	;CHECK IF CS2 CORRECT
1907	012324	001407				BEQ	8\$;YES,CONTINUE
1908	012326	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
1909	012334	012737	067460	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
1910	012342	104003				ERROR	3	
1911	012344				85:			
1912	012344	016237	000012	001126		MOV	RKDS(R2),\$BDDAT	;STORE DRIVE STATUS REG
1913	012352	001406				BEQ	9\$;CHECK IF ZERO
1914	012354	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1915	012360	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
1916	012366	104003				ERROR	3	
1917	012370	016237	000014	001126	95:			
1918	012376	001406				MOV	RKER(R2),\$BDDAT	;STORE ERROR REG
1919	012400	005037	001124			BEQ	10\$;CHECK IF ZERO
1920	012404	012737	067537	001322		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
1921	012412	104003				MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
						ERROR	3	

```

1922 012414          105:
1923 012414 016237 000020 001126   MOV   RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
1924 012422 001406          BEQ   125                ;CHECK IF ZERO
1925 012424 005037 001124          CLR   $GDDAT            ;LOAD EXPECTED CONTENTS
1926 012430 012737 067634 001322   MOV   #EM1025, EM3N+2 ;LOAD ERROR MESSAGE
1927 012436 104003          ERROR 3
1928 012440          125:
1929 012440 016237 000026 001126   MOV   RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG.1
1930 012446 012737 002000 001124   MOV   #MEWD, $GDDAT    ;LOAD EXPECTED MRI
1931 012454 032737 020000 001126   BIT   #ECCW, $BDDAT
1932 012462 001403          BEQ   135
1933 012464 052737 020000 001124   BIS   #ECCW, $GDDAT
1934 012472 023737 001124 001126 135:   CMP   $GDDAT, $BDDAT  ;CHECK IF MRI CORRECT
1935 012500 001404          BEQ   145                ;YES, ISSUE CONTROLLER CLEAR
1936 012502 012737 067661 001322   MOV   #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
1937 012510 104003          ERROR 3
1938 012512          145:
1939 012512 016237 000032 001126   MOV   RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
1940 012520 001406          BEQ   155                ;CHECK IF ZERO
1941 012522 005037 001124          CLR   $GDDAT            ;LOAD EXPECTED CONTENTS
1942 012526 012737 067737 001322   MOV   #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
1943 012534 104003          ERROR 3
1944 012536 016237 000030 001126 155:   MOV   RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
1945 012544 032737 010000 002010   BIT   #CFMT, CONFIG   ;CHECK IF IN 18 BIT FORMAT
1946 012552 001404          BEQ   165                ;NO, USE 4066
1947 012554 012737 005066 001124   MOV   #5066, $GDDAT   ;USE 5066
1948 012562 000403          BR   175                ;CHECK IF POSITION CORRECT
1949
1950 012564 012737 004066 001124 165:   MOV   #4066, $GDDAT   ;USE 4066
1951 012572 023737 001124 001126 175:   CMP   $GDDAT, $BDDAT  ;CHECK IF ECC POSITION CORRECT
1952 012600 001404          BEQ   185                ;YES, INITIALIZE RK611
1953 012602 012737 067712 001322   MOV   #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
1954 012610 104003          ERROR 3
1955 012612 016237 000000 002014 185:   MOV   RKCS1(R2), PREREG ;GET PREVIOUS CONTENTS
1956 012620 012762 100000 000000   MOV   #CLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
1957 012626 016237 000000 001126   MOV   RKCS1(R2), $BDDAT ;GET CURRENT VALUE
1958 012634 012737 000200 001124   MOV   #RDY, $GDDAT    ;LOAD EXPECTED CONTENTS
1959 012642 023737 001124 001126   CMP   $GDDAT, $BDDAT  ;CHECK IF RKCS1 CORRECT
1960 012650 001407          BEQ   195                ;YES, CHECK IF FINISHED
1961 012652 012737 063413 001330   MOV   #EM3, EM4N
1962 012660 012737 066604 001332   MOV   #EM1000, EM4N+2 ;LOAD ERROR MESSAGE
1963 012666 104004          ERROR 4
1964 012670 104415          195:   SCOP1 ;CHECK IF LOOP ON ERROR
1965 012672 000241          CLC   ;SHIFT IN ZERO
1966 012674 006137 002012          ROL   CONFIG1
1967 012700 013737 002012 002010   MOV   CONFIG1, CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
1968 012706 042737 100001 002010   BIC   #CLR!GO, CONFIG ;GO DO NOT SET
1969 012714 005301          DEC   R1                ;CHECK IF FINISHED
1970 012716 001402          BEQ   TST15            ;;YES, GO ON TO NEXT TEST
1971 012720 000137 012106          JMP   15

```

```

*****
*TEST 15 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)
*
* THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
* BY WRITING THE WORD COUNT TO ZERO.

```

1972
1973
1974
1975
1976
1977

J05

1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033

012724	000004		
012726	012737	000144	001200
012734	012701	000021	
012740	012737	077776	002010
012746	012737	177776	002012
012754	012737	064331	001320
012762	012737	100000	000000
012770	012737	012776	001110
012776			
012776	005062	000002	
013002	013762	002010	000000
013010	016237	000000	001126
013016	013737	002010	001124
013024	042737	064000	001124
013032	052737	000200	001124
013040	023737	001124	001126
013046	001404		
013050	012737	066604	001322
013056	104003		
013060			
013060	016237	000004	001126
013066	001406		
013070	005037	001124	
013074	012737	067375	001322
013102	104003		
013104			
013104	016237	000002	001126
013112	001406		
013114	005037	001124	
013120	012737	067345	001322
013126	104003		
013130			
013130	016237	000006	001126
013136	001406		
013140	005037	001124	
013144	012737	067422	001322
013152	104003		
013154			
013154	016237	000016	001126
013162	001406		
013164	005037	001124	

```

;*
;* 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
;*
;*****
;ST15: SCOPE
;MOV #100,$TIMES ;DO 100. ITERATIONS
;MOV #17,R1 ;LOAD NUMBER OF PATTERNS
;MOV #077776,CONFIG ;LOAD INITIAL CONFIGURATION
;MOV #177776,CONFIG1
;MOV #EM16,EM3N ;LOAD ERROR MESSAGE
;MOV #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
;MOV #15,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; ; SUBTEST LOOP

15: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV CONFIG,RKCS1(R2) ;WRITE RKCS1
MOV RKCS1(R2),$BDDAT ;STORE RKCS1
MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
BIC #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
BIS #RDY,$GDDAT
CMP $GDDAT,$BDDAT ;CHECK IF RKCS1 CORRECT
BEQ 2$ ;YES,TEST IF ANY OTHER REG MODIFIED
MOV #EM1000,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

25: MOV RKBA(R2),$BDDAT ;STORE BUS AND REG
BEQ 4$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

45: MOV RKWC(R2),$BDDAT ;STORE WORD COUNT REG
BEQ 5$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

55: MOV RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
BEQ 6$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

65: MO' RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
BEQ 7$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
    
```

K05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 40
 DZR6AB.P11 T15 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)

SEQ 0041

2034	013170	012737	067566	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
2035	013176	104003				ERROR	3	
2036	013200				7\$:			
2037	013200	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2
2038	013206	022737	000100	001126		CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
2039	013214	001407				BEQ	8\$;YES, CONTINUE
2040	013216	012737	000100	001124		MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
2041	013224	012737	067460	001322		MCV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
2042	013232	104003				ERROR	3	
2043	013234				8\$:			
2044	013234	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
2045	013242	001406				BEQ	9\$;CHECK IF ZERO
2046	013244	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2047	013250	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
2048	013256	104003				ERROR	3	
2049	013260	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
2050	013266	001406				BEQ	10\$;CHECK IF ZERO
2051	013270	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2052	013274	012737	067537	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
2053	013302	104003				ERROR	3	
2054	013304				10\$:			
2055	013304	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
2056	013312	001406				BEQ	12\$;CHECK IF ZERO
2057	013314	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2058	013320	012737	067634	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
2059	013326	104003				ERROR	3	
2060	013330				12\$:			
2061	013330	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
2062	013336	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MR1
2063	013344	032737	020000	001126		BIT	#ECCW, \$BDDAT	
2064	013352	001403				BEQ	13\$	
2065	013354	052737	020000	001124		BIS	#ECCW, \$GDDAT	
2066	013362	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
2067	013370	001404				BEQ	14\$;YES, ISSUE CONTROLLER CLEAR
2068	013372	012737	067661	001322		MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
2069	013400	104003				ERROR	3	
2070	013402				14\$:			
2071	013402	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
2072	013410	001406				BEQ	15\$;CHECK IF ZERO
2073	013412	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2074	013416	012737	067737	001322		MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
2075	013424	104003				ERROR	3	
2076	013426	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
2077	013434	032737	010000	002010		BIT	#CFMT, CONFIG	;CHECK IF IN 18 BIT FORMAT
2078	013442	001404				BEQ	16\$;NO, USE 4066
2079	013444	012737	005066	001124		MOV	#5066, \$GDDAT	;USE 5066
2080	013452	000403				BR	17\$;CHECK IF POSITION CORRECT
2081								
2082	013454	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	;USE 4066
2083	013462	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT
2084	013470	001404				BEQ	18\$;YES, INITIALIZE RK611
2085	013472	012737	067712	001322		MOV	#EM1029,EM3N+2	;LOAD ERROR MESSAGE
2086	013500	104003				ERROR	3	
2087	013502	016237	000000	002014	18\$:	MOV	RKCS1(R2), PREREG	;GET PREVIOUS CONTENTS
2088	013510	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	;CLEAR RK611 CONTROLLER
2089	013516	016237	000000	001126		MOV	RKCS1(R2), \$BDDAT	;GET CURRENT VALUE

L05

```

2090 013524 012737 000200 001124      MOV      #RDY,$GDDAT      ;LOAD EXPECTED CONTENTS
2091 013532 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;CHECK IF RKCS1 CORRECT
2092 013540 001407                BEQ      19$             ;YES, CHECK IF FINISHED
2093 013542 012737 063413 001330      MOV      #EM3,EM4N      ;LOAD ERROR MESSAGE
2094 013550 012737 066604 001332      MOV      #EM1000,EM4N+2
2095 013556 104004                ERROR    4
2096 013550 104415                19$:    SCOPE1          ;CHECK IF LOOP ON ERROR
2097 013562 000261                SEC                        ;SHIFT IN ONE
2098 013564 006137 002012      ROL      CONFIG1        ;
2099 013570 013737 002012 002010      MOV      CONFIG1,CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
2100 013576 042737 100001 002010      BIC      #CCLR!GO,CONFIG ; GO DO NOT SET
2101 013604 005301                DEC      R1              ;CHECK IF FINISHED
2102 013606 001402                BEQ      TST16           ;;YES, GO ON TO NEXT TEST
2103 013610 000137 012776      JMP      1$
2104
2105 ;*****
2106 ;*TEST 16 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)
2107 ;*
2108 ;* THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
2109 ;* BY WRITING THE WORD COUNT TO ZERO.
2110 ;*
2111 ;* THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
2112 ;* AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
2113 ;* SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
2114 ;* THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
2115 ;* REGISTER INTERACTION TAKES PLACE.
2116 ;*
2117 ;* 000002 000076 001776 037776
2118 ;* 000006 000176 003776 077776
2119 ;* 000016 000376 007776 000000
2120 ;* 000036 000776 017776
2121 ;*
2122 ;*****
2123 ;*TST16: SCOPE
2124 013614 000004                MOV      #100,$TIMES    ;;DO 100. ITERATIONS
2125 013616 012737 000144 001200      MOV      #17,R1         ;LOAD NUMBER OF PATTERNS
2126 013624 012701 000021                CLR      CONFIG        ;LOAD INITIAL CONFIGURATION
2127 013630 005037 002010      CLR      CONFIG1
2128 013634 005037 002012      MOV      #EM16,EM3N     ;LOAD ERROR MESSAGE
2129 013640 012737 064331 001320      MOV      #CCLR,RKCS1   ;CLEAR RK611 WITH CONTROLLER CLEAR
2130 013646 012737 100000 000000      MOV      #1$, $LPERR   ;LOAD LOOP ON ERROR LOCATION FOR
2131 013654 012737 013662 001110      MOV                        ; SUBTEST LOOP
2132
2133 013662                1$:    CLR      RKWC(R2)      ;CLEAR WORD COUNT REG.
2134 013662 005062 000002                MOV      CONFIG,RKCS1(R2) ;WRITE RKCS1
2135 013666 013762 002010 000000      MOV      RKCS1(R2),$BDDAT ;STORE RKCS1
2136 013674 016237 000000 001126      MOV      CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
2137 013702 013737 002010 001124      BIC      #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
2138 013710 042737 064000 001124      BIS      #RDY,$GDDAT
2139 013716 052737 000200 001124      CMP      $GDDAT,$BDDAT ;CHECK IF RKCS1 CORRECT
2140 013724 023737 001124 001126      BEQ      2$             ;YES, TEST IF ANY OTHER REG MODIFIED
2141 013732 001404                MOV      #EM1000,EM3N+2 ;LOAD ERROR MESSAGE
2142 013734 012737 066604 001322      ERROR    3
2143 013742 104003                2$:    MOV      RKBA(R2),$BDDAT ;STORE BUS AND REG
2144 013744
2145 013744 016237 000004 001126

```

M05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 42
 DZR6AB.P11 T16 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)

SEQ 0043

2146	013752	001406			BEQ	4\$;CHECK IF ZERO
2147	013754	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2148	013760	012737	067375	001322	MOV	#EM1019,EM3N+2		;LOAD ERROR MESSAGE
2149	013766	104003			ERROR	3		
2150	013770						4\$:	
2151	013770	016237	000002	001126	MOV	RKWC(R2), \$BDDAT		;STORE WORD COUNT REG
2152	013776	001406			BEQ	5\$;CHECK IF ZERO
2153	014000	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2154	014004	012737	067345	001322	MOV	#EM1018,EM3N+2		;LOAD ERROR MESSAGE
2155	014012	104003			ERROR	3		
2156	014014						5\$:	
2157	014014	016237	000006	001126	MOV	RKDA(R2), \$BDDAT		;STORE DISK AVERAGE REG
2158	014022	001406			BEQ	6\$;CHECK IF ZERO
2159	014024	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2160	014030	012737	067422	001322	MOV	#EM1020,EM3N+2		;LOAD ERROR MESSAGE
2161	014036	104003			ERROR	3		
2162	014040						6\$:	
2163	014040	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT		;STORE ATTENTION SUMMARY/OFFSET REG.
2164	014046	001406			BEQ	7\$;CHECK IF ZERO
2165	014050	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2166	014054	012737	067566	001322	MOV	#EM1024,EM3N+2		;LOAD ERROR MESSAGE
2167	014062	104003			ERROR	3		
2168	014064						7\$:	
2169	014064	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT		;STORE COMMAND AND STATUS REG.2
2170	014072	022737	000100	001126	CMP	#IR, \$BDDAT		;CHECK IF CS2 CORRECT
2171	014100	001407			BEQ	8\$;YES, CONTINUE
2172	014102	012737	000100	001124	MOV	#IR, \$GDDAT		;LOAD EXPECTED CONTENTS
2173	014110	012737	067460	001322	MOV	#EM1021,EM3N+2		;LOAD ERROR MESSAGE
2174	014116	104003			ERROR	3		
2175	014120						8\$:	
2176	014120	016237	000012	001126	MOV	RKDS(R2), \$BDDAT		;STORE DRIVE STATUS REG
2177	014126	001406			BEQ	9\$;CHECK IF ZERO
2178	014130	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2179	014134	012737	067501	001322	MOV	#EM1022,EM3N+2		;LOAD ERROR MESSAGE
2180	014142	104003			ERROR	3		
2181	014144	016237	000014	001126	MOV	RKER(R2), \$BDDAT		;STORE ERROR REG
2182	014152	001406			BEQ	10\$;CHECK IF ZERO
2183	014154	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2184	014160	012737	067537	001322	MOV	#EM1023,EM3N+2		;LOAD ERROR MESSAGE
2185	014166	104003			ERROR	3		
2186	014170						10\$:	
2187	014170	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT		;STORE CYLINDER ADD REG
2188	014176	001406			BEQ	12\$;CHECK IF ZERO
2189	014200	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
2190	014204	012737	067634	001322	MOV	#EM1025,EM3N+2		;LOAD ERROR MESSAGE
2191	014212	104003			ERROR	3		
2192	014214						12\$:	
2193	014214	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT		;STORE MAINTENANCE REG.1
2194	014222	012737	002000	001124	MOV	#MEWD, \$GDDAT		;LOAD EXPECTED MR1
2195	014230	032737	020000	001126	BIT	#ECCW, \$BDDAT		
2196	014236	001403			BEQ	13\$		
2197	014240	052737	020000	001124	BIS	#ECCW, \$GDDAT		
2198	014246	023737	001124	001126	CMP	\$GDDAT, \$BDDAT		;CHECK IF MR1 CORRECT
2199	014254	001404			BEQ	14\$;YES, ISSUE CONTROLLER CLEAR
2200	014256	012737	067661	001322	MOV	#EM1026,EM3N+2		;LOAD ERROR MESSAGE
2201	014264	104003			ERROR	3		

```

2202 014266          14$:
2203 014266 016237 000032 001126   MOV   RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2204 014274 001406          BEQ   15$                ;CHECK IF ZERO
2205 014276 005037 001124          CLR   $GDDAT             ;LOAD EXPECTED CONTENTS
2206 014302 012737 067737 001322   MOV   #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
2207 014310 104003          ERROR 3
2208 014312 016237 000030 001126   15$: MOV   RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2209 014320 032737 010000 002010   BIT   #CFMT, CONFIG    ;CHECK IF IN 18 BIT FORMAT
2210 014326 001404          BEQ   16$                ;NO, USE 4066
2211 014330 012737 005066 001124   MOV   #5066, $GDDAT    ;USE 5066
2212 014336 000403          BR   17$                ;CHECK IF POSITION CORRECT
2213
2214 014340 012737 004066 001124   16$: MOV   #4066, $GDDAT ;USE 4066
2215 014346 023737 001124 001126   17$: CMP   $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
2216 014354 001404          BEQ   18$                ;YES, INITIALIZE RK611
2217 014356 012737 067712 001322   MOV   #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
2218 014364 104003          ERROR 3
2219 014366 016237 000000 002014   18$: MOV   RKCS1(R2), PREREG ;GET PREVIOUS CONTENTS
2220 014374 012762 100000 000000   MOV   #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2221 014402 016237 000000 001126   MOV   RKCS1(R2), $BDDAT ;GET CURRENT VALUE
2222 014410 012737 000200 001124   MOV   #RDY, $GDDAT     ;LOAD EXPECTED CONTENTS
2223 014416 023737 001124 001126   CMP   $GDDAT, $BDDAT  ;CHECK IF RKCS1 CORRECT
2224 014424 001407          BEQ   19$                ;YES, CHECK IF FINISHED
2225 014426 012737 063413 001330   MOV   #EM3, EM4N      ;LOAD ERROR MESSAGE
2226 014434 012737 066604 001332   MOV   #EM1000, EM4N+2
2227 014442 104004          ERROR 4
2228 014444 104415          19$: SCOPI                ;CHECK IF LOOP ON ERROR
2229 014446 000261          SEC                    ;SHIFT IN ONE
2230 014450 006137 002012          ROL   CONFIG1
2231 014454 013737 002012 002010   MOV   CONFIG1, CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
2232 014462 042737 100001 002010   BIC   #CCLR!GO, CONFIG ; GO DO NOT SET
2233 014470 005301          DEC   R1                ;CHECK IF FINISHED
2234 014472 001402          BEQ   TST17             ;;YES, GO ON TO NEXT TEST
2235 014474 000137 013662          JMP   1$

```

```

2236
2237
2238 *****
2239 *TEST 17 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)
2240 *
2241 * THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
2242 * BY WRITING THE WORD COUNT TO ZERO.
2243 *
2244 * THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
2245 * AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
2246 * SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
2247 * THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
2248 * REGISTER INTERACTION TAKES PLACE.
2249 *
2250 * 000000 074000 077600 077770
2251 * 040000 076000 077700 077774
2252 * 060000 077000 077740 077776
2253 * 070000 077400 077760
2254 *****
2255 014500 000004 TST17: SCOPE
2256 014502 012737 000144 001200 MOV   #100., $TIMES ;;DO 100. ITERATIONS
2257 014510 012701 000021 MOV   #17., R1 ;LOAD NUMBER OF PATTERNS

```

B06

RAC:1 DYSKLESS CONTROLLER DIAGNOSTIC: F1 MD-11-CJR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 44
 CJR6A8.P:1 T17 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)

SEC 0045

2258	014514	005037	002010		CLR	CONFIG	;LOAD INITIAL CONFIGURATION
2259	014520	005037	002012		CLR	CONFIG	
2260	014524	012737	064331	001320	MOV	#EM16,EM3N	;LOAD ERROR MESSAGE
2261	014532	012737	100000	000000	MOV	#CCLR,RKCS1	;CLEAR RK611 WITH CONTROLLER CLEAR
2262	014540	012737	014546	001110	MOV	#IS,SLPERR	;LOAD LOOP ON ERROR LOCATION FOR ; SUBTEST LOOP
2263							
2264							
2265	014546				15:		
2266	014546	005062	000002		CLR	RKWC(R2)	;CLEAR WORD COUNT REG.
2267	014552	013762	002010	000000	MOV	CONFIG,RKCS1(R2)	;WRITE RKCS1
2268	014560	016237	000000	001126	MOV	RKCS1(R2),SBDDAT	;STORE RKCS1
2269	014566	013737	002010	001124	MOV	CONFIG,\$GDDAT	;PREPARE EXPECTED RESULTS
2270	014574	042737	064000	001124	BIC	#DI:SPAR:CTO,\$GDDAT	;INITIALIZE READ ONLY BITS
2271	014602	052737	000200	001124	BIS	#RDY,\$GDDAT	
2272	014610	023737	001124	001126	CMP	\$GDDAT,SBDDAT	;CHECK IF RKCS1 CORRECT
2273	014616	001404			BEQ	25	;YES,TEST IF ANY OTHER REG MODIFIED
2274	014620	012737	066604	001322	MOV	#EM100,EM3N+2	;LOAD ERROR MESSAGE
2275	014626	104003			ERROR	3	
2276	014630				25:		
2277	014630	016237	000004	001126	MOV	RKBA(R2),SBDDAT	;STORE BUS AND REG
2278	014636	001406			BEQ	45	;CHECK IF ZERO
2279	014640	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2280	014644	012737	067375	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
2281	014652	104003			ERROR	3	
2282	014654				45:		
2283	014654	016237	000002	001126	MOV	RKWC(R2),SBDDAT	;STORE WORD COUNT REG
2284	014662	001406			BEQ	55	;CHECK IF ZERO
2285	014664	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2286	014670	012737	067345	001322	MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
2287	014676	104003			ERROR	3	
2288	014700				55:		
2289	014700	016237	000006	001126	MOV	RKDA(R2),SBDDAT	;STORE DISK AVERAGE REG
2290	014706	001406			BEQ	65	;CHECK IF ZERO
2291	014710	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2292	014714	012737	067422	001322	MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
2293	014722	104003			ERROR	3	
2294	014724				65:		
2295	014724	016237	000016	001126	MOV	RKASOF(R2),SBDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
2296	014732	001406			BEQ	75	;CHECK IF ZERO
2297	014734	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2298	014740	012737	067566	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
2299	014746	104003			ERROR	3	
2300	014750				75:		
2301	014750	016237	000010	001126	MOV	RKCS2(R2),SBDDAT	;STORE COMMAND AND STATUS REG.2
2302	014756	022737	000100	001126	CMP	#IR,SBDDAT	;CHECK IF CS2 CORRECT
2303	014764	001407			BEQ	85	;YES,CONTINUE
2304	014766	012737	000100	001124	MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
2305	014774	012737	067460	001322	MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
2306	015002	104003			ERROR	3	
2307	015004				85:		
2308	015004	016237	000012	001126	MOV	RKDS(R2),SBDDAT	;STORE DRIVE STATUS REG
2309	015012	001406			BEQ	95	;CHECK IF ZERO
2310	015014	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2311	015020	012737	067501	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
2312	015026	104003			ERROR	3	
2313	015030	016237	000014	001126	95:	MOV	RKER(R2),SBDDAT ;STORE ERROR REG

C06

```

2314 015036 001406          BEQ      10$          ;CHECK IF ZERO
2315 015040 005037 001124      CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
2316 015044 012737 067537 001322  MOV      #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
2317 015052 104003          ERROR     3
2318 015054          10$:
2319 015054 016237 000020 001126  MOV      RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
2320 015062 001406          BEQ      12$          ;CHECK IF ZERO
2321 015064 005037 001124      CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
2322 015070 012737 067634 001322  MOV      #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
2323 015076 104003          ERROR     3
2324 015100          12$:
2325 015100 016237 000026 001126  MOV      RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG.1
2326 015106 012737 002000 001124  MOV      #MEWD, $GDDAT ;LOAD EXPECTED MRI
2327 015114 032737 020000 001126  BIT      #ECCW, $BDDAT
2328 015122 001403          BEQ      13$          ;CHECK IF MRI CORRECT
2329 015124 052737 020000 001124  BIS      #ECCW, $GDDAT
2330 015132 023737 001124 001126  13$:  CMP      $GDDAT, $BDDAT
2331 015140 001404          BEQ      14$          ;YES, ISSUE CONTROLLER CLEAR
2332 015142 012737 067661 001322  MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2333 015150 104003          ERROR     3
2334 015152          14$:
2335 015152 016237 000032 001126  MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2336 015160 001406          BEQ      15$          ;CHECK IF ZERO
2337 015162 005037 001124      CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
2338 015166 012737 067737 001322  MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
2339 015174 104003          ERROR     3
2340 015176 016237 000030 001126  15$:  MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2341 015204 032737 010000 002010  BIT      #CFMT, CONFIG ;CHECK IF IN 18 BIT FORMAT
2342 015212 001404          BEQ      16$          ;NO, USE 4066
2343 015214 012737 005066 001124  MOV      #5066, $GDDAT ;USE 5066
2344 015222 000403          BR       17$          ;CHECK IF POSITION CORRECT
2345
2346 015224 012737 004066 001124  16$:  MOV      #4066, $GDDAT ;USE 4066
2347 015232 023737 001124 001126  17$:  CMP      $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
2348 015240 001404          BEQ      18$          ;YES, INITIALIZE RK611
2349 015242 012737 067712 001322  MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
2350 015250 104003          ERROR     3
2351 015252 016237 000000 002014  18$:  MOV      RKCS1(R2), PREREG ;GET PREVIOUS CONTENTS
2352 015260 012762 100000 000000  MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2353 015266 016237 000000 001126  MOV      RKCS1(R2), $BDDAT ;GET CURRENT VALUE
2354 015274 012737 000200 001124  MOV      #RDY, $GDDAT ;LOAD EXPECTED CONTENTS
2355 015302 023737 001124 001126  CMP      $GDDAT, $BDDAT ;CHECK IF RKCS1 CORRECT
2356 015310 001407          BEQ      19$          ;YES, CHECK IF FINISHED
2357 015312 012737 063413 001330  MOV      #EM3, EM4N ;LOAD ERROR MESSAGE
2358 015320 012737 066604 001332  MOV      #EM1000, EM4N+2
2359 015326 104004          ERROR     4
2360 015330 104415          19$:  SCOP1          ;CHECK IF LOOP ON ERROR
2361 015332 000261          SEC          ;SHIFT IN ONE
2362 015334 006037 002012          ROR      CONFIG1
2363 015340 013737 002012 002010  MOV      CONFIG1, CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
2364 015346 042737 100001 002010  BIC      #CCLR!GO, CONFIG ; GO DO NOT SET
2365 015354 005301          DEC      R1 ;CHECK IF FINISHED
2366 015356 001402          BEQ      TST20 ;;YES, GO ON TO NEXT TEST
2367 015360 000137 014546          JMP      1$
2368
2369
;*****

```



```

2370
2371
2372
2373
2374
2375
2376
2377
2378
2379 015364 000004
2380 015366 012737 000764 001200
2381 015374 013702 001270
2382 015400 012737 177777 002010
2383 015406 012737 063767 001320
2384 015414 012762 100000 000000
2385 015422 005062 000002
2386 015426 012762 177777 000022
2387 015434 016237 000000 001126
2388 015442 022737 000200 001126
2389 015450 001407
2390 015452 012737 000200 001124
2391 015460 012737 067324 001322
2392 015466 104003
2393 015470 016237 000004 001126 15:
2394 015476 001406
2395 015500 005037 001124
2396 015504 012737 067375 001322
2397 015512 104003
2398 015514 016237 000002 001126 25:
2399 015522 001406
2400 015524 005037 001124
2401 015530 012737 067345 001322
2402 015536 104003
2403 015540 016237 000006 001126 35:
2404 015546 001406
2405 015550 005037 001124
2406 015554 012737 067422 001322
2407 015562 104003
2408 015564 016237 000016 001126 45:
2409 015572 001406
2410 015574 005037 001124
2411 015600 012737 067566 001322
2412 015606 104003
2413 015610 016237 000010 001126 55:
2414 015616 022737 000100 001126
2415 015624 001407
2416 015626 012737 000100 001124
2417 015634 012737 067460 001322
2418 015642 104003
2419 015644 016237 000012 001126 65:
2420 015652 001406
2421 015654 005037 001124
2422 015660 012737 067501 001322
2423 015666 104003
2424 015670 016237 000014 001126 75:
2425 015676 001406

```

```

: *TEST 20 REGISTER INTERACTION USING SPARE REG
: *
: * ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
: * WRITE THE WORD COUNT REGISTER WITH C.
: *
: * WRITE THE SPARE REGISTER WITH 177777 AND MAKE SURE
: * NO INTERACTION TAKES PLACE.
: *
: *****
↑ST20: SCOPE
MOV #500, STIMES ; DO 500. ITERATIONS
MOV $BASE, R2 ; LOAD RK611 BASE ADDRESS
MOV #177777, CONFIG ; LOAD CONFIGURATION WORD
MOV #EM9, EM3N ; LOAD ERROR MESSAGE
MOV #CLR, RKCS1(R2) ; CLEAR RK611 WITH CONTROLLER CLEAR
CLR FKWC(R2) ; CLEAR WORD COUNT REG.
MOV #177777, RKSPAR(R2) ; WRITE RKSPAR WITH 177777
MOV RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
CMP #RDY, $BDDAT ; CHECK IF CS1 CORRECT
BEQ 1$ ; YES, CONTINUE
MOV #RDY, $GDDAT ; LOAD EXPECTED RESULTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKBA(R2), $BDDAT ; STORE BUS ADD REG.
BEQ 2$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKWC(R2), $BDDAT ; STORE WORK COUNT REG.
BEQ 3$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKDA(R2), $BDDAT ; STORE DISK ADD REG
BEQ 4$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY/OFFSET REG
BEQ 5$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1024, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 2
CMP #IR, $BDDAT ; CHECK IF CS2 CORRECT
BEQ 6$ ; YES, CONTINUE
MOV #IR, $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1021, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKDS(R2), $BDDAT ; STORE DRIVE STATUS REG.
BEQ 7$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1022, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKER(R2), $BDDAT ; STORE ERROR REG
BEQ 8$ ; CHECK IF ZERO

```

E06

```

2426 015700 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2427 015704 012737 067537 001322 MOV #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
2428 015712 104003 ERROR 3
2429 015714 016237 000020 001126 8S: MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
2430 015722 001406 BEQ 10$ ;CHECK IF ZERO
2431 015724 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
2432 015730 012737 067634 001322 MOV #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
2433 015736 104003 ERROR 3
2434 015740 016237 000026 001126 10S: MOV RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG 1
2435 015746 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED CONTENTS
2436 015754 032737 020000 001126 BIT #ECCW, $BDDAT
2437 015762 001403 BEQ 11$
2438 015764 052737 020000 001124 BIS #ECCW, $GDDAT
2439 015772 023737 001124 001126 11S: CMP $GDDAT, $BDDAT ;CHECK IF MRI CORRECT
2440 016000 001404 BEQ 12$ ;YES, CONTINUE TEST
2441 016002 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2442 016010 104003 ERROR 3
2443 016012 016237 000032 001126 12S: MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2444 016020 001406 BEQ 13$ ;CHECK IF ZERO
2445 016022 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2446 016026 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
2447 016034 104003 ERROR 3
2448 016036 016237 000030 001126 13S: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG:
2449 016044 022737 004066 001126 CMP #4066, $BDDAT ;CHECK IF ECC POSITION CORRECT
2450 016052 001407 BEQ TST21 ;YES, GO TO NEXT TEST
2451 016054 012737 004066 001124 MOV #4066, $GDDAT ;LOAD EXPECTED CONTENTS
2452 016062 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
2453 016070 104003 ERROR 3

```

```

*****
:TEST 21 REGISTER INTERACTION USING WORD COUNT (PART 1)
*

```

```

* ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
* 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

```

```

*****
TST21: SCOPE

```

```

MOV #500, $TIMES ;DO 500. ITERATIONS
MOV #17, A1 ;LOAD NUMBER OF PATTERNS
MOV #000001, CONFIG ;LOAD INITIAL CONFIGURATION
MOV #EM7, EM3N ;LOAD ERROR MESSAGE
MOV #CCLR, RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #15, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

```

```

2478 016136 013762 002010 C00002 1S: MOV CONFIG, RKWC(R2) ;WRITE RKWC
2479 016136 016237 000002 001126 MOV RKWC(R2), $BDDAT ;STORE RKWC
2480 016144 016237 000002 001126 MOV CONFIG, $GDDAT ;PREPARE EXPECTED RESULTS
2481 016152 013737 002010 001124

```

2482	016160	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	:CHECK IF RKWC CORRECT
2483	016166	001404			BEQ	25	:YES TEST IF ANY OTHER REG MODIFIED
2484	016170	012737	066622	001322	MOV	#EM1001,EM3N+2	:LOAD ERROR MESSAGE
2485	016176	104003			ERROR	3	
2486	016200				25:		
2487	016200	016237	000000	001126	MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG. 1
2488	016206	022737	000200	001126	CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
2489	016214	001407			BEQ	35	:YES CONTINUE
2490	016216	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED RESULTS
2491	016224	012737	067324	001322	MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
2492	016232	104003			ERROR	3	
2493	016234				35:		
2494	016234	016237	000004	001126	MOV	RKBA(R2),\$BDDAT	:STORE BUS AND REG
2495	016242	001406			BEQ	45	:CHECK IF ZERO
2496	016244	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2497	016250	012737	067375	001322	MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
2498	016256	104003			ERROR	3	
2499	016260				45:		
2500	016260	016237	000006	001126	MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
2501	016266	001406			BEQ	65	:CHECK IF ZERO
2502	016270	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2503	016274	012737	067422	001322	MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
2504	016302	104003			ERROR	3	
2505	016304				65:		
2506	016304	016237	000016	001126	MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
2507	016312	001406			BEQ	75	:CHECK IF ZERO
2508	016314	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2509	016320	012737	067566	001322	MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
2510	016326	104003			ERROR	3	
2511	016330				75:		
2512	016330	016237	000010	001126	MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
2513	016336	022737	000100	001126	CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
2514	016344	001407			BEQ	85	:YES CONTINUE
2515	016346	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
2516	016354	012737	067460	001322	MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
2517	016362	104003			ERROR	3	
2518	016364				85:		
2519	016364	016237	000012	001126	MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
2520	016372	001406			BEQ	95	:CHECK IF ZERO
2521	016374	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2522	016400	012737	067501	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
2523	016406	104003			ERROR	3	
2524	016410	016237	000014	001126	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
2525	016416	001406			BEQ	105	:CHECK IF ZERO
2526	016420	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2527	016424	012737	067537	001322	MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
2528	016432	104003			ERROR	3	
2529	016434				105:		
2530	016434	016237	000020	001126	MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
2531	016442	001406			BEQ	125	:CHECK IF ZERO
2532	016444	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2533	016450	012737	067634	001322	MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
2534	016456	104003			ERROR	3	
2535	016460				125:		
2536	016460	016237	000026	001126	MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG.1
2537	016466	012737	002000	001124	MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1

```

2538 016474 032737 020000 001126 BIT #ECCW,$BDDAT
2539 016502 001403 BEQ 13$
2540 016504 052737 020000 001124 BIS #ECCW,$GDDAT
2541 016512 023737 001124 001126 13$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
2542 016520 001404 BEQ 14$ ;YES,ISSUE CONTROLLER CLEAR
2543 016522 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2544 016530 104003 ERROR 3
2545 016532 14$:
2546 016532 016237 000032 001126 MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2547 016540 001406 BEQ 15$ ;CHECK IF ZERO
2548 016542 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2549 016546 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
2550 016554 104003 ERROR 3
2551 016556 016237 000030 001126 15$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2552 016564 012737 004066 001124 16$: MOV #4066,$GDDAT ;USE 4066
2553 016572 023737 001124 001126 17$: CMP $GDDAT,$BDDAT ;CHECK IF ECC POSITION CORRECT
2554 016600 001404 BEQ 18$ ;YES,INITIALIZE RK611
2555 016602 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
2556 016610 104003 ERROR 3
2557 016612 016237 000002 002014 18$: MOV RKWC(R2),PREREG ;GET PREVIOUS CONTENTS
2558 016620 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
2559 016626 016237 000002 001126 MOV RKWC(R2),$BDDAT ;GET CURRENT VALUE
2560 016634 023737 002010 001126 CMP CONFIG,$BDDAT ;CHECK IF WORD COUNT NOT CHANGE
2561 ; BY CONTROLLER CLEAR
2562 016642 001412 BEQ 19$ ;YES,CHECK IF FINISHED
2563 016644 013737 002010 001124 MOV CONFIG,$GDDAT ;LOAD EXPECTED DATA
2564 016652 012737 063413 001310 MOV #EM3,EM2N ;LOAD ERROR MESSAGE
2565 016660 012737 067345 001312 MOV #EM1018,EM2N+2
2566 016666 104002 ERROR 2
2567 016670 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
2568 016672 000241 CLC ;SHIFT IN ZERO
2569 016674 006137 002010 ROL CONFIG
2570 016700 005301 DEC R1 ;CHECK IF FINISHED
2571 016702 001402 BEQ TST22 ;;YES,GO ON TO NEXT TEST
2572 016704 000137 016136 JMP 1$

```

```

*****
*TEST 22 REGISTER INTERACTION USING WORD COUNT (PART 2)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
* 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
*
*****

```

```

2588 016710 000004 TST22: SCOPE
2589 016712 012737 000764 001200 MOV #500,$TIMES ;DO 500. ITERATIONS
2590 016720 012701 000021 MOV #17,R1 ;LOAD NUMBER OF PATTERNS
2591 016724 012737 177776 002010 MOV #177776,CONFIG ;LOAD INITIAL CONFIGURATION
2592 016732 012737 063673 001320 MOV #EM7,EM3N ;LOAD ERROR MESSAGE
2593 016740 012737 100000 000000 MOV #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR

```

H06

2594	016746	012737	016754	001110		MOV	#15,SLPERR	;LOAD LOOP ON ERROR LOCATION FOR ; SUBTEST LOOP
2595								
2596								
2597	016754				15:			
2598	016754	013762	002010	000002		MOV	CONFIG,RKWC(R2)	;WRITE RKWC
2599	016762	016237	000002	001126		MOV	RKWC(R2),SBDDAT	;STORE RKWC
2600	016770	013737	002010	001124		MOV	CONFIG,\$GDDAT	;PREPARE EXPECTED RESULTS
2601	016776	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;CHECK IF RKWC CORRECT
2602	017004	001404				BEQ	2\$;YES,TEST IF ANY OTHER REG MODIFIED
2603	017006	012737	066622	001322		MOV	#EM1001,EM3N+2	;LOAD ERROR MESSAGE
2604	017014	104003				ERROR	3	
2605	017016				25:			
2606	017016	016237	000000	001126		MOV	RKCS1(R2),SBDDAT	;STORE COMMAND AND STATUS REG. 1
2607	017024	022737	000200	001126		CMP	#RDY,\$BDDAT	;CHECK IF CS1 CORRECT
2608	017032	001407				BEQ	3\$;YES,CONTINUE
2609	017034	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED RESULTS
2610	017042	012737	067324	001322		MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
2611	017050	104003				ERROR	3	
2612	017052				35:			
2613	017052	016237	000004	001126		MOV	RKBA(R2),SBDDAT	;STORE BUS AND REG
2614	017060	001406				BEQ	4\$;CHECK IF ZERO
2615	017062	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2616	017066	012737	067375	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
2617	017074	104003				ERROR	3	
2618	017076				45:			
2619	017076	016237	000006	001126		MOV	RKDA(R2),SBDDAT	;STORE DISK AVERAGE REG
2620	017104	001406				BEQ	6\$;CHECK IF ZERO
2621	017106	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2622	017112	012737	067422	001322		MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
2623	017120	104003				ERROR	3	
2624	017122				65:			
2625	017122	016237	000016	001126		MOV	RKASOF(R2),SBDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
2626	017130	001406				BEQ	7\$;CHECK IF ZERO
2627	017132	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2628	017136	012737	067566	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
2629	017144	104003				ERROR	3	
2630	017146				75:			
2631	017146	016237	000010	001126		MOV	RKCS2(R2),SBDDAT	;STORE COMMAND AND STATUS REG.2
2632	017154	022737	000100	001126		CMP	#IR,\$BDDAT	;CHECK IF CS2 CORRECT
2633	017162	001407				BEQ	8\$;YES,CONTINUE
2634	017164	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
2635	017172	012737	067460	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
2636	017200	104003				ERROR	3	
2637	017202				85:			
2638	017202	016237	000012	001126		MOV	RKDS(R2),SBDDAT	;STORE DRIVE STATUS REG
2639	017210	001406				BEQ	9\$;CHECK IF ZERO
2640	017212	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2641	017216	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
2642	017224	104003				ERROR	3	
2643	017226	016237	000014	001126	95:			
2644	017234	001406				MOV	RKER(R2),SBDDAT	;STORE ERROR REG
2645	017236	005037	001124			BEQ	10\$;CHECK IF ZERO
2646	017242	012737	067537	001322		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
2647	017250	104003				MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
2648	017252					ERROR	3	
2649	017252	016237	000020	001126	105:			
						MOV	RKDCYL(R2),SBDDAT	;STORE CYLINDER ADD REG

```

2650 017260 001406          BEQ      12$          ;CHECK IF ZERO
2651 017262 005037 001124    CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
2652 017266 012737 067634 001322  MOV      #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
2653 017274 104003          ERROR    3
2654 017276          12$:
2655 017276 016237 000026 001126  MOV      RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG.1
2656 017304 012737 002000 001124  MOV      #MEWD, $GDDAT    ;LOAD EXPECTED MRI
2657 017312 032737 020000 001126  BIT      #ECCW, $BDDAT
2658 017320 001403          BEQ      13$
2659 017322 052737 020000 001124  BIS      #ECCW, $GDDAT
2660 017330 023737 001124 001126 13$:  CMP      $GDDAT, $BDDAT    ;CHECK IF MRI CORRECT
2661 017336 001404          BEQ      14$          ;YES, ISSUE CONTROLLER CLEAR
2662 017340 012737 067661 001322  MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2663 017346 104003          ERROR    3
2664 017350          14$:
2665 017350 016237 000032 001126  MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2666 017356 001406          BEQ      15$          ;CHECK IF ZERO
2667 017360 005037 001124    CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
2668 017364 012737 067737 001322  MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
2669 017372 104003          ERROR    3
2670 017374 016237 000030 001126 15$:  MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2671 017402 012737 004066 001124 16$:  MOV      #4066, $GDDAT    ;USE 4066
2672 017410 023737 001124 001126 17$:  CMP      $GDDAT, $BDDAT    ;CHECK IF ECC POSITION CORRECT
2673 017416 001404          BEQ      18$          ;YES, INITIALIZE RK611
2674 017420 012737 067712 001322  MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
2675 017426 104003          ERROR    3
2676 017430 016237 000002 002014 18$:  MOV      RKWC(R2), PREREG ;GET PREVIOUS CONTENTS
2677 017436 012762 100000 000000  MOV      #CLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2678 017444 016237 000002 001126  MOV      RKWC(R2), $BDDAT ;GET CURRENT VALUE
2679 017452 023737 002010 001126  CMP      CONFIG, $BDDAT   ;CHECK IF WORD COUNT NOT CHANGE
2680          ;BY CONTROLLER CLEAR
2681 017460 001412          BEQ      19$          ;YES, CHECK IF FINISHED
2682 017462 013737 002010 001124  MOV      CONFIG, $GDDAT  ;LOAD EXPECTED DATA
2683 017470 012737 063413 001310  MOV      #EM3, EM2N      ;LOAD ERROR MESSAGE
2684 017476 012737 067345 001312  MOV      #EM1018,EM2N+2
2685 017504 104002          ERROR    2
2686 017506 104415          19$:  SCOP1    ;CHECK IF LOOP ON ERROR
2687 017510 000261          SEC      ;SHIFT IN ONE
2688 017512 006137 002010  ROL      CONFIG
2689 017516 005301          DEC      R1            ;CHECK IF FINISHED
2690 017520 001402          BEQ      TST23        ;;YES, GO ON TO NEXT TEST
2691 017522 000137 016754          JMP      1$

```

```

*****
*TEST 23 REGISTER INTERACTION USING WORD COUNT (PART 3)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
* 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
*

```

```

2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705

```

JOB

```

2706
2707 017526 000004
2708 017530 012737 000764 001200
2709 017536 012701 000021
2710 017542 005037 002010
2711 017546 012737 063673 001320
2712 017554 012737 100000 000000
2713 017562 012737 017570 001110
2714
2715
2716 017570
2717 017570 013762 002010 000002
2718 017576 016237 000002 001126
2719 017604 013737 002010 001124
2720 017612 023737 001124 001126
2721 017620 001404
2722 017622 012737 066622 001322
2723 017630 104003
2724 017632
2725 017632 016237 000000 001126
2726 017640 022737 000200 001126
2727 017646 001407
2728 017650 012737 000100 001124
2729 017656 012737 067324 001322
2730 017664 104003
2731 017666
2732 017666 016237 000004 001126
2733 017674 001406
2734 017676 005037 001124
2735 017702 012737 067375 001322
2736 017710 104003
2737 017712
2738 017712 016237 000006 001126
2739 017720 001406
2740 017722 005037 001124
2741 017726 012737 067422 001322
2742 017734 104003
2743 017736
2744 017736 016237 000016 001126
2745 017744 001406
2746 017746 005037 001124
2747 017752 012737 067566 001322
2748 017760 104003
2749 017762
2750 017762 016237 000010 001126
2751 017770 022737 000100 001126
2752 017776 001407
2753 020000 012737 000100 001124
2754 020006 012737 067460 001322
2755 020014 104003
2756 020016
2757 020016 016237 000012 001126
2758 020024 001406
2759 020026 005037 001124
2760 020032 012737 067501 001322
2761 020040 104003

*****
↑ST23: SCOPE
MOV #500,$TIMES ;DO 500. ITERATIONS
MOV #17,R1 ;LOAD NUMBER OF PATTERNS
CLR CONFIG ;LOAD INITIAL CONFIGURATION
MOV #EM7,EM3N ;LOAD ERROR MESSAGE
MOV #CLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #IS,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1$:
MOV CONFIG,RKWC(R2) ;WRITE RKWC
MOV RKWC(R2),$BDDAT ;STORE RKWC
MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
CMP $GDDAT,$BDDAT ;CHECK IF RKWC CORRECT
BEQ 2$ ;YES,TEST IF ANY OTHER REG MODIFIED
MOV #EM1001,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 #IR,$GDDAT ;LOAD EXPECTED RESULTS
MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

3$:
MOV RKBA(R2),$BDDAT ;STORE BUS AND REG
BEQ 4$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

4$:
MOV RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
BEQ 6$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

6$:
MOV RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
BEQ 7$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1024,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

7$:
MOV RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
CMP #IR,$BDDAT ;CHECK IF CS2 CORRECT
BEQ 8$ ;YES, CONTINUE
MOV #IR,$GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1021,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

8$:
MOV RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
BEQ 9$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```



```

2762 020042 016237 000014 001126 9$:  MOV  RKER(R2), $BDDAT ;STORE ERROR REG
2763 020050 001406 001124 001124  BEQ  10$              ;CHECK IF ZERO
2764 020052 005037 001124 001124  CLR  $GDDAT          ;LOAD EXPECTED CONTENTS
2765 020056 012737 067537 001322  MOV  #EM1023, EM3N+2 ;LOAD ERROR MESSAGE
2766 020064 104003 001126 001126  ERROR 3
2767 020066 001126 001126 001126 10$:  MOV  RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
2768 020066 016237 000020 001126  BEQ  12$              ;CHECK IF ZERO
2769 020074 001406 001124 001124  CLR  $GDDAT          ;LOAD EXPECTED CONTENTS
2770 020076 005037 001124 001322  MOV  #EM1025, EM3N+2 ;LOAD ERROR MESSAGE
2771 020102 012737 067634 001322  ERROR 3
2772 020110 104003 001126 001126 12$:  MOV  RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG. 1
2773 020112 001126 001126 001126  MOV  #MEWD, $GDDAT   ;LOAD EXPECTED MRI
2774 020112 016237 000026 001126  MOV  #ECCW, $BDDAT
2775 020120 012737 002000 001124  BIT  #ECCW, $BDDAT
2776 020126 032737 020000 001126  BEQ  13$
2777 020134 001403 001124 001124  BIS  #ECCW, $GDDAT
2778 020136 052737 020000 001124  CMP  $GDDAT, $BDDAT ;CHECK IF MRI CORRECT
2779 020144 023737 001124 001126 13$:  BEQ  14$              ;YES, ISSUE CONTROLLER CLEAR
2780 020152 001404 001124 001126  MOV  #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
2781 020154 012737 067661 001322  ERROR 3
2782 020162 104003 001126 001126 14$:  MOV  RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2783 020164 001126 001126 001126  BEQ  15$              ;CHECK IF ZERO
2784 020164 016237 000032 001126  CLR  $GDDAT          ;LOAD EXPECTED CONTENTS
2785 020172 001406 001124 001124  MOV  #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
2786 020174 005037 001124 001322  MOV  #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
2787 020200 012737 067737 001322  ERROR 3
2788 020206 104003 001126 001126 15$:  MOV  RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2789 020210 016237 000030 001126 16$:  MOV  #4066, $GDDAT   ;USE 4066
2790 020216 012737 004066 001124 17$:  CMP  $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
2791 020224 023737 001124 001126  BEQ  18$              ;YES, INITIALIZE RK611
2792 020232 001404 001126 001126  MOV  #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
2793 020234 012737 067712 001322  ERROR 3
2794 020242 104003 001126 002014 18$:  MOV  RKWC(R2), PREREG ;GET PREVIOUS CONTENTS
2795 020244 016237 000002 002014  MOV  #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2796 020252 012762 100000 000000  MOV  RKWC(R2), $BDDAT ;GET CURRENT VALUE
2797 020260 016237 000002 001126  CMP  CONFIG, $BDDAT ;CHECK IF WORD COUNT NOT CHANGE
2798 020266 023737 002010 001126  BY CONTROLLER CLEAR
2799 020274 001412 001126 001126  BEQ  19$              ;YES, CHECK IF FINISHED
2800 020274 001412 001126 001126  MOV  CONFIG, $GDDAT ;LOAD EXPECTED DATA
2801 020276 013737 002010 001124  MOV  #EM3, EM2N     ;LOAD ERROR MESSAGE
2802 020304 012737 063413 001310  MOV  #EM1018, EM2N+2
2803 020312 012737 067345 001312  ERROR 2
2804 020320 104002 001126 001126 19$:  SCOP1 ;CHECK IF LOOP ON ERROR
2805 020322 104415 001126 001126  SEC  ;SHIFT IN ONE
2806 020324 000261 001126 001126  ROL  CONFIG
2807 020326 006137 002010 001126  DEC  R1              ;CHECK IF FINISHED
2808 020332 005301 001126 001126  BEQ  TST24          ;;YES, GO ON TO NEXT TEST
2809 020334 001402 001126 001126  JMP  1$
2810 020336 000137 017570 001126 1$

```

```

2811 *****
2812 ;*TEST 24 REGISTER INTERACTION USING WORD COUNT (PART 4)
2813 ;*
2814 ;*
2815 ;* ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
2816 ;* WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
2817 ;* AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE

```

Address	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	
2818																					
2819																					
2820																					
2821																					
2822																					
2823																					
2824																					
2825																					
2826	020342	000004																			
2827	020344	012737	000764	001200																	
2828	020352	012701	000021																		
2829	020356	005037	002010																		
2830	020362	012737	063673	001320																	
2831	020370	012737	100000	000000																	
2832	020376	012737	020404	001110																	
2833																					
2834																					
2835	020404																				
2836	020404	013762	002010	000002																	
2837	020412	016237	000002	001126																	
2838	020420	013737	002010	001124																	
2839	020426	023737	001124	001126																	
2840	020434	001404																			
2841	020436	012737	066622	001322																	
2842	020444	104003																			
2843	020446																				
2844	020446	016237	000000	001126																	
2845	020454	022737	000200	001126																	
2846	020462	001407																			
2847	020464	012737	000100	001124																	
2848	020472	012737	067324	001322																	
2849	020500	104003																			
2850	020502																				
2851	020502	016237	000004	001126																	
2852	020510	001406																			
2853	020512	005037	001124																		
2854	020516	012737	067375	001322																	
2855	020524	104003																			
2856	020526																				
2857	020526	016237	000006	001126																	
2858	020534	001406																			
2859	020536	005037	001124																		
2860	020542	012737	067422	001322																	
2861	020550	104003																			
2862	020552																				
2863	020552	016237	000016	001126																	
2864	020560	001406																			
2865	020562	005037	001124																		
2866	020566	012737	067566	001322																	
2867	020574	104003																			
2868	020576																				
2869	020576	016237	000010	001126																	
2870	020604	022737	000100	001126																	
2871	020612	001407																			
2872	020614	012737	000100	001124																	
2873	020622	012737	067460	001322																	

```

; * PLACE.
; *
; * 100000 174000 177600 177770 000000
; * 140000 176000 177700 177774
; * 160000 177000 177740 177776
; * 170000 177400 177760 177777
; *
; *****
; ST24: SCOPE
; MOV #500, $TIMES ; DO 500. ITERATIONS
; MOV #17, R1 ; LOAD NUMBER OF PATTERNS
; CLR CONFIG ; LOAD INITIAL CONFIGURATION
; MOV #EM7, EM3N ; LOAD ERROR MESSAGE
; MOV #CCLR, RKCS1 ; CLEAR RK611 WITH CONTROLLER CLEAR
; MOV #1$, $LPERR ; LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP
;
1$: MOV CONFIG, RKWC(R2) ; WRITE RKWC
; MOV RKWC(R2), $BDDAT ; STORE RKWC
; MOV CONFIG, $GDDAT ; PREPARE EXPECTED RESULTS
; CMP $GDDAT, $BDDAT ; CHECK IF RKWC CORRECT
; BEQ 2$ ; YES, TEST IF ANY OTHER REG MODIFIED
; MOV #EM1001, 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 #IR, $GDDAT ; LOAD EXPECTED RESULTS
; MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
; ERROR 3
;
3$: MOV RKBA(R2), $BDDAT ; STORE BUS AND REG
; BEQ 4$ ; CHECK IF ZERO
; CLR $GDDAT ; LOAD EXPECTED CONTENTS
; MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
; ERROR 3
;
4$: MOV RKDA(R2), $BDDAT ; STORE DISK AVERAGE REG
; BEQ 5$ ; CHECK IF ZERO
; CLR $GDDAT ; LOAD EXPECTED CONTENTS
; MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
; ERROR 3
;
5$: MOV RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY/OFFSET REG.
; BEQ 7$ ; CHECK IF ZERO
; CLR $GDDAT ; LOAD EXPECTED CONTENTS
; MOV #EM1024, EM3N+2 ; LOAD ERROR MESSAGE
; ERROR 3
;
7$: MOV RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG.2
; CMP #IR, $BDDAT ; CHECK IF CS2 CORRECT
; BEQ 8$ ; YES, CONTINUE
; MOV #IR, $GDDAT ; LOAD EXPECTED CONTENTS
; MOV #EM1021, EM3N+2 ; LOAD ERROR MESSAGE

```

2874	020630	104003				ERROR	3	
2875	020632				8\$:			
2876	020632	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
2877	020640	001406				BEQ	9\$; CHECK IF ZERO
2878	020642	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
2879	020646	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
2880	020654	104003				ERROR	3	
2881	020656	016737	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
2882	020664	001406				BEQ	10\$; CHECK IF ZERO
2883	020666	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
2884	020672	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
2885	020700	104003				ERROR	3	
2886	020702				10\$:			
2887	020702	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
2888	020710	001406				BEQ	12\$; CHECK IF ZERO
2889	020712	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
2890	020716	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
2891	020724	104003				ERROR	3	
2892	020726				12\$:			
2893	020726	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG. 1
2894	020734	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MRI
2895	020742	032737	020000	001126		BIT	#ECCW, \$BDDAT	
2896	020750	001403				BEQ	13\$	
2897	020752	052737	020000	001124		BIS	#ECCW, \$GDDAT	
2898	020760	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MRI CORRECT
2899	020766	001404				BEQ	14\$; YES, ISSUE CONTROLLER CLEAR
2900	020770	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
2901	020776	104003				ERROR	3	
2902	021000				14\$:			
2903	021000	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
2904	021006	001406				BEQ	15\$; CHECK IF ZERO
2905	021010	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
2906	021014	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
2907	021022	104003				ERROR	3	
2908	021024	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG.
2909	021032	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	; USE 4066
2910	021040	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF ECC POSITION CORRECT
2911	021046	001404				BEQ	18\$; YES, INITIALIZE RK611
2912	021050	012737	067712	001322		MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE
2913	021056	104003				ERROR	3	
2914	021060	016237	000002	002014	18\$:	MOV	RKWC(R2), PREREG	; GET PREVIOUS CONTENTS
2915	021066	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	; CLEAR RK611 CONTROLLER
2916	021074	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	; GET CURRENT VALUE
2917	021102	023737	002010	001126		CMP	CONFIG, \$BDDAT	; CHECK IF WORD COUNT NOT CHANGE
2918								; BY CONTROLLER CLEAR
2919	021110	001412				BEQ	19\$; YES, CHECK IF FINISHED
2920	021112	013737	002010	001124		MOV	CONFIG, \$GDDAT	; LOAD EXPECTED DATA
2921	021120	012737	063413	001310		MOV	#EM3, EM2N	; LOAD ERROR MESSAGE
2922	021126	012737	067345	001312		MOV	#EM1018, EM2N+2	
2923	021134	104002				ERROR	2	
2924	021136	104415			19\$:	SCOPI		; CHECK IF LOOP ON ERROR
2925	021140	000261				SEC		; SHIFT IN ONE
2926	021142	006037	002010			ROR	CONFIG	
2927	021146	005301				DEC	R1	; CHECK IF FINISHED
2928	021150	001402				BEQ	TST25	; YES, GO ON TO NEXT TEST
2929	021152	000137	020404			JMP	1\$	

2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985

021156	000004		
021160	012737	000764	001200
021166	012701	000021	
021172	012737	000001	002010
021200	012737	063732	001320
021206	012737	100000	000000
021214	012737	021222	001110
021222			
021222	005062	000002	
021226	013762	002010	000006
021234	016237	000006	001126
021242	013737	002010	001124
021250	042737	174340	001124
021256	023737	001124	001126
021264	001404		
021266	012737	066671	001322
021274	104003		
021276			
021276	016237	000000	001126
021304	022737	000200	001126
021312	001407		
021314	012737	000100	001124
021322	012737	067324	001322
021330	104003		
021332			
021332	016237	000004	001126
021340	001406		
021342	005037	001124	
021346	012737	067375	001322
021354	104003		
021356			
021356	016237	000002	001126
021364	001406		
021366	005037	001124	
021372	012737	067345	001322
021400	104003		
021402			

: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	

```

1$: SCOPE
   MOV #500, $TIMES ; DO 500. ITERATIONS
   MOV #17, R1 ; LOAD NUMBER OF PATTERNS
   MOV #000001, CONFIG ; LOAD INITIAL CONFIGURATION
   MOV #EM8, EM3N ; LOAD ERROR MESSAGE
   MOV #CCLR, RKCS1 ; CLEAR RK611 WITH CONTROLLER CLEAR
   MOV #1$, $LPERR ; LOAD LOOP ON ERROR LOCATION FOR
   ; SUBTEST LOOP

```

```

1$: CLR RKWC(R2) ; CLEAR WORD COUNT REG.
   MOV CONFIG, RKDA(R2) ; WRITE RKDA
   MOV RKDA(R2), $BDDAT ; STORE RKDA
   MOV CONFIG, $GDDAT ; PREPARE EXPECTED RESULTS
   BIC #174340, $GDDAT ; INITIALIZE READ ONLY BITS
   CMP $GDDAT, $BDDAT ; CHECK IF RKDA CORRECT
   BEQ 2$ ; YES, TEST IF ANY OTHER REG MODIFIED
   MOV #EM1003, 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 #IR, $GDDAT ; LOAD EXPECTED RESULTS
   MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
   ERROR 3

```

```

3$: MOV RKBA(R2), $BDDAT ; STORE BUS AND REG
   BEQ 4$ ; CHECK IF ZERO
   CLR $GDDAT ; LOAD EXPECTED CONTENTS
   MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
   ERROR 3

```

```

4$: MOV RKWC(R2), $BDDAT ; STORE WORD COUNT REG
   BEQ 5$ ; CHECK IF ZERO
   CLR $GDDAT ; LOAD EXPECTED CONTENTS
   MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
   ERROR 3

```

5\$:

2986	021402	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
2987	021410	001406			BEQ	7\$:CHECK IF ZERO
2988	021412	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2989	021416	012737	067566	001322	MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
2990	021424	104003			ERROR	3	
2991	021426						7\$:
2992	021426	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
2993	021434	022737	000100	001126	CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
2994	021442	001407			BEQ	8\$:YES, CONTINUE
2995	021444	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
2996	021452	012737	067460	001322	MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
2997	021460	104003			ERROR	3	
2998	021462						8\$:
2999	021462	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
3000	021470	001406			BEQ	9\$:CHECK IF ZERO
3001	021472	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3002	021476	012737	067501	001322	MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
3003	021504	104003			ERROR	3	
3004	021506	016237	000014	001126	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
3005	021514	001406			BEQ	10\$:CHECK IF ZERO
3006	021516	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3007	021522	012737	067537	001322	MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
3008	021530	104003			ERROR	3	
3009	021532						10\$:
3010	021532	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
3011	021540	001406			BEQ	12\$:CHECK IF ZERO
3012	021542	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3013	021546	012737	067634	001322	MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
3014	021554	104003			ERROR	3	
3015	021556						12\$:
3016	021556	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
3017	021564	012737	002000	001124	MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MRI
3018	021572	032737	020000	001126	BIT	#ECCW, \$BDDAT	
3019	021600	001403			BEQ	13\$	
3020	021602	052737	020000	001124	BIS	#ECCW, \$GDDAT	
3021	021610	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF MRI CORRECT
3022	021616	001404			BEQ	14\$:YES, ISSUE CONTROLLER CLEAR
3023	021620	012737	067661	001322	MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
3024	021626	104003			ERROR	3	
3025	021630						14\$:
3026	021630	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
3027	021636	001406			BEQ	15\$:CHECK IF ZERO
3028	021640	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3029	021644	012737	067737	001322	MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
3030	021652	104003			ERROR	3	
3031	021654	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
3032	021662	012737	004066	001124	MOV	#4066, \$GDDAT	:USE 4066
3033	021670	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
3034	021676	001404			BEQ	18\$:YES, INITIALIZE RK611
3035	021700	012737	067712	001322	MOV	#EM1029, EM3N+2	:LOAD ERROR MESSAGE
3036	021706	104003			ERROR	3	
3037	021710	016237	000006	002014	MOV	RKDA(R2), PREREG	:GET PREVIOUS CONTENTS
3038	021716	012762	100000	000000	MOV	#CLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
3039	021724	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	:GET CURRENT VALUE
3040	021732	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3041	021736	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF RKDA CORRECT

3042	021744	001407			BEQ	19\$:YES, CHECK IF FINISHED
3043	021746	012737	063413	001330	MOV	#EM3,EM4N	:LOAD ERROR MESSAGE
3044	021754	012737	066671	001332	MOV	#EM1003,EM4N+2	
3045	021762	104004			ERROR	4	
3046	021764	104415			19\$: SCOP1		:CHECK IF LOOP ON ERROR
3047	021766	000241			CLC		:SHIFT IN ZERO
3048	021770	006137	002010		ROL	CONFIG	
3049	021774	035301			DEC	R1	:CHECK IF FINISHED
3050	021776	001402			BEQ	TST26	:;YES, GO ON TO NEXT TEST
3051	022000	000137	021222		JMP	1\$	

 TEST 26 REGISTER INTERACTION USING DISK ADDRESS (PART 2)

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.

17777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

 TST26: SCOPE *****

3069	022004	000004			MOV	#500,\$TIMES	:DO 500. ITERATIONS
3070	022006	012737	000764	001200	MOV	#17,R1	:LOAD NUMBER OF PATTERNS
3071	022014	012701	000021		MOV	#177776,CONFIG	:LOAD INITIAL CONFIGURATION
3072	022020	012737	177776	002010	MOV	#EM8,EM3N	:LOAD ERROR MESSAGE
3073	022026	012737	063732	001320	MOV	#CCLR,RKCS1	:CLEAR RK611 WITH CONTROLLER CLEAR
3074	022034	012737	100000	000000	MOV	#1\$,SLPERR	:LOAD LOOP ON ERROR LOCATION FOR
3075	022042	012737	022050	001110	MOV		: SUBTEST LOOP

1\$: CLR RKWC(R2) :CLEAR WORD COUNT REG.

3078	022050	005062	000002		MOV	CONFIG,RKDA(R2)	:WRITE RKDA
3079	022050	013762	002010	000006	MOV	RKDA(R2),SBDDAT	:STORE RKDA
3080	022054	013762	002010	000006	MOV	CONFIG,\$GDDAT	:PREPARE EXPECTED RESULTS
3081	022062	016237	000006	001126	BIC	#174340,\$GDDAT	:INITIALIZE READ ONLY BITS
3082	022070	013737	002010	001124	CMP	\$GDDAT,\$BDDAT	:CHECK IF RKDA CORRECT
3083	022076	042737	174340	001124	BEQ	2\$:YES, TEST IF ANY OTHER REG MODIFIED
3084	022104	023737	001124	001126	MOV	#EM1003,EM3N+2	:LOAD ERROR MESSAGE
3085	022112	001404			ERROR	3	

2\$: MOV RKCS1(R2),SBDDAT :STORE COMMAND AND STATUS REG. 1

3088	022124	016237	000000	001126	CMP	#RDY,\$BDDAT	:CHECK IF CSI CORRECT
3089	022124	022737	000200	001126	BEQ	3\$:YES, CONTINUE
3090	022132	022737	000200	001126	MOV	#IR,\$GDDAT	:LOAD EXPECTED RESULTS
3091	022140	001407			MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
3092	022142	012737	000100	001124	ERROR	3	
3093	022150	012737	067324	001322			
3094	022156	104003					

3\$: MOV RKBA(R2),SBDDAT :STORE BUS AND REG

3095	022160	016237	000004	001126	BEQ	4\$:CHECK IF ZERO
3096	022160	001406					
3097	022166	001406					

3098	022170	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3099	022174	012737	067375	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
3100	022202	104003			ERROR	3	
3101	022204				45:		
3102	022204	016237	000002	001126	MOV	RKWC(R2), \$BDDAT	;STORE WORD COUNT REG
3103	022212	001406			BEQ	55	;CHECK IF ZERO
3104	022214	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3105	022220	012737	067345	001322	MOV	#EM1018 EM3N+2	;LOAD ERROR MESSAGE
3106	022226	104003			ERROR	3	
3107	022230				55:		
3108	022230	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
3109	022236	001406			BEQ	75	;CHECK IF ZERO
3110	022240	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3111	022244	012737	067566	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
3112	022252	104003			ERROR	3	
3113	022254				75:		
3114	022254	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2
3115	022262	022737	000100	001126	CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
3116	022270	001407			BEQ	85	;YES, CONTINUE
3117	022272	012737	000100	001124	MOV	#IR \$GDDAT	;LOAD EXPECTED CONTENTS
3118	022300	012737	067460	001322	MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
3119	022306	104003			ERROR	3	
3120	022310				85:		
3121	022310	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
3122	022316	001406			BEQ	95	;CHECK IF ZERO
3123	022320	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3124	022324	012737	067501	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
3125	022332	104003			ERROR	3	
3126	022334	016237	000014	001126	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
3127	022342	001406			BEQ	105	;CHECK IF ZERO
3128	022344	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3129	022350	012737	067537	001322	MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
3130	022356	104003			ERROR	3	
3131	022360				105:		
3132	022360	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
3133	022366	001406			BEQ	125	;CHECK IF ZERO
3134	022370	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3135	022374	012737	067634	001322	MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
3136	022402	104003			ERROR	3	
3137	022404				125:		
3138	022404	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
3139	022412	012737	002000	001124	MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MRI
3140	022420	032737	020000	001126	BIT	#ECCW, \$BDDAT	
3141	022426	001403			BEQ	135	
3142	022430	052737	020000	001124	BIS	#ECCW, \$GDDAT	
3143	022436	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
3144	022444	001404			BEQ	145	;YES, ISSUE CONTROLLER CLEAR
3145	022446	012737	067661	001322	MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
3146	022454	104003			ERROR	3	
3147	022456				145:		
3148	022456	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
3149	022464	001406			BEQ	155	;CHECK IF ZERO
3150	022466	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3151	022472	012737	067737	001322	MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
3152	022500	104003			ERROR	3	
3153	022502	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.

E07

```

3154 022510 012737 004066 001124 16$: MOV #4066,$GDDAT ;USE 4066
3155 022516 023737 001124 001126 17$: CMP $GDDAT,$BDDAT ;CHECK IF ECC POSITION CORRECT
3156 022524 001404 BEQ 18$ ;YES,INITIALIZE RK611
3157 022526 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3158 022534 104003 ERROR 3
3159 022536 016237 000006 002014 18$: MOV RKDA(R2),PREREG ;GET PREVIOUS CONTENTS
3160 022544 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3161 022552 016237 000006 001126 MOV RKDA(R2),$BDDAT ;GET CURRENT VALUE
3162 022560 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
3163 022564 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKDA CORRECT
3164 022572 001407 BEQ 19$ ;YES,CHECK IF FINISHED
3165 022574 012737 063413 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
3166 022602 012737 066671 001332 MOV #EM1003,EM4N+2
3167 022610 104004 ERROR 4
3168 022612 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
3169 022614 000261 SEC ;SHIFT IN ONE
3170 022616 006137 002010 ROL CONFIG
3171 022622 005301 DEC R1 ;CHECK IF FINISHED
3172 022624 001402 BEQ TST27 ;;YES,GO ON TO NEXT TEST
3173 022626 000137 022050 JMP 1$
3174
3175
3176 *****
3177 *TEST 27 REGISTER INTERACTION USING DISK ADDRESS (PART 3)
3178 *
3179 * ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
3180 * WRITE THE WORD COUNT REGISTER WITH 0.
3181 *
3182 * WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
3183 * THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
3184 * SURE NO INTERACTION TAKES PLACE.
3185 *
3186 * 000001 000037 000777 017777 000000
3187 * 000003 000077 001777 037777
3188 * 000007 000177 003777 077777
3189 * 000017 000377 007777 177777
3190 *****
3191 022632 000004 TST27: SCOPE
3192 022634 012737 000764 001200 MOV #500,$TIMES ;;DO 500. ITERATIONS
3193 022642 012701 000021 MOV #17,R1 ;LOAD NUMBER OF PATTERNS
3194 022646 005037 002010 CLR CONFIG ;LOAD INITIAL CONFIGURATION
3195 022652 012737 063732 001320 MOV #EM8,EM3N ;LOAD ERROR MESSAGE
3196 022660 012737 100000 000000 MOV #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
3197 022666 012737 022674 001110 MOV #1$,SLPERR ;LOAD LOOP ON ERROR LOCATION FOR
3198 ; SUBTEST LOOP
3199
3200 022674 1$:
3201 022674 005062 000002 CLR RKWC(R2) ;CLEAR WORD COUNT REG.
3202 022700 013762 002010 000006 MOV CONFIG,RKDA(R2) ;WRITE RKDA
3203 022706 016237 000006 001126 MOV RKDA(R2),$BDDAT ;STORE RKDA
3204 022714 013737 002010 001124 MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
3205 022722 042737 174340 001124 BIC #174340,$GDDAT ;INITIALIZE READ ONLY BITS
3206 022730 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKDA CORRECT
3207 022736 001404 BEQ 2$ ;YES,TEST IF ANY OTHER REG MODIFIED
3208 022740 012737 066671 001322 MOV #EM1003,EM3N+2 ;LOAD ERROR MESSAGE
3209 022746 104003 ERROR 3

```

F07

3210	022750				25:				
3211	022750	016237	000000	001126		MOV	RKCS1(R2), \$BDDAT	: STORE COMMAND AND STATUS REG. 1	
3212	022756	022737	000200	001126		CMP	#RDY, \$BDDAT	: CHECK IF CS1 CORRECT	
3213	022764	001407				BEQ	3\$: YES, CONTINUE	
3214	022766	012737	000100	001124		MOV	#IR, \$GDDAT	: LOAD EXPECTED RESULTS	
3215	022774	012737	067324	001322		MOV	#EM1017, EM3N+2	: LOAD ERROR MESSAGE	
3216	023002	104003				ERROR	3		
3217	023004				35:				
3218	023004	016237	000004	001126		MOV	RKBA(R2), \$BDDAT	: STORE BUS AND REG	
3219	023012	001406				BEQ	4\$: CHECK IF ZERO	
3220	023014	005037	001124			CLR	\$GDDAT	: LOAD EXPECTED CONTENTS	
3221	023020	012737	067375	001322		MOV	#EM1019, EM3N+2	: LOAD ERROR MESSAGE	
3222	023026	104003				ERROR	3		
3223	023030				45:				
3224	023030	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	: STORE WORD COUNT REG	
3225	023036	001406				BEQ	5\$: CHECK IF ZERO	
3226	023040	005037	001124			CLR	\$GDDAT	: LOAD EXPECTED CONTENTS	
3227	023044	012737	067345	001322		MOV	#EM1018, EM3N+2	: LOAD ERROR MESSAGE	
3228	023052	104003				ERROR	3		
3229	023054				55:				
3230	023054	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	: STORE ATTENTION SUMMARY/OFFSET REG.	
3231	023062	001406				BEQ	7\$: CHECK IF ZERO	
3232	023064	005037	001124			CLR	\$GDDAT	: LOAD EXPECTED CONTENTS	
3233	023070	012737	067566	001322		MOV	#EM1024, EM3N+2	: LOAD ERROR MESSAGE	
3234	023076	104003				ERROR	3		
3235	023100				75:				
3236	023100	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	: STORE COMMAND AND STATUS REG. 2	
3237	023106	022737	000100	001126		CMP	#IR, \$BDDAT	: CHECK IF CS2 CORRECT	
3238	023114	001407				BEQ	8\$: YES, CONTINUE	
3239	023116	012737	000100	001124		MOV	#IR, \$GDDAT	: LOAD EXPECTED CONTENTS	
3240	023124	012737	067460	001322		MOV	#EM1021, EM3N+2	: LOAD ERROR MESSAGE	
3241	023132	104003				ERROR	3		
3242	023134				85:				
3243	023134	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	: STORE DRIVE STATUS REG	
3244	023142	001406				BEQ	9\$: CHECK IF ZERO	
3245	023144	005037	001124			CLR	\$GDDAT	: LOAD EXPECTED CONTENTS	
3246	023150	012737	067501	001322		MOV	#EM1022, EM3N+2	: LOAD ERROR MESSAGE	
3247	023156	104003				ERROR	3		
3248	023160	016237	000014	001126	95:				
3249	023166	001406				MOV	RKER(R2), \$BDDAT	: STORE ERROR REG	
3250	023170	005037	001124			BEQ	10\$: CHECK IF ZERO	
3251	023174	012737	067537	001322		CLR	\$GDDAT	: LOAD EXPECTED CONTENTS	
3252	023202	104003				MOV	#EM1023, EM3N+2	: LOAD ERROR MESSAGE	
3253	023204				105:				
3254	023204	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	: STORE CYLINDER ADD REG	
3255	023212	001406				BEQ	12\$: CHECK IF ZERO	
3256	023214	005037	001124			CLR	\$GDDAT	: LOAD EXPECTED CONTENTS	
3257	023220	012737	067634	001322		MOV	#EM1025, EM3N+2	: LOAD ERROR MESSAGE	
3258	023226	104003				ERROR	3		
3259	023230				125:				
3260	023230	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	: STORE MAINTENANCE REG. 1	
3261	023236	012737	002000	001124		MOV	#MEWD, \$GDDAT	: LOAD EXPECTED MRI	
3262	023244	032737	020000	001126		BIT	#ECCW, \$BDDAT		
3263	023252	001403				BEQ	13\$		
3264	023254	052737	020000	001124		BIS	#ECCW, \$GDDAT		
3265	023262	023737	001124	001126	135:	CMP	\$GDDAT, \$BDDAT	: CHECK IF MR1 CORRECT	

G07

3266	023270	001404				BEQ	14\$:YES,ISSUE CONTROLLER CLEAR
3267	023272	012737	067661	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
3268	023300	104003				ERROR	3	
3269	023302				14\$:			
3270	023302	016237	000032	001126		MOV	RKECPT(R2),SBDDAT	:STORE ECC PATTERN REG.
3271	023310	001406				BEQ	15\$:CHECK IF ZERO
3272	023312	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3273	023316	012737	067737	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
3274	023324	104003				ERROR	3	
3275	023326	016237	000030	001126	15\$:	MOV	RKECPS(R2) SBDDAT	:STORE ECC POSITION REG.
3276	023334	012737	004066	001124	16\$:	MOV	#4066,\$GDDAT	:USE 4066
3277	023342	023737	001124	001126	17\$:	CMP	\$GDDAT,\$BDDAT	:CHECK IF ECC POSITION CORRECT
3278	023350	001404				BEQ	18\$:YES,INITIALIZE RK611
3279	023352	012737	067712	001322		MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
3280	023360	104003				ERROR	3	
3281	023362	016237	000006	002014	18\$:	MOV	RKDA(R2),PREREG	:GET PREVIOUS CONTENTS
3282	023370	012762	100000	000000		MOV	#CCLR,RKCSI(R2)	:CLEAR RK611 CONTROLLER
3283	023376	016237	000006	001126		MOV	RKDA(R2),\$BDDAT	:GET CURRENT VALUE
3284	023404	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3285	023410	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:CHECK IF RKDA CORRECT
3286	023416	001407				BEQ	19\$:YES,CHECK IF FINISHED
3287	023420	012737	063413	001330		MOV	#EM3,EM4N	:LOAD ERROR MESSAGE
3288	023426	012737	066671	001332		MOV	#EM1003,EM4N+2	
3289	023434	104004				ERROR	4	
3290	023436	104415			19\$:	SCOPI		:CHECK IF LOOP ON ERROR
3291	023440	000261				SEC		:SHIFT IN ONE
3292	023442	006137	002010			ROL	CONFIG	
3293	023446	005301				DEC	R1	:CHECK IF FINISHED
3294	023450	001402				BEQ	TST30	:;YES,GO ON TO NEXT TEST
3295	023452	000137	022674			JMP	1\$	
3296								
3297								
3298								
3299								
3300								
3301								
3302								
3303								
3304								
3305								
3306								
3307								
3308								
3309								
3310								
3311								
3312								
3313	023456	000004				TST30:	SCOPE	
3314	023460	012737	000764	001200		MOV	#500,\$TIMES	:DO 500. ITERATIONS
3315	023466	012701	000021			MOV	#17,R1	:LOAD NUMBER OF PATTERNS
3316	023472	005037	002010			CLR	CONFIG	:LOAD INITIAL CONFIGURATION
3317	023476	012737	063732	001320		MOV	#EM8,EM3N	:LOAD ERROR MESSAGE
3318	023504	012737	100000	000000		MOV	#CCLR,RKCSI	:CLEAR RK611 WITH CONTROLLER CLEAR
3319	023512	012737	023520	001110		MOV	#1\$,SLPERR	:LOAD LOOP ON ERROR LOCATION FOR
3320								:SUBTEST LOOP
3321								

 :TEST 30 REGISTER INTERACTION USING DISK ADDRESS (PART 4)
 :*

:* 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.
 :*

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

 :TST30: SCOPE
 :*

H07

RA611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 63
 DZR6A8.P11 T30 REGISTER INTERACTION USING DISK ADDRESS (PART 4)

SEQ 0064

3322	023520				15:				
3323	023520	005062	000002			CLR	RKWC(R2)	; CLEAR WORD COUNT REG.	
3324	023524	013762	002010	000006		MOV	CONFIG,RKDA(R2)	; WRITE RKDA	
3325	023532	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	; STORE RKDA	
3326	023540	013737	002010	001124		MOV	CONFIG, \$GDDAT	; PREPARE EXPECTED RESULTS	
3327	023546	042737	174340	001124		BIC	#174340, \$GDDAT	; INITIALIZE READ ONLY BITS	
3328	023554	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; CHECK IF RKDA CORRECT	
3329	023562	001404				BEQ	25	; YES, TEST IF ANY OTHER REG MODIFIED	
3330	023564	012737	066671	001322		MOV	#EM1003, EM3N+2	; LOAD ERROR MESSAGE	
3331	023572	104003				ERROR	3		
3332	023574				25:				
3333	023574	016237	000000	001126		MOV	RKCS1(R2), \$BDDAT	; STORE COMMAND AND STATUS REG. 1	
3334	023602	022737	000200	001126		CMP	#RDY, \$BDDAT	; CHECK IF CS1 CORRECT	
3335	023610	001407				BEQ	35	; YES, CONTINUE	
3336	023612	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED RESULTS	
3337	023620	012737	067324	001322		MOV	#EM1017, EM3N+2	; LOAD ERROR MESSAGE	
3338	023626	104003				ERROR	3		
3339	023630				35:				
3340	023630	016237	000004	001126		MOV	RKBA(R2), \$BDDAT	; STORE BUS AND REG	
3341	023636	001406				BEQ	45	; CHECK IF ZERO	
3342	023640	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
3343	023644	012737	067375	001322		MOV	#EM1019, EM3N+2	; LOAD ERROR MESSAGE	
3344	023652	104003				ERROR	3		
3345	023654				45:				
3346	023654	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	; STORE WORD COUNT REG	
3347	023662	001406				BEQ	55	; CHECK IF ZERO	
3348	023664	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
3349	023670	012737	067345	001322		MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE	
3350	023676	104003				ERROR	3		
3351	023700				55:				
3352	023700	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG.	
3353	023706	001406				BEQ	75	; CHECK IF ZERO	
3354	023710	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
3355	023714	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE	
3356	023722	104003				ERROR	3		
3357	023724				75:				
3358	023724	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG.2	
3359	023732	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT	
3360	023740	001407				BEQ	85	; YES, CONTINUE	
3361	023742	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS	
3362	023750	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE	
3363	023756	104003				ERROR	3		
3364	023760				85:				
3365	023760	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG	
3366	023766	001406				BEQ	95	; CHECK IF ZERO	
3367	023770	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
3368	023774	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE	
3369	024002	104003				ERROR	3		
3370	024004	016237	000014	001126	95:				
3371	024012	001406				MOV	RKER(R2), \$BDDAT	; STORE ERROR REG	
3372	024014	005037	001124			BEQ	105	; CHECK IF ZERO	
3373	024020	012737	067537	001322		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS	
3374	024026	104003				MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE	
3375	024030					ERROR	3		
3376	024030	016237	000020	001126	105:				
3377	024036	001406				MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG	
						BEQ	125	; CHECK IF ZERO	

```

3378 024040 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
3379 024044 012737 067634 001322 MOV #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
3380 024052 104003 ERROR 3
3381 024054 125:
3382 024054 016237 000026 001126 MOV RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG.1
3383 024062 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED MRI
3384 024070 032737 020000 001126 BIT #ECCW, $BDDAT
3385 024076 001403 BEQ 135
3386 024100 052737 020000 001124 BIS #ECCW, $GDDAT
3387 024106 023737 001124 001126 135: CMP $GDDAT, $BDDAT ;CHECK IF MRI CORRECT
3388 024114 001404 BEQ 145 ;YES, ISSUE CONTROLLER CLEAR
3389 024116 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
3390 024124 104003 ERROR 3
3391 024126 145:
3392 024126 016237 000032 001126 MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
3393 024134 001406 BEQ 155 ;CHECK IF ZERO
3394 024136 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
3395 024142 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
3396 024150 104003 ERROR 3
3397 024152 016237 000030 001126 155: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
3398 024160 012737 004066 001124 165: MOV #4066, $GDDAT ;USE 4066
3399 024166 023737 001124 001126 175: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
3400 024174 001404 BEQ 185 ;YES, INITIALIZE RK611
3401 024176 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3402 024204 104003 ERROR 3
3403 024206 016237 000006 002014 185: MOV RKDA(R2), PREREG ;GET PREVIOUS CONTENTS
3404 024214 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
3405 024222 016237 000006 001126 MOV RKDA(R2), $BDDAT ;GET CURRENT VALUE
3406 024230 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
3407 024234 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKDA CORRECT
3408 024242 001407 BEQ 195 ;YES, CHECK IF FINISHED
3409 024244 012737 063413 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
3410 024252 012737 066671 001332 MOV #EM1003,EM4N+2
3411 024260 104004 ERROR 4
3412 024262 104415 195: SCOP1 ;CHECK IF LOOP ON ERROR
3413 024264 000261 SEC ;SHIFT IN ONE
3414 024266 006137 002010 ROL CONFIG
3415 024272 005301 DEC R1 ;CHECK IF FINISHED
3416 024274 001402 BEQ TST31 ;;YES, GO ON TO NEXT TEST
3417 024276 000137 023520 JMP 15

```

```

3418
3419
3420 :*****
3421 :*TEST 31 REGISTER INTERACTION USING ATTN/OFFSET (PART 1)
3422 :*
3423 :* ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
3424 :* WRITE THE WORD COUNT REGISTER WITH 0.
3425 :*
3426 :* WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
3427 :* WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
3428 :* MAKE SURE NO INTERACTION TAKES PLACE.
3429 :*
3430 :* 000000 000010 000200 004000 100000
3431 :* 000001 000020 000400 010000
3432 :* 000002 000040 001000 020000
3433 :* 000004 000100 002000 040000

```

```

3434
3435 024302 000004
3436 024304 012737 000764 001200
3437 024312 012701 000021
3438 024316 012737 000001 002010
3439 024324 012737 064361 001320
3440 024332 012737 100000 000000
3441 024340 012737 024346 001110
3442
3443
3444 024346
3445 024346 005062 000002
3446 024352 013762 002010 000016
3447 024360 016237 000016 001126
3448 024366 013737 002010 001124
3449 024374 042737 177400 001124
3450 024402 023737 001124 001126
3451 024410 001404
3452 024412 012737 067011 001322
3453 024420 104003
3454 024422
3455 024422 016237 000000 001126
3456 024430 022737 000200 001126
3457 024436 001407
3458 024440 012737 000100 001124
3459 024446 012737 067324 001322
3460 024454 104003
3461 024456
3462 024456 016237 000004 001126
3463 024464 001406
3464 024466 005037 001124
3465 024472 012737 067375 001322
3466 024500 104003
3467 024502
3468 024502 016237 000002 001126
3469 024510 001406
3470 024512 005037 001124
3471 024516 012737 067345 001322
3472 024524 104003
3473 024526
3474 024526 016237 000006 001126
3475 024534 001406
3476 024536 005037 001124
3477 024542 012737 067422 001322
3478 024550 104003
3479 024552
3480 024552 016237 000010 001126
3481 024560 022737 000100 001126
3482 024566 001407
3483 024570 012737 000100 001124
3484 024576 012737 067460 001322
3485 024604 104003
3486 024606
3487 024606 016237 000012 001126
3488 024614 001406
3489 024616 005037 001124

:*****
†ST31: SCOPE
MOV #500, $TIMES ; DO 500. ITERATIONS
MOV #17, R1 ; LOAD NUMBER OF PATTERNS
MOV #000001, CONFIG ; LOAD INITIAL CONFIGURATION
MOV #EM17, EM3N ; LOAD ERROR MESSAGE
MOV #CCLR, RKCS1 ; CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1$, $LPERR ; LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1$:
CLR RKWC(R2) ; CLEAR WORD COUNT REG.
MOV CONFIG, RKASOF(R2) ; WRITE RKASOF
MOV RKASOF(R2), $BDDAT ; STORE RKASOF
MOV CONFIG, $GDDAT ; PREPARE EXPECTED RESULTS
BIC #177400, $GDDAT ; INITIALIZE READ ONLY BITS
CMP $GDDAT, $BDDAT ; CHECK IF RKASOF CORRECT
BEQ 2$ ; YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1007, 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 #IR, $GDDAT ; LOAD EXPECTED RESULTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

3$:
MOV RKBA(R2), $BDDAT ; STORE BUS AND REG
BEQ 4$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

4$:
MOV RKWC(R2), $GDDAT ; STORE WORD COUNT REG
BEQ 5$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

5$:
MOV RKDA(R2), $BDDAT ; STORE DISK AVERAGE REG
BEQ 6$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

6$:
MOV RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG.2
CMP #IR, $BDDAT ; CHECK IF CS2 CORRECT
BEQ 8$ ; YES, CONTINUE
MOV #IR, $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1021, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

8$:
MOV RKDS(R2), $BDDAT ; STORE DRIVE STATUS REG
BEQ 9$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS

```

K07

3490	024622	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
3491	024630	104003				ERROR	3	
3492	024632	016237	000014	001126	9%:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
3493	024640	001406				BEQ	10\$;CHECK IF ZERO
3494	024642	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3495	024646	012737	067537	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
3496	024654	104003				ERROR	3	
3497	024656				10%:			
3498	024656	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
3499	024664	001406				BEQ	12\$;CHECK IF ZERO
3500	024666	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3501	024672	012737	067634	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
3502	024700	104003				ERROR	3	
3503	024702				12%:			
3504	024702	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
3505	024710	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MR1
3506	024716	032737	020000	001126		BIT	#ECCW, \$BDDAT	
3507	024724	001403				BEQ	13\$	
3508	024726	052737	020000	001124		BIS	#ECCW, \$GDDAT	
3509	024734	023737	001124	001126	13%:	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
3510	024742	001404				BEQ	14\$;YES, ISSUE CONTROLLER CLEAR
3511	024744	012737	067661	001322		MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
3512	024752	104003				ERROR	3	
3513	024754				14%:			
3514	024754	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
3515	024762	001406				BEQ	15\$;CHECK IF ZERO
3516	024764	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3517	024770	012737	067737	001322		MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
3518	024776	104003				ERROR	3	
3519	025000	016237	000030	001126	15%:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
3520	025006	012737	004066	001124	16%:	MOV	#4066, \$GDDAT	;USE 4066
3521	025014	023737	001124	001126	17%:	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT
3522	025022	001404				BEQ	18\$;YES, INITIALIZE RK611
3523	025024	012737	067712	001322		MOV	#EM1029,EM3N+2	;LOAD ERROR MESSAGE
3524	025032	104003				ERROR	3	
3525	025034	016237	000016	002014	18%:	MOV	RKASOF(R2), PREREG	;GET PREVIOUS CONTENTS
3526	025042	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	;CLEAR RK611 CONTROLLER
3527	025050	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	;GET CURRENT VALUE
3528	025056	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3529	025062	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	;CHECK IF RKASOF CORRECT
3530	025070	001407				BEQ	19\$;YES, CHECK IF FINISHED
3531	025072	012737	063413	001330		MOV	#EM3,EM4N	;LOAD ERROR MESSAGE
3532	025100	012737	067011	001332		MOV	#EM1007,EM4N+2	
3533	025106	104004				ERROR	4	
3534	025110	104415			19%:	SCOPI		;CHECK IF LOOP ON ERROR
3535	025112	000241				CLC		;SHIFT IN ZERO
3536	025114	006137	002010			ROL	CONFIG	
3537	025120	005301				DEC	R1	;CHECK IF FINISHED
3538	025122	001402				BEQ	TST32	;YES, GO ON TO NEXT TEST
3539	025124	000137	024346			JMP	1\$	

```

3540
3541 *****
3542 *TEST 32 REGISTER INTERACTION USING ATTN/OFFSET (PART 2)
3543 *
3544 * ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
3545 * WRITE THE WORD COUNT REGISTER WITH 0.
  
```



```

3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557 025130 000004
3558 025132 012737 000764 001200
3559 025140 012701 000021
3560 025144 012737 177776 002010
3561 025152 012737 064361 001320
3562 025160 012737 100000 000000
3563 025166 012737 025174 001110
3564
3565
3566 025174
3567 025174 005062 000002
3568 025200 013762 002010 000016
3569 025206 016237 000016 001126
3570 025214 013737 002010 001124
3571 025222 042737 177400 001124
3572 025230 023737 001124 001126
3573 025236 001404
3574 025240 012737 067011 001322
3575 025246 104003
3576 025250
3577 025250 016237 000000 001126
3578 025256 022737 000200 001126
3579 025264 001407
3580 025266 012737 000100 001124
3581 025274 012737 067324 001322
3582 025302 104003
3583 025304
3584 025304 016237 000004 001126
3585 025312 001406
3586 025314 005037 001124
3587 025320 012737 067375 001322
3588 025326 104003
3589 025330
3590 025330 016237 000002 001126
3591 025336 001406
3592 025340 005037 001124
3593 025344 012737 067345 001322
3594 025352 104003
3595 025354
3596 025354 016237 000006 001126
3597 025362 001406
3598 025364 005037 001124
3599 025370 012737 067422 001322
3600 025376 104003
3601 025400

```

```

:
:
: 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
:
: *****
: ST32: SCOPE
: MOV #500,$TIMES ;;DO 500. ITERATIONS
: MOV #17,R1 ;;LOAD NUMBER OF PATTERNS
: MOV #177776,CONFIG ;;LOAD INITIAL CONFIGURATION
: MOV #EM17,EM3N ;;LOAD ERROR MESSAGE
: MOV #CCLR,RKCS1 ;;CLEAR RK611 WITH CONTROLLER CLEAR
: MOV #1$, $LPERR ;;LOAD LOOP ON ERROR LOCATION FOR
: ; SUBTEST LOOP
:
: 1$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
: MOV CONFIG,RKASOF(R2) ;WRITE RKASOF
: MOV RKASOF(R2), $BDDAT ;STORE RKASOF
: MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
: BIC #177400,$GDDAT ;INITIALIZE READ ONLY BITS
: CMP $GDDAT,$BDDAT ;CHECK IF RKASOF CORRECT
: BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
: MOV #EM1007,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 #IR,$GDDAT ;LOAD EXPECTED RESULTS
: MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
: ERROR 3
:
: 3$: MOV RKBA(R2), $BDDAT ;STORE BUS AND REG
: BEQ 4$ ;CHECK IF ZERO
: CLR $GDDAT ;LOAD EXPECTED CONTENTS
: MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
: ERROR 3
:
: 4$: MOV RKWC(R2), $BDDAT ;STORE WORD COUNT REG
: BEQ 5$ ;CHECK IF ZERO
: CLR $GDDAT ;LOAD EXPECTED CONTENTS
: MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
: ERROR 3
:
: 5$: MOV RKDA(R2), $BDDAT ;STORE DISK AVERAGE REG
: BEQ 6$ ;CHECK IF ZERO
: CLR $GDDAT ;LOAD EXPECTED CONTENTS
: MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
: ERROR 3
:
: 6$:

```

3602	025400	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
3603	025406	022737	000100	001126		CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
3604	025414	001407				BEQ	3\$:YES, CONTINUE
3605	025416	012737	0001C0	001124		MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
3606	025424	012737	067460	001322		MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
3607	025432	104003				ERROR	3	
3608	025434				8\$:			
3609	025434	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
3610	025442	001406				BEQ	9\$:CHECK IF ZERO
3611	025444	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3612	025450	012737	067501	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
3613	025456	104003				ERROR	3	
3614	025460	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
3615	025466	001406				BEQ	10\$:CHECK IF ZERO
3616	025470	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3617	025474	012737	067537	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
3618	025502	104003				ERROR	3	
3619	025504				10\$:			
3620	025504	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
3621	025512	001406				BEQ	12\$:CHECK IF ZERO
3622	025514	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3623	025520	012737	067634	001322		MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
3624	025526	104003				ERROR	3	
3625	025530				12\$:			
3626	025530	016237	000026	001126		MOV	RKMRI(R2), \$BDDAT	:STORE MAINTENANCE REG.1
3627	025536	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MRI
3628	025544	032737	020000	001126		BIT	#ECCW, \$BDDAT	
3629	025552	001403				BEQ	13\$	
3630	025554	052737	020000	001124		BIS	#ECCW, \$GDDAT	
3631	025562	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MRI CORRECT
3632	025570	001404				BEQ	14\$:YES, ISSUE CONTROLLER CLEAR
3633	025572	012737	067661	001322		MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
3634	025600	104003				ERROR	3	
3635	025602				14\$:			
3636	025602	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
3637	025610	001406				BEQ	15\$:CHECK IF ZERO
3638	025612	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3639	025616	012737	067737	001322		MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
3640	025624	104003				ERROR	3	
3641	025626	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
3642	025634	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	:USE 4066
3643	025642	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
3644	025650	001404				BEQ	18\$:YES, INITIALIZE RK611
3645	025652	012737	067712	001322		MOV	#EM1029, EM3N+2	:LOAD ERROR MESSAGE
3646	025660	104003				ERROR	3	
3647	025662	016237	000016	002014	18\$:	MOV	RKASOF(R2), PREREG	:GET PREVIOUS CONTENTS
3648	025670	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
3649	025676	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	:GET CURRENT VALUE
3650	025704	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3651	025710	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	:CHECK IF RKASOF CORRECT
3652	025716	001407				BEQ	19\$:YES, CHECK IF FINISHED
3653	025720	012737	063413	001330		MOV	#EM3, EM4N	:LOAD ERROR MESSAGE
3654	025726	012737	067011	001332		MOV	#EM1007, EM4N+2	
3655	025734	104004				ERROR	4	
3656	025736	104415			19\$:	SCOP1		:CHECK IF LOOP ON ERROR
3657	025740	000261				SEC		:SHIFT IN ONE

3658 025742 006137 002010
3659 025746 005301
3660 025750 001402
3661 025752 000137 025174

RUL CONFIG
DEC R1 ;CHECK IF FINISHED
BEQ TST33 ;;YES, GO ON TO NEXT TEST
JMP 1\$

*TEST 33 REGISTER INTERACTION USING ATTN/OFFSET (PART 3)

* ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
* 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.

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

TST33: SCOPE
MOV #500, \$TIMES ;; DO 500. ITERATIONS
MOV #17, R1 ;; LOAD NUMBER OF PATTERNS
CLR CONFIG ;; LOAD INITIAL CONFIGURATION
MOV #EM17, EM3N ;; LOAD ERROR MESSAGE
MOV #CCLR, RKCS1 ;; CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1\$, \$LPERR ;; LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

3679 025756 000004
3680 025760 012737 000764 001200
3681 025766 012701 000021
3682 025772 005037 002010
3683 025776 012737 064361 001320
3684 026004 012737 100000 000000
3685 026012 012737 026020 001110

1\$: CLR RKWC(R2) ; CLEAR WORD COUNT REG.
MOV CONFIG, RKASOF(R2) ; WRITE RKASOF
MOV RKASOF(R2), \$BDDAT ; STORE RKASOF
MOV CONFIG, \$GDDAT ; PREPARE EXPECTED RESULTS
BIC #177400, \$GDDAT ; INITIALIZE READ ONLY BITS
CMP \$GDDAT, \$BDDAT ; CHECK IF RKASOF CORRECT
BEQ 2\$; YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1007, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

3688 026020
3689 026020 005062 000002
3690 026024 013762 002010 000016
3691 026032 016237 000016 001126
3692 026040 013737 002010 001124
3693 026046 042737 177400 001124
3694 026054 023737 001124 001126
3695 026062 001404
3696 026064 012737 067011 001322
3697 026072 104003

2\$: MOV RKCS1(R2), \$BDDAT ; STORE COMMAND AND STATUS REG. 1
CMP #RDY, \$BDDAT ; CHECK IF CSI CORRECT
BEQ 3\$; YES, CONTINUE
MOV #IR, \$GDDAT ; LOAD EXPECTED RESULTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

3698 026074
3699 026074 016237 000000 001126
3700 026102 022737 000200 001126
3701 026110 001407
3702 026112 012737 000100 001124
3703 026120 012737 067324 001322
3704 026126 104003

3\$: MOV RKBA(R2), \$BDDAT ; STORE BUS AND REG
BEQ 4\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

3705 026130
3706 026130 016237 000004 001126
3707 026136 001406
3708 026140 005037 001124
3709 026144 012737 067375 001322
3710 026152 104003

4\$: MOV RKWC(R2), \$BDDAT ; STORE WORD COUNT REG
BEQ 5\$; CHECK IF ZERO

3711 026154
3712 026154 016237 000002 001126
3713 026162 001406

3714	026164	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
3715	026170	012737	067345	001322	MOV	#EM1018,EM3N+2		;LOAD ERROR MESSAGE
3716	026176	104003			ERROR	3		
3717	026200						5\$:	
3718	026200	016237	000006	001126	MOV	RKDA(R2), \$BDDAT		;STORE DISK AVERAGE REG
3719	026206	001406			BEQ	6\$;CHECK IF ZERO
3720	026210	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
3721	026214	012737	067422	001322	MOV	#EM1020,EM3N+2		;LOAD ERROR MESSAGE
3722	026222	104003			ERROR	3		
3723	026224						6\$:	
3724	026224	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT		;STORE COMMAND AND STATUS REG.2
3725	026232	022737	000100	001126	CMP	#IR, \$BDDAT		;CHECK IF CS2 CORRECT
3726	026240	001407			BEQ	8\$;YES, CONTINUE
3727	026242	012737	000100	001124	MOV	#IR, \$GDDAT		;LOAD EXPECTED CONTENTS
3728	026250	012737	067460	001322	MOV	#EM1021,EM3N+2		;LOAD ERROR MESSAGE
3729	026256	104003			ERROR	3		
3730	026260						9\$:	
3731	026260	016237	000012	001126	MOV	RKDS(R2), \$BDDAT		;STORE DRIVE STATUS REG
3732	026266	001406			BEQ	9\$;CHECK IF ZERO
3733	026270	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
3734	026274	012737	067501	001322	MOV	#EM1022,EM3N+2		;LOAD ERROR MESSAGE
3735	026302	104003			ERROR	3		
3736	026304	016237	000014	001126	MOV	RKER(R2), \$BDDAT		;STORE ERROR REG
3737	026312	001406			BEQ	10\$;CHECK IF ZERO
3738	026314	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
3739	026320	012737	067537	001322	MOV	#EM1023,EM3N+2		;LOAD ERROR MESSAGE
3740	026326	104003			ERROR	3		
3741	026330						10\$:	
3742	026330	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT		;STORE CYLINDER ADD REG
3743	026336	001406			BEQ	12\$;CHECK IF ZERO
3744	026340	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
3745	026344	012737	067634	001322	MOV	#EM1025,EM3N+2		;LOAD ERROR MESSAGE
3746	026352	104003			ERROR	3		
3747	026354						12\$:	
3748	026354	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT		;STORE MAINTENANCE REG.1
3749	026362	012737	002000	001124	MOV	#MEMD, \$GDDAT		;LOAD EXPECTED MRI
3750	026370	032737	020000	001126	BIT	#ECCW, \$BDDAT		
3751	026376	001403			BEQ	13\$		
3752	026400	052737	020000	001124	BIS	#ECCW, \$GDDAT		
3753	026406	023737	001124	001126	CMP	\$GDDAT, \$BDDAT		;CHECK IF MRI CORRECT
3754	026414	001404			BEQ	14\$;YES, ISSUE CONTROLLER CLEAR
3755	026416	012737	067661	001322	MOV	#EM1026,EM3N+2		;LOAD ERROR MESSAGE
3756	026424	104003			ERROR	3		
3757	026426						14\$:	
3758	026426	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT		;STORE ECC PATTERN REG.
3759	026434	001406			BEQ	15\$;CHECK IF ZERO
3760	026436	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
3761	026442	012737	067737	001322	MOV	#EM1030,EM3N+2		;LOAD ERROR MESSAGE
3762	026450	104003			ERROR	3		
3763	026452	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT		;STORE ECC POSITION REG.
3764	026460	012737	004066	001124	MOV	#4065, \$GDDAT		;USE 4066
3765	026466	023737	001124	001126	CMP	\$GDDAT, \$BDDAT		;CHECK IF ECC POSITION CORRECT
3766	026474	001404			BEQ	18\$;YES, INITIALIZE RK611
3767	026476	012737	067712	001322	MOV	#EM1029,EM3N+2		;LOAD ERROR MESSAGE
3768	026504	104003			ERROR	3		
3769	026506	016237	000016	002014	MOV	RKASOF(R2), PREREG		;GET PREVIOUS CONTENTS

```

3770 026514 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3771 026522 016237 000016 001126      MOV      RKASOF(R2),%BDDAT ;GET CURRENT VALUE
3772 026530 005037 001124      CLR      %GDAT ;LOAD EXPECTED CONTENTS
3773 026534 023737 001124 001126      CMP      %GDAT,%BDDAT ;CHECK IF RKASOF CORRECT
3774 026542 001407      BEQ      19$ ;YES, CHECK IF FINISHED
3775 026544 012737 063413 001330      MOV      #EM3,EM4N ;LOAD ERROR MESSAGE
3776 026552 012737 067011 001332      MOV      #EM1007,EM4N+2
3777 026560 104004      ERROR   4
3778 026562 104415      19$:    SCOPI ;CHECK IF LOOP ON ERROR
3779 026564 000261      SEC ;SHIFT IN ONE
3780 026566 006137 002010      ROL     CONFIG
3781 026572 005301      DEC     R1 ;CHECK IF FINISHED
3782 026574 001402      SEQ     TST34 ;;YES, GO ON TO NEXT TEST
3783 026576 000137 026020      JMP     1$

```

```

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

```

```

:ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
: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.

```

```

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

```

```

*****
:TST34: SCOPE

```

```

MOV      #500,%STIMES ;DO 500 ITERATIONS
MOV      #17,R1 ;LOAD NUMBER OF PATTERNS
CLR      CONFIG ;LOAD INITIAL CONFIGURATION
MOV      #EM17,EM3N ;LOAD ERROR MESSAGE
MOV      #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV      #1$,%LPERR ;LOAD LOOP ON ERROR LOCATION FOR
:SUBTEST LOOP

```

```

3801 026602 000004
3802 026604 012737 000764 001200
3803 026612 012701 000021
3804 026616 005037 002010
3805 026622 012737 064361 001320
3806 026630 012737 100000 000000
3807 026636 012737 026644 001110
3808
3809
3810 026644
3811 026644 005062 000002
3812 026650 013762 002010 000016
3813 026656 016237 000016 001126
3814 026664 013737 002010 001124
3815 026672 042737 177400 001124
3816 026700 023737 001124 001126
3817 026706 001404
3818 026710 012737 067011 001322
3819 026716 104003
3820 026720
3821 026720 016237 000000 001126
3822 026726 022737 000200 001126
3823 026734 001407
3824 026736 012737 000100 001124
3825 026744 012737 067324 001322

```

```

1$:    CLR     RKWC(R2) ;CLEAR WORD COUNT REG.
MOV     CONFIG,RKASOF(R2) ;WRITE RKASOF
MOV     RKASOF(R2),%BDDAT ;STORE RKASOF
MOV     CONFIG,%GDAT ;PREPARE EXPECTED RESULTS
BIC     #177400,%GDAT ;INITIALIZE READ ONLY BITS
CMP     %GDAT,%BDDAT ;CHECK IF RKASOF CORRECT
BEQ     2$ ;YES, TEST IF ANY OTHER REG MODIFIED
MOV     #EM1007,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     #IR,%GDAT ;LOAD EXPECTED RESULTS
MOV     #EM1017,EM3N+2 ;LOAD ERROR MESSAGE

```

3826	026752	104003				35:	ERROR	3	
3827	026754								
3828	026754	016237	000004	001126			MOV	RKBA(R2), \$BDDAT	; STORE BUS AND REG
3829	026762	001406					BEQ	15	; CHECK IF ZERO
3830	026764	005037	001124				CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
3831	026770	012737	067375	001322			MOV	#EM1019, EM3N+2	; LOAD ERROR MESSAGE
3832	026776	104003					ERROR	3	
3833	027000					45:			
3834	027000	016237	000002	001126			MOV	RKWC(R2), \$BDDAT	; STORE WORD COUNT REG
3835	027006	001406					BEQ	55	; CHECK IF ZERO
3836	027010	005037	001124				CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
3837	027014	012737	067345	001322			MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
3838	027022	104003					ERROR	3	
3839	027024					55:			
3840	027024	016237	000006	001126			MOV	RKDA(R2), \$BDDAT	; STORE DISK AVERAGE REG
3841	027032	001406					BEQ	65	; CHECK IF ZERO
3842	027034	005037	001124				CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
3843	027040	012737	067422	001322			MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
3844	027046	104003					ERROR	3	
3845	027050					65:			
3846	027050	016237	000010	001126			MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG.2
3847	027056	022737	000100	001126			CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
3848	027064	001407					BEQ	85	; YES, CONTINUE
3849	027066	012737	000100	001124			MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
3850	027074	012737	067460	001322			MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
3851	027102	104003					ERROR	3	
3852	027104					85:			
3853	027104	016237	000012	001126			MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
3854	027112	001406					BEQ	95	; CHECK IF ZERO
3855	027114	005037	001124				CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
3856	027120	012737	067501	001322			MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
3857	027126	104003					ERROR	3	
3858	027130	016237	000014	001126		95:			
3859	027136	001406					MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
3860	027140	005037	001124				BEQ	105	; CHECK IF ZERO
3861	027144	012737	067537	001322			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
3862	027152	104003					MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
3863	027154						ERROR	3	
3864	027154	016237	000020	001126		105:			
3865	027162	001406					MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
3866	027164	005037	001124				BEQ	125	; CHECK IF ZERO
3867	027170	012737	067634	001322			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
3868	027176	104003					MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
3869	027200						ERROR	3	
3870	027200	016237	000026	001126		125:			
3871	027206	012737	002000	001124			MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG. 1
3872	027214	032737	020000	001126			MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MR1
3873	027222	001403					BIT	#ECCW, \$BDDAT	
3874	027224	052737	020000	001124			BEQ	135	
3875	027232	023737	001124	001126		135:	BIS	#ECCW, \$GDDAT	
3876	027240	001404					CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT
3877	027242	012737	067661	001322			BEQ	145	; YES, ISSUE CONTROLLER CLEAR
3878	027250	104003					MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
3879	027252						ERROR	3	
3880	027252	016237	000032	001126		145:			
3881	027260	001406					MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
							BEQ	155	; CHECK IF ZERO

```

3982 027262 005037 001124          CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
3983 027266 012737 067737 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
3984 027274 104003          ERROR    3
3985 027276 016237 000030 001126 15$:      MOV      RKECP5(R2), $BDDAT ;STORE ECC POSITION REG.
3986 027304 012737 004066 001124 16$:      MOV      #4066,$GDDAT    ;USE 4066
3987 027312 023737 001124 001126 17$:      CMP      $GDDAT,$BDDAT   ;CHECK IF ECC POSITION CORRECT
3988 027320 001404          SEQ      18$            ;YES, INITIALIZE RK611
3989 027322 012737 067712 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3990 027330 104003          ERROR    3
3991 027332 016237 000016 002014 18$:      MOV      RKASOF(R2), PREREG ;GET PREVIOUS CONTENTS
3992 027340 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3993 027346 016237 000016 001126      MOV      RKASOF(R2), $BDDAT ;GET CURRENT VALUE
3994 027354 005037 001124          CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
3995 027360 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;CHECK IF RKASOF CORRECT
3996 027366 001407          BEQ     19$            ;YES, CHECK IF FINISHED
3997 027370 012737 063413 001330      MOV      #EM3,EM4N      ;LOAD ERROR MESSAGE
3998 027376 012737 067011 001332      MOV      #EM1007,EM4N+2
3999 027404 104004          ERROR    4
3900 027406 104415          19$:      SCOPE1                ;CHECK IF LOOP ON ERROR
3901 027410 000261          SEC                        ;SHIFT IN ONE
3902 027412 006037 002010      ROR     CONFIG
3903 027416 005301          DEC     R1              ;CHECK IF FINISHED
3904 027420 001402          BEQ     TST35           ;;YES, GO ON TO NEXT TEST
3905 027422 000137 026644          JMP     1$

```

```

*****
;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 CONFIGURATIONS
; SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
; REG. 2 AND CHECK FOR REGISTER INTERACTION.

```

```

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

```

```

*****
TST35: SCOPE

```

```

3923 027426 000004          MOV      #500,$TIMES    ;;DO 500. ITERATIONS
3924 027430 012737 000764 001200      MOV      #17,R1        ;LOAD NUMBER OF PATTERNS
3925 027436 012701 000021          MOV      #000001,CONFIG ;LOAD INITIAL CONFIGURATION
3926 027442 012737 000001 002010      MOV      #000001,CONF1
3927 027450 012737 000001 002012      MOV      #EM18,EM3N    ;LOAD ERROR MESSAGE
3928 027456 012737 064436 001320      MOV      #CCLR,RKCS1   ;CLEAR RK611 WITH CONTROLLER CLEAR
3929 027464 012737 100000 000000      MOV      #1$, $LPERR   ;LOAD LOOP ON ERROR LOCATION FOR
3930 027472 012737 027500 001110          MOV      ; SUBTEST LOOP
3931
3932
3933 027500          1$:      CLR      RKWC(R2)      ;CLEAR WORD COUNT REG.
3934 027500 005062 000002          MOV      CONFIG,RKCS2(R2) ;WRITE RKCS2
3935 027504 013762 002010 000010      MOV      RKCS2(R2), $BDDAT ;STORE RKCS2
3936 027512 016237 000010 001126      MOV      CONFIG,$GDDAT  ;PREPARE EXPECTED RESULTS
3937 027520 013737 002010 001124

```


F08

RA611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 74
 DZR6A8.P11 T35 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1)

SEQ 0075

3938	027526	042737	177600	001124	BIC	#DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,\$GDDAT
3939	027534	052737	000100	001124	BIS	#IR,\$GDDAT ;INITIALIZE READ ONLY BITS
3940	027542	023737	001124	001126	CMP	\$GDDAT,\$BDDAT ;CHECK IF RKCS2 CORRECT
3941	027550	001404			BEQ	2\$;YES,TEST IF ANY OTHER REG MODIFIED
3942	027552	012737	066714	001322	MOV	#EM1004,EM3N+2 ;LOAD ERROR MESSAGE
3943	027560	104003			ERROR	3
3944	027562					2\$:
3945	027562	016237	000000	001126	MOV	RKCS1(R2),\$BDDAT ;STORE COMMAND AND STATUS REG. 1
3946	027570	022737	000200	001126	CMP	#RDY,\$BDDAT ;CHECK IF CS1 CORRECT
3947	027576	001407			BEQ	3\$;YES,CONTINUE
3948	027600	012737	000100	001124	MOV	#IR,\$GDDAT ;LOAD EXPECTED RESULTS
3949	027606	012737	067324	001322	MOV	#EM1017,EM3N+2 ;LOAD ERROR MESSAGE
3950	027614	104003			ERROR	3
3951	027616					3\$:
3952	027616	016237	000004	001126	MOV	RKBA(R2),\$BDDAT ;STORE BUS AND REG
3953	027624	001406			BEQ	4\$;CHECK IF ZERO
3954	027626	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3955	027632	012737	067375	001322	MOV	#EM1019,EM3N+2 ;LOAD ERROR MESSAGE
3956	027640	104003			ERROR	3
3957	027642					4\$:
3958	027642	016237	000002	001126	MOV	RKWC(R2),\$BDDAT ;STORE WORD COUNT REG
3959	027650	001406			BEQ	5\$;CHECK IF ZERO
3960	027652	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3961	027656	012737	067345	001322	MOV	#EM1018,EM3N+2 ;LOAD ERROR MESSAGE
3962	027664	104003			ERROR	3
3963	027666					5\$:
3964	027666	016237	000006	001126	MOV	RKDA(R2),\$BDDAT ;STORE DISK AVERAGE REG
3965	027674	001406			BEQ	6\$;CHECK IF ZERO
3966	027676	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3967	027702	012737	067422	001322	MOV	#EM1020,EM3N+2 ;LOAD ERROR MESSAGE
3968	027710	104003			ERROR	3
3969	027712					6\$:
3970	027712	016237	000016	001126	MOV	RKASOF(R2),\$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
3971	027720	001406			BEQ	7\$;CHECK IF ZERO
3972	027722	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3973	027726	012737	067566	001322	MOV	#EM1024,EM3N+2 ;LOAD ERROR MESSAGE
3974	027734	104003			ERROR	3
3975	027736					7\$:
3976	027736	016237	000012	001126	MOV	RKDS(R2),\$BDDAT ;STORE DRIVE STATUS REG
3977	027744	001406			BEQ	8\$;CHECK IF ZERO
3978	027746	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3979	027752	012737	067501	001322	MOV	#EM1022,EM3N+2 ;LOAD ERROR MESSAGE
3980	027760	104003			ERROR	3
3981	027762	016237	000014	001126	MOV	RKER(R2),\$BDDAT ;STORE ERROR REG
3982	027770	001406			BEQ	9\$;CHECK IF ZERO
3983	027772	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3984	027776	012737	067537	001322	MOV	#EM1023,EM3N+2 ;LOAD ERROR MESSAGE
3985	030004	104003			ERROR	3
3986	030006					10\$:
3987	030006	016237	000020	001126	MOV	RKDCYL(R2),\$BDDAT ;STORE CYLINDER ADD REG
3988	030014	001406			BEQ	12\$;CHECK IF ZERO
3989	030016	005037	001124		CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
3990	030022	012737	067634	001322	MOV	#EM1025,EM3N+2 ;LOAD ERROR MESSAGE
3991	030030	104003			ERROR	3
3992	030032					12\$:
3993	030032	016237	000026	001126	MOV	RKMR1(R2),\$BDDAT ;STORE MAINTENANCE REG.1

```

3994 030040 012737 002000 001124      MOV      #MEWD,$GDDAT      ;LOAD EXPECTED MR1
3995 030046 032737 020000 001126      BIT      #ECCW,$BDDAT
3996 030054 001403                      BEQ      13$
3997 030056 052737 020000 001124      BIS      #ECCW,$GDDAT
3998 030064 023737 001124 001126 13$:    CMP      $GDDAT,$BDDAT    ;CHECK IF MR1 CORRECT
3999 030072 001404                      BEQ      14$              ;YES,ISSUE CONTROLLER CLEAR
4000 030074 012737 067661 001322      MOV      #EM1026,EM3N+2  ;LOAD ERROR MESSAGE
4001 030102 104003                      ERROR    3
4002 030104                      14$:
4003 030104 016237 000032 001126      MOV      RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
4004 030112 001406                      BEQ      15$              ;CHECK IF ZERO
4005 030114 005037 001124                      CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
4006 030120 012737 067737 001322      MOV      #EM1030,EM3N+2  ;LOAD ERROR MESSAGE
4007 030126 104003                      ERROR    3
4008 030130 016237 000030 001126 15$:    MOV      RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
4009 030136 012737 004066 001124 16$:    MOV      #4066,$GDDAT    ;USE 4066
4010 030144 023737 001124 001126 17$:    CMP      $GDDAT,$BDDAT    ;CHECK IF ECC POSITION CORRECT
4011 030152 001404                      BEQ      18$              ;YES,INITIALIZE RK611
4012 030154 012737 067712 001322      MOV      #EM1029,EM3N+2  ;LOAD ERROR MESSAGE
4013 030162 104003                      ERROR    3
4014 030164 016237 000010 002014 18$:    MOV      RKCS2(R2),PREREG  ;GET PREVIOUS CONTENTS
4015 030172 012762 100000 000000      MOV      #CLR,RKCS1(R2)  ;CLEAR RK611 CONTROLLER
4016 030200 016237 000010 001126      MOV      RKCS2(R2),$BDDAT ;GET CURRENT VALUE
4017 030206 012737 000100 001124      MOV      #IR,$GDDAT      ;LOAD EXPECTED CONTENTS
4018 030214 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;CHECK IF RKCS2 CORRECT
4019 030222 001407                      BEQ      19$              ;YES,CHECK IF FINISHED
4020 030224 012737 063413 001330      MOV      #EM3,EM4N      ;LOAD ERROR MESSAGE
4021 030232 012737 066714 001332      MOV      #EM1004,EM4N+2
4022 030240 104004                      ERROR    4
4023 030242 104415                      19$:    SCOPI
4024 030244 000241                      CLC
4025 030246 006137 002012                      ROL      CONFIG1
4026 030252 013737 002012 002010      MOV      CONFIG1,CONFIG  ;MAKE SURE SUBSYSTEM CLEAR
4027 030260 042737 000040 002010      BIC      #SCLR,CONFIG    ;DOES NOT SET
4028 030266 005301                      DEC      R1
4029 030270 001402                      BEQ      TST36
4030 030272 000137 027500                      JMP      1$
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049

```

```

*****
*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 CONFIGURATIONS
* SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
* REG. 2 AND CHECK FOR REGISTER INTERACTION.
*
* 177737 177727 177337 167737
* 177736 177717 176737 157737
* 177735 177637 175737 137737
* 177733 177537 173737 077737
*****
TST36: SCOPE
MOV #500.,$TIMES ;;DO 500. ITERATIONS

```

H08

4050	030306	012701	000021		MOV	#17,R1	;LOAD NUMBER OF PATTERNS
4051	030312	012737	177736	002010	MOV	#177736,CONFIG	;LOAD INITIAL CONFIGURATION
4052	030320	012737	177776	002012	MOV	#177776,CONFIG1	
4053	030326	012737	064436	001320	MOV	#EM18,EM3N	;LOAD ERROR MESSAGE
4054	030334	012737	100000	000000	MOV	#CLR,RKCS1	;CLEAR RK611 WITH CONTROLLER CLEAR
4055	030342	012737	030350	001110	MOV	#15,\$LPERR	;LOAD LOOP ON ERROR LOCATION FOR ; SUBTEST LOOP
4056							
4057							
4058	030350						
4059	030350	005062	000002		15:	CLR	RKWC(R2) ;CLEAR WORD COUNT REG.
4060	030354	013762	002010	000010	MOV	CONFIG,RKCS2(R2)	;WRITE RKCS2
4061	030362	016237	000010	001126	MOV	RKCS2(R2),SBDDAT	;STORE RKCS2
4062	030370	013737	002010	001124	MOV	CONFIG,\$GDDAT	;PREPARE EXPECTED RESULTS
4063	030376	042737	177600	001124	BIC	#DLT!WCE!UPE!NED	;NEM!PGE!MDS!UFE!OR,\$GDDAT
4064	030404	052737	000100	001124	BIS	#IR,\$GDDAT	;INITIALIZE READ ONLY BITS
4065	030412	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;CHECK IF RKCS2 CORRECT
4066	030420	001404			BEQ	25	;YES TEST IF ANY OTHER REG MODIFIED
4067	030422	012737	066714	001322	MOV	#EM1004,EM3N+2	;LOAD ERROR MESSAGE
4068	030430	104003			ERROR	3	
4069	030432						
4070	030432	016237	000000	001126	25:	MOV	RKCS1(R2),SBDDAT ;STORE COMMAND AND STATUS REG. 1
4071	030440	022737	000200	001126	CMP	#RDY,\$BDDAT	;CHECK IF CS1 CORRECT
4072	030446	001407			BEQ	35	;YES, CONTINUE
4073	030450	012737	000100	001124	MOV	#IR,\$GDDAT	;LOAD EXPECTED RESULTS
4074	030456	012737	067324	001322	MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
4075	030464	104003			ERROR	3	
4076	030466						
4077	030466	016237	000004	001126	35:	MOV	RKBA(R2),SBDDAT ;STORE BUS AND REG
4078	030474	001406			BEQ	45	;CHECK IF ZERO
4079	030476	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4080	030502	012737	067375	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
4081	030510	104003			ERROR	3	
4082	030512						
4083	030512	016237	000002	001126	45:	MOV	RKWC(R2),SBDDAT ;STORE WORD COUNT REG
4084	030520	001406			BEQ	55	;CHECK IF ZERO
4085	030522	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4086	030526	012737	067345	001322	MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
4087	030534	104003			ERROR	3	
4088	030536						
4089	030536	016237	000006	001126	55:	MOV	RKDA(R2),SBDDAT ;STORE DISK AVERAGE REG
4090	030544	001406			BEQ	65	;CHECK IF ZERO
4091	030546	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4092	030552	012737	067422	001322	MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
4093	030560	104003			ERROR	3	
4094	030562						
4095	030562	016237	000016	001126	65:	MOV	RKASOF(R2),SBDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
4096	030570	001406			BEQ	75	;CHECK IF ZERO
4097	030572	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4098	030576	012737	067566	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
4099	030604	104003			ERROR	3	
4100	030606						
4101	030606	016237	000012	001126	75:	MOV	RKDS(R2),SBDDAT ;STORE DRIVE STATUS REG
4102	030614	001406			BEQ	95	;CHECK IF ZERO
4103	030616	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4104	030622	012737	067501	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
4105	030630	104003			ERROR	3	

```

4106 030632 016237 000014 001126 95:  MOV  RKER(R2),SBDDAT ;STORE ERROR REG
4107 030640 001406          BEQ  105             ;CHECK IF ZERO
4108 030642 005037 001124          CLR  $GDDAT         ;LOAD EXPECTED CONTENTS
4109 030646 012737 067537 001322  MOV  #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
4110 030654 104003          ERROR 3
4111 030656          105:
4112 030656 016237 000020 001126  MOV  RKDCYL(R2),SBDDAT ;STORE CYLINDER ADD REG
4113 030664 001406          BEQ  125             ;CHECK IF ZERO
4114 030666 005037 001124          CLR  $GDDAT         ;LOAD EXPECTED CONTENTS
4115 030672 012737 067634 001322  MOV  #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
4116 030700 104003          ERROR 3
4117 030702          125:
4118 030702 016237 000026 001126  MOV  RKMRI(R2),SBDDAT ;STORE MAINTENANCE REG.1
4119 030710 012737 002000 001124  MOV  #MEWD,$GDDAT    ;LOAD EXPECTED MRI
4120 030716 032737 020000 001126  BIT  #ECCW,SBDDAT
4121 030724 001403          BEQ  135
4122 030726 052737 020000 001124  BIS  #ECCW,$GDDAT
4123 030734 023737 001124 001126  135:  CMP  $GDDAT,SBDDAT  ;CHECK IF MRI CORRECT
4124 030742 001404          BEQ  145             ;YES,ISSUE CONTROLLER CLEAR
4125 030744 012737 067661 001322  MOV  #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
4126 030752 104003          ERROR 3
4127 030754          145:
4128 030754 016237 000032 001126  MOV  RKECPT(R2),SBDDAT ;STORE ECC PATTERN REG.
4129 030762 001406          BEQ  155             ;CHECK IF ZERO
4130 030764 005037 001124          CLR  $GDDAT         ;LOAD EXPECTED CONTENTS
4131 030770 012737 067737 001322  MOV  #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
4132 030776 104003          ERROR 3
4133 031000 016237 000030 001126  155:  MOV  RKECPS(R2),SBDDAT ;STORE ECC POSITION REG.
4134 031006 012737 004066 001124  165:  MOV  #4066,$GDDAT   ;USE 4066
4135 031014 023737 001124 001126  175:  CMP  $GDDAT,SBDDAT  ;CHECK IF ECC POSITION CORRECT
4136 031022 001404          BEQ  185             ;YES,INITIALIZE RK611
4137 031024 012737 067712 001322  MOV  #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
4138 031032 104003          ERROR 3
4139 031034 016237 000010 002014  185:  MOV  RKCS2(R2),PREREG ;GET PREVIOUS CONTENTS
4140 031042 012762 100000 000000  MOV  #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4141 031050 016237 000010 001126  MOV  RKCS2(R2),SBDDAT ;GET CURRENT VALUE
4142 031056 012737 000100 001124  MOV  #IR,$GDDAT     ;LOAD EXPECTED CONTENTS
4143 031064 023737 001124 001126  CMP  $GDDAT,SBDDAT  ;CHECK IF RKCS2 CORRECT
4144 031072 001407          BEQ  195             ;YES,CHECK IF FINISHED
4145 031074 012737 063413 001330  MOV  #EM3,EM4N      ;LOAD ERROR MESSAGE
4146 031102 012737 066714 001332  MOV  #EM1004,EM4N+2
4147 031110 104004          ERROR 4
4148 031112 104415          195:  SCOP1 ;CHECK IF LOOP ON ERROR
4149 031114 000261          SEC  ;SHIFT IN ONE
4150 031116 006137 002012          ROL  CONFIG1
4151 031122 013737 002012 002010  MOV  CONFIG1,CONFIG ;MAKE SURE SUBSYSTEM CLEAR
4152 031130 042737 000040 002010  BIC  #SCLR,CONFIG   ;DOES NOT SET
4153 031136 005301          DEC  R1             ;CHECK IF FINISHED
4154 031140 001402          BEQ  TST37         ;YES, GO ON TO NEXT TEST
4155 031142 000137 030350          JMP  15

```

```

4156
4157 ;*****
4158 ;*TEST 37 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 3)
4159 ;*
4160 ;* RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
4161 ;* WRITE THE WORD COUNT REGISTER WITH ZERO.

```

JOB

```

4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173 031146 000004
4174 031150 012737 000764 001200
4175 031156 012701 000021
4176 031162 005037 002010
4177 031166 005037 002012
4178 031172 012737 064436 001320
4179 031200 012737 100000 000000
4180 031206 012737 031214 001110
4181
4182
4183 031214
4184 031214 005062 000002
4185 031220 013762 002010 000010
4186 031226 016237 000010 001126
4187 031234 013737 002010 001124
4188 031242 042737 177600 001124
4189 031250 052737 000100 001124
4190 031256 023737 001124 001126
4191 031264 001404
4192 031266 012737 066714 001322
4193 031274 104003
4194 031276
4195 031276 016237 000000 001126
4196 031304 022737 000200 001126
4197 031312 001407
4198 031314 012737 000100 001124
4199 031322 012737 067324 001322
4200 031330 104003
4201 031332
4202 031332 016237 000004 001126
4203 031340 001406
4204 031342 005037 001124
4205 031346 012737 067375 001322
4206 031354 104003
4207 031356
4208 031356 016237 000002 001126
4209 031364 001406
4210 031366 005037 001124
4211 031372 012737 067345 001322
4212 031400 104003
4213 031402
4214 031402 016237 000006 001126
4215 031410 001406
4216 031412 005037 001124
4217 031416 012737 067422 001322

```

```

*****
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 07 37
000007 000337 007737 17 37
000017 000737 017737 000000
*****
TST37: 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 ; CLEAR RK611 WITH CONTROLLER CLEAR
MOV #15, $LPERR ; LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

15:
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 25 ; YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1004, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

25:
MOV RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
CMP #RDY, $BDDAT ; CHECK IF CS1 CORRECT
BEQ 35 ; YES, CONTINUE
MOV #IR, $GDDAT ; LOAD EXPECTED RESULTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

35:
MOV RKBA(R2), $BDDAT ; STORE BUS AND REG
BEQ 45 ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

45:
MOV RKWC(R2), $BDDAT ; STORE WORD COUNT REG
BEQ 55 ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

55:
MOV RKDA(R2), $BDDAT ; STORE DISK AVERAGE REG
BEQ 65 ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE

```

K08

4218	031424	104003				ERROR	3	
4219	031426				6\$:			
4220	031426	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG.
4221	031434	001406				BEQ	7\$; CHECK IF ZERO
4222	031436	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4223	031442	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
4224	031450	104003				ERROR	3	
4225	031452				7\$:			
4226	031452	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
4227	031460	001406				BEQ	9\$; CHECK IF ZERO
4228	031462	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4229	031466	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
4230	031474	104003				ERROR	3	
4231	031476	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
4232	031504	001406				BEQ	10\$; CHECK IF ZERO
4233	031506	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4234	031512	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
4235	031520	104003				ERROR	3	
4236	031522				10\$:			
4237	031522	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
4238	031530	001406				BEQ	12\$; CHECK IF ZERO
4239	031532	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4240	031536	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
4241	031544	104003				ERROR	3	
4242	031546				12\$:			
4243	031546	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG. 1
4244	031554	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MRI
4245	031562	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4246	031570	001403				BEQ	13\$	
4247	031572	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4248	031600	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MRI CORRECT
4249	031606	001404				BEQ	14\$; YES, ISSUE CONTROLLER CLEAR
4250	031610	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
4251	031616	104003				ERROR	3	
4252	031620				14\$:			
4253	031620	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
4254	031626	001406				BEQ	15\$; CHECK IF ZERO
4255	031630	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4256	031634	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
4257	031642	104003				ERROR	3	
4258	031644	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG.
4259	031652	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	; USE 4066
4260	031660	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF ECC POSITION CORRECT
4261	031666	001404				BEQ	18\$; YES, INITIALIZE RK611
4262	031670	012737	067712	001322		MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE
4263	031676	104003				ERROR	3	
4264	031700	016237	000010	002014	18\$:	MOV	RKCS2(R2), PREREG	; GET PREVIOUS CONTENTS
4265	031706	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	; CLEAR RK611 CONTROLLER
4266	031714	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	; GET CURRENT VALUE
4267	031722	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
4268	031730	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; CHECK IF RKCS2 CORRECT
4269	031736	001407				BEQ	19\$; YES, CHECK IF FINISHED
4270	031740	012737	063413	001330		MOV	#EM3, EM4N	; LOAD ERROR MESSAGE
4271	031746	012737	066714	001332		MOV	#EM1004, EM4N+2	
4272	031754	104004				ERROR	4	
4273	031756	104415			19\$:	SCOP1		; CHECK IF LOOP ON ERROR

4274	031760	000261			SEC		;SHIFT IN ONE
4275	031762	006137	002012		ROL	CONFIG1	
4276	031766	013737	002012	002010	MOV	CONFIG1,CONFIG	;MAKE SURE SUBSYSTEM CLEAR
4277	031774	042737	000040	002010	BIC	#SCLR,CONFIG	; DOES NOT SET
4278	032002	005301			DEC	R1	;CHECK IF FINISHED
4279	032004	001402			BEQ	TST40	;YES, GO ON TO NEXT TEST
4280	032006	000137	031214		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

4298	032012	000004			MOV	#500,\$TIMES	;DO 500. ITERATIONS
4299	032014	012737	000764	001200	MOV	#17,R1	;LOAD NUMBER OF PATTERNS
4300	032022	012701	000021		CLR	CONFIG	;LOAD INITIAL OCNFIGURATION
4301	032026	005037	002010		CLR	CONFIG1	
4302	032032	005037	002012		MOV	#EM18,EM3N	;LOAD ERROR MESSAGE
4303	032036	012737	064436	001320	MOV	#CCLR,RKCS1	;CLEAR RK611 WITH CONTROLLER CLEAR
4304	032044	012737	100000	000000	MOV	#1\$, \$LPERR	;LOAD LOOP ON ERROR LOCATION FOR
4305	032052	012737	032060	001110			; SUBTEST LOOP

4308	032060				1\$: CLR	RKWC(R2)	;CLEAR WORD COUNT REG.
4309	032060	005062	000002		MOV	CONFIG,RKCS2(R2)	;WRITE RKCS2
4310	032064	013762	002010	000010	MOV	RKCS2(R2), \$BDDAT	;STORE RKCS2
4311	032072	016237	000010	001126	MOV	CONFIG,\$GDDAT	;PREPARE EXPECTED RESULTS
4312	032100	013737	002010	001124	BIC	#DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,\$GDDAT	
4313	032106	042737	177600	001124	BIS	#IR,\$GDDAT	;INITIALIZE READ ONLY BITS
4314	032114	052737	000100	001124	CMP	\$GDDAT,\$BDDAT	;CHECK IF RKCS2 CORRECT
4315	032122	023737	001124	001126	BEQ	2\$;YES,TEST IF ANY OTHER REG MODIFIED
4316	032130	001404			MOV	#EM1004,EM3N+2	;LOAD ERROR MESSAGE
4317	032132	012737	066714	001322	ERROR	3	

4319	032142				2\$: MOV	RKCS1(R2), \$BDDAT	;STORE COMMAND AND STATUS REG. 1
4320	032142	016237	000000	001126	CMP	#RDY,\$BDDAT	;CHECK IF CS1 CORRECT
4321	032150	022737	000200	001126	BEQ	3\$;YES, CONTINUE
4322	032156	001407			MOV	#IR,\$GDDAT	;LOAD EXPECTED RESULTS
4323	032160	012737	000100	001124	MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
4324	032166	012737	067324	001322	ERROR	3	

4326	032176				3\$: MOV	RKBA(R2), \$BDDAT	;STORE BUS AND REG
4327	032176	016237	000004	001126	BEQ	4\$;CHECK IF ZERO
4328	032204	001406			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4329	032206	005037	001124				

MO8

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 81
 DZR6A8.P11 T40 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 4)

SEQ 0082

4330	032212	012737	067375	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
4331	032220	104003				ERROR	3	
4332	032222				4\$:			
4333	032222	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	;STORE WORD COUNT REG
4334	032230	001406				BEQ	5\$;CHECK IF ZERO
4335	032232	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4336	032236	012737	067345	001322		MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
4337	032244	104003				ERROR	3	
4338	032246				5\$:			
4339	032246	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	;STORE DISK AVERAGE REG
4340	032254	001406				BEQ	6\$;CHECK IF ZERO
4341	032256	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4342	032262	012737	067422	001322		MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
4343	032270	104003				ERROR	3	
4344	032272				6\$:			
4345	032272	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
4346	032300	001406				BEQ	7\$;CHECK IF ZERO
4347	032302	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4348	032306	012737	067566	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
4349	032314	104003				ERROR	3	
4350	032316				7\$:			
4351	032316	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
4352	032324	001406				BEQ	9\$;CHECK IF ZERO
4353	032326	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4354	032332	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
4355	032340	104003				ERROR	3	
4356	032342	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
4357	032350	001406				BEQ	10\$;CHECK IF ZERO
4358	032352	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4359	032356	012737	067537	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
4360	032364	104003				ERROR	3	
4361	032366				10\$:			
4362	032366	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
4363	032374	001406				BEQ	12\$;CHECK IF ZERO
4364	032376	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4365	032402	012737	067634	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
4366	032410	104003				ERROR	3	
4367	032412				12\$:			
4368	032412	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
4369	032420	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MR1
4370	032426	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4371	032434	001403				BEQ	13\$	
4372	032436	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4373	032444	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
4374	032452	001404				BEQ	14\$;YES, ISSUE CONTROLLER CLEAR
4375	032454	012737	067661	001322		MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
4376	032462	104003				ERROR	3	
4377	032464				14\$:			
4378	032464	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
4379	032472	001406				BEQ	15\$;CHECK IF ZERO
4380	032474	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4381	032500	012737	067737	001322		MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
4382	032506	104003				ERROR	3	
4383	032510	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
4384	032516	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	;USE 4066
4385	032524	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT

N08

```
4386 032532 001404          BEQ      18$          ;YES, INITIALIZE RK611
4387 032534 012737 067712 001322  MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
4388 032542 104003          ERROR    3
4389 032544 016237 000010 002014 18$:  MOV      RKCS2(R2),PREREG ;GET PREVIOUS CONTENTS
4390 032552 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4391 032560 016237 000010 001126  MOV      RKCS2(R2),SBDDAT ;GET CURRENT VALUE
4392 032566 012737 000100 001124  MOV      #IR,$GDDAT ;LOAD EXPECTED CONTENTS
4393 032574 023737 001124 001126  CMP      $GDDAT,$BDDAT ;CHECK IF RKCS2 CORRECT
4394 032602 001407          BEQ      19$          ;YES, CHECK IF FINISHED
4395 032604 012737 063413 001330  MOV      #EM3,EM4N ;LOAD ERROR MESSAGE
4396 032612 012737 066714 001332  MOV      #EM1004,EM4N+2
4397 032620 104004          ERROR    4
4398 032622 104415          19$:  SCOPE1 ;CHECK IF LOOP ON ERROR
4399 032624 000261          SEC ;SHIFT IN ONE
4400 032626 006037 002012  ROR      CONFIG1
4401 032632 013737 002012 002010  MOV      CONFIG1,CONFIG ;MAKE SURE SUBSYSTEM CLEAR
4402 032640 042737 000040 002010  BIC      #SCLR,CONFIG ;DOES NOT SET
4403 032646 005301          DEC      R1 ;CHECK IF FINISHED
4404 032650 001402          BEQ      TST41 ;YES, GO ON TO NEXT TEST
4405 032652 000137 032060  JMP      1$
4406
4407 ;*****
4408 ;*TEST 41 CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS
4409 ;*
4410 ;* THIS TEST WILL TEST THE ABILITY OF THE SUBSYSTEM CLEAR TO
4411 ;* INITIALIZE THE BUS ADDRESS REGISTER AND COMMAND
4412 ;* AND STATUS REGISTER 1. IT WILL ALSO VERIFY THAT ALL
4413 ;* OTHER REGISTERS REMAIN IN THE INITIALIZED STATE.
4414 ;*
4415 ;*****
4416 032656 000004          TST41: SCOPE
4417 032660 012737 000764 001200  MOV      #500,$TIMES ;DO 500. ITERATIONS
4418 032666 013702 001270  MOV      $BASE,R2 ;LOAD RK611 BASE
4419 032672 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4420 032700 005062 000002  CLR      RKWC(R2) ;CLEAR WORD COUNT REG
4421 032704 012762 177776 000004  MOV      #177776,RKBA(R2) ;WRITE WORD COUNT REG.
4422 032712 012762 013776 000000  MOV      #013776,RKCS1(R2) ;WRITE COMMAND AND STATUS REG 1
4423 032720 012762 000040 000010  MOV      #SCLR,RKCS2(R2) ;ISSUE A SUBSYSTEM CLEAR
4424 032726 016237 000004 001126  MOV      RKBA(R2),SBDDAT ;STORE BUS ADDRESS REG.
4425 032734 001414          BEQ      1$          ;CHECK IF ZERO
4426 032736 012737 177776 002014  MOV      #177776,PREREG ;LOAD PREVIOUS CONTENTS
4427 032744 005037 001124  CLR      $GDDAT ;LOAD EXPECTED CONTENTS
4428 032750 012737 064567 001330  MOV      #EM21,EM4N ;LOAD ERROR MESSAGE
4429 032756 012737 066647 001330  MOV      #EM1002,EM4N
4430 032764 104004          ERROR    4
4431 032766 016237 000000 001126 1$:  MOV      RKCS1(R2),SBDDAT ;STORE COMMAND AND STATUS REG.1
4432 032774 022737 000200 001126  CMP      #RDY,$BDDAT ;CHECK IF CS1 CORRECT
4433 033002 001415          BEQ      2$          ;YES, CHECK IF ALL OTHER REGISTER
4434 ;* IN INITIALIZE STATE
4435 033004 012737 013776 002014  MOV      #013776,PREREG ;LOAD PREVIOUS CONTENTS
4436 033012 012737 000200 001124  MOV      #RDY,$GDDAT ;LOAD EXPECTED CONTENTS
4437 033020 012737 064644 001330  MOV      #EM22,EM4N ;LOAD ERROR MESSAGE
4438 033026 012737 066647 001332  MOV      #EM1002,EM4N+2
4439 033034 104004          ERROR    4
4440 033036 012737 064715 001310 2$:  MOV      #EM23,EM2N ;LOAD ERROR MESSAGE
4441 033044 016237 000002 001126  MOV      RKWC(R2),SBDDAT ;STORE WORD COUNT REG.
```

```

4442 033052 001406 BEQ 35 ;CHECK IF ZERO
4443 033054 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4444 033060 012737 066622 001312 MOV #EM1001,EM2N+2 ;LOAD ERROR MESSAGE
4445 033066 104002 ERROR 2
4446 033070 016237 000006 001126 35: MOV RKDA(R2),SBDDAT ;STORE DISK ADDRESS REG.
4447 033076 001406 BEQ 45 ;CHECK IF ZERO
4448 033100 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4449 033104 012737 066671 001312 MOV #EM1003,EM2N+2 ;LOAD ERROR MESSAGE
4450 033112 104002 ERROR 2
4451 033114 016237 000016 001126 45: MOV RKASOF(R2),SBDDAT ;STORE ATTENTION SUMMARY AND OFFSET REG.
4452 033122 001406 BEQ 55 ;CHECK IF ZERO
4453 033124 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4454 033130 012737 067011 001312 MOV #EM1007,EM2N+2 ;LOAD ERROR MESSAGE
4455 033136 104002 ERROR 2
4456 033140 016237 000010 001126 55: MOV RKCS2(R2),SBDDAT ;STORE COMMAND AND STATUS REG.2
4457 033146 022737 000100 001126 CMP #IR,SBDDAT ;CHECK IF CS2 CORRECT
4458 033154 001407 BEQ 65 ;YES, CONTINUE
4459 033156 012737 000100 001124 MOV #IR,$GDDAT ;LOAD EXPECTED CONTENTS
4460 033164 012737 066714 001312 MOV #EM1004,EM2N+2 ;LOAD ERROR MESSAGE
4461 033172 104002 ERROR 2
4462 033174 016237 000012 001126 65: MOV RKDS(R2),SBDDAT ;STORE DRIVE STATUS REG.
4463 033202 001406 BEQ 75 ;CHECK IF ZERO
4464 033204 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4465 033210 012737 066732 001312 MOV #EM1005,EM2N+2 ;LOAD ERROR MESSAGE
4466 033216 104002 ERROR 2
4467 033220 016237 000014 001126 75: MOV RKER(R2),SBDDAT ;STORE ERROR REGISTER
4468 033226 001406 BEQ 95 ;CHECK IF ZERO
4469 033230 005037 001126 CLR $BDDAT ;LOAD EXPECTED CONTENTS
4470 033234 012737 066765 001312 MOV #EM1006,EM2N+2 ;LOAD ERROR MESSAGE
4471 033242 104002 ERROR 2
4472 033244 016237 000020 001126 85: MOV RKDCYL(R2),SBDDAT ;STORE CYLINDER ADD REG.
4473 033252 001406 BEQ 105 ;CHECK IF ZERO
4474 033254 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4475 033260 012737 067302 001312 MOV #EM1016,EM2N+2 ;LOAD ERROR MESSAGE
4476 033266 104002 ERROR 2
4477 033270 016237 000026 001126 105: MOV RKMR1(R2),SBDDAT ;STORE CYLINDER ADD REG.
4478 033276 012737 002000 001124 MOV #MEWD,$GDDAT ;LOAD EXPECTED MR1
4479 033304 032737 020000 001126 BIT #ECCW,SBDDAT
4480 033312 001403 BEQ 115
4481 033314 052737 020000 001124 BIS #ECCW,$GDDAT
4482 033322 023737 001124 001126 115: CMP $GDDAT,SBDDAT ;CHECK IF MR1 CORRECT
4483 033330 001404 BEQ 125 ;YES, CONTINUE TEST
4484 033332 012737 067102 001312 MOV #EM1009,EM2N+2 ;LOAD ERROR MESSAGE
4485 033340 104002 ERROR 2
4486 033342 016237 000032 001126 125: MOV RKECPT(R2),SBDDAT ;STORE ECC PATTERN REG.
4487 033350 001406 BEQ 135 ;CHECK IF ZERO
4488 033352 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
4489 033356 012737 067152 001312 MOV #EM1013,EM2N+2 ;LOAD ERROR MESSAGE
4490 033364 104002 ERROR 2
4491 033366 016237 000030 001126 135: MOV RKECPS(R2),SBDDAT ;STORE ECC POSITION REG.
4492 033374 022737 004066 001126 CMP #4066,SBDDAT ;CHECK IF ECC POSITION CORRECT
4493 033402 001407 BEQ TST42 ;YES, GO TO NEXT TEST
4494 033404 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED RESULTS
4495 033412 012737 067130 001312 MOV #EM1012,EM2N+2 ;LOAD ERROR MESSAGE
4496 033420 104002 ERROR 2
4497 ;*****

```

```

4498                                     : *TEST 42      REGISTER INTERACTION USING DRIVE STATUS
4499                                     : *
4500                                     : *          ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
4501                                     : *          WRITE WORD COUNT TO 0, WRITE DRIVE STATUS REGISTER
4502                                     : *          WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
4503                                     : *          INTERACTION TAKES PLACE.
4504                                     : *
4505                                     : *****
4506 033422 000004 15: ST42: SCOPE
4507 033424 012737 000764 001200 MOV      #500, $TIMES      ;; DO 500. ITERATIONS
4508 033432 013702 001270      MOV      $BASE, R2      ;; LOAD RK611 BASE ADDRESS
4509 033436 012737 177777 002010 MOV      #177777, CONFIG ;; LOAD CONFIGURATION WORD
4510 033444 012737 064025 001320 MOV      #EM10, EM3N      ;; LOAD ERROR MESSAGE
4511 033452 012762 100000 000000 MOV      #CCLR, RKCS1(R2) ;; CLEAR RK611 WITH CONTROLLER CLEAR
4512 033460 005062 000002      CLR      RKWC(R2)      ;; CLEAR WORD COUNT REG.
4513 033464 012762 177777 000012 MOV      #177777, RKDS(R2) ;; WRITE RKDS WITH 177777
4514 033472 016237 000000 001126 MOV      RKCS1(R2), $BDDAT ;; STORE COMMAND AND STATUS REG. 1
4515 033500 022737 000200 001126 CMP      #RDY, $BDDAT      ;; CHECK IF CS1 CORRECT
4516 033506 001407      BEQ      1$              ;; YES, CONTINUE
4517 033510 012737 000200 001124 MOV      #RDY, $GDDAT      ;; LOAD EXPECTED RESULTS
4518 033516 012737 067324 001322 MOV      #EM1017, EM3N+2 ;; LOAD ERROR MESSAGE
4519 033524 104003      ERROR      3
4520 033526 016237 000004 001126 15: MOV      RKBA(R2), $BDDAT ;; STORE BUS ADD REG.
4521 033534 001406      BEQ      2$              ;; CHECK IF ZERO
4522 033536 005037 001124      CLR      $GDDAT      ;; LOAD EXPECTED CONTENTS
4523 033542 012737 067375 001322 MOV      #EM1019, EM3N+2 ;; LOAD ERROR MESSAGE
4524 033550 104003      ERROR      3
4525 033552 016237 000002 001126 25: MOV      RKWC(R2), $BDDAT ;; STORE WORK COUNT REG.
4526 033560 001406      BEQ      3$              ;; CHECK IF ZERO
4527 033562 005037 001124      CLR      $GDDAT      ;; LOAD EXPECTED CONTENTS
4528 033566 012737 067345 001322 MOV      #EM1018, EM3N+2 ;; LOAD ERROR MESSAGE
4529 033574 104003      ERROR      3
4530 033576 016237 000006 001126 35: MOV      RKDA(R2), $BDDAT ;; STORE DISK ADD REG
4531 033604 001406      BEQ      4$              ;; CHECK IF ZERO
4532 033606 005037 001124      CLR      $GDDAT      ;; LOAD EXPECTED CONTENTS
4533 033612 012737 067422 001322 MOV      #EM1020, EM3N+2 ;; LOAD ERROR MESSAGE
4534 033620 104003      ERROR      3
4535 033622 016237 000016 001126 45: MOV      RKASOF(R2), $BDDAT ;; STORE ATTENTION SUMMARY/OFFSET REG
4536 033630 001406      BEQ      5$              ;; CHECK IF ZERO
4537 033632 005037 001124      CLR      $GDDAT      ;; LOAD EXPECTED CONTENTS
4538 033636 012737 067566 001322 MOV      #EM1024, EM3N+2 ;; LOAD ERROR MESSAGE
4539 033644 104003      ERROR      3
4540 033646 016237 000010 001126 55: MOV      RKCS2(R2), $BDDAT ;; STORE COMMAND AND STATUS REG. 2
4541 033654 022737 000100 001126 CMP      #IR, $BDDAT      ;; CHECK IF CS2 CORRECT
4542 033662 001407      BEQ      6$              ;; YES, CONTINUE
4543 033664 012737 000100 001124 MOV      #IR, $GDDAT      ;; LOAD EXPECTED CONTENTS
4544 033672 012737 067460 001322 MOV      #EM1021, EM3N+2 ;; LOAD ERROR MESSAGE
4545 033700 104003      ERROR      3
4546 033702 016237 000012 001126 65: MOV      RKDS(R2), $BDDAT ;; STORE DRIVE STATUS REG.
4547 033710 001406      BEQ      7$              ;; CHECK IF ZERO
4548 033712 005037 001124      CLR      $GDDAT      ;; LOAD EXPECTED CONTENTS
4549 033716 012737 067501 001322 MOV      #EM1022, EM3N+2 ;; LOAD ERROR MESSAGE
4550 033724 104003      ERROR      3
4551 033726 016237 000014 001126 75: MOV      RKER(R2), $BDDAT ;; STORE ERROR REG
4552 033734 001406      BEQ      8$              ;; CHECK IF ZERO
4553 033736 005037 001124      CLR      $GDDAT      ;; LOAD EXPECTED CONTENTS

```

```

4554 033742 012737 067537 001322      MOV      #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
4555 033750 104003      ERROR   3
4556 033752 016237 000020 001126 95:    MOV      RKDCYL(R2),SBDDAT ;STORE CYLINDER ADD REG
4557 033760 001406      BEQ     105 ;CHECK IF ZERO
4558 033762 005037 001124      CLR     $GDDAT ;LOAD EXPECTED RESULTS
4559 033766 012737 067634 001322      MOV      #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
4560 033774 104003      ERROR   3
4561 033776 016237 000026 001126 105:    MOV      RKMR1(R2),SBDDAT ;STORE MAINTENANCE REG 1
4562 034004 012737 002000 001124      MOV      #MEWD,$GDDAT ;LOAD EXPECTED CONTENTS
4563 034012 032737 020000 001126      BIT     #ECCW,SBDDAT
4564 034020 001403      BEQ     115
4565 034022 052737 020000 001124      BIS     #ECCW,$GDDAT
4566 034030 023737 001124 001126 115:    CMP     $GDDAT,SBDDAT ;CHECK IF MR1 CORRECT
4567 034036 001404      BEQ     125 ;YES,CONTINUE TEST
4568 034040 012737 067661 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
4569 034046 104003      ERROR   3
4570 034050 016237 000032 001126 125:    MOV      RKECPT(R2),SBDDAT ;STORE ECC PATTERN REG.
4571 034056 001406      BEQ     135 ;CHECK IF ZERO
4572 034060 005037 001124      CLR     $GDDAT ;LOAD EXPECTED CONTENTS
4573 034064 012737 067737 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
4574 034072 104003      ERROR   3
4575 034074 016237 000030 001126 135:    MOV      RKECPS(R2),SBDDAT ;STORE ECC POSITION REG:
4576 034102 022737 004066 001126      CMP     #4066,SBDDAT ;CHECK IF ECC POSITION CORRECT
4577 034110 001407      BEQ     TST43 ;YES,GO TO NEXT TEST
4578 034112 012737 004066 001124      MOV      #4066,$GDDAT ;LOAD EXPECTED CONTENTS
4579 034120 012737 067712 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
4580 034126 104003      ERROR   3

```

```

*****
:TEST 43 REGISTER INTERACTION USING ERROR REGISTER
:
: ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
: WRITE THE WORD COUNT REGISTER TO ZERO. WRITE ERROR REGISTER
: WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
: INTERACTION TAKES PLACE.
:
*****

```

```

4591 034130 000004      TST43: SCOPE
4592 034132 012737 000764 001200      MOV      #500, $TIMES ;DO 500. ITERATIONS
4593 034140 013702 001270      MOV      $BASE,R2 ;LOAD RK611 BASE ADDRESS
4594 034144 012737 177777 002010      MOV      #177777,CONFIG ;LOAD CONFIGURATION WORD
4595 034152 012737 064072 001320      MOV      #EM11,EM3N ;LOAD ERROR MESSAGE
4596 034160 012762 100000 000000      MOV      #CLR,RKCSI(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4597 034166 005062 000002      CLR     RKWC(R2) ;CLEAR WORD COUNT REG.
4598 034172 012762 177777 000014      MOV      #177777,RKER(R2) ;WRITE RKER WITH 177777
4599 034200 016237 000000 001126      MOV      RKCS1(R2),SBDDAT ;STORE COMMAND AND STATUS REG.1
4600 034206 022737 000200 001126      CMP     #RDY,SBDDAT ;CHECK IF CS1 CORRECT
4601 034214 001407      BEQ     15 ;YES,CONTINUE
4602 034216 012737 000200 001124      MOV      #RDY,$GDDAT ;LOAD EXPECTED RESULTS
4603 034224 012737 067324 001322      MOV      #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
4604 034232 104003      ERROR   3
4605 034234 016237 000004 001126 15:    MOV      RKBA(R2),SBDDAT ;STORE BUS ADD REG.
4606 034242 001406      BEQ     25 ;CHECK IF ZERO
4607 034244 005037 001124      CLR     $GDDAT ;LOAD EXPECTED CONTENTS
4608 034250 012737 067375 001322      MOV      #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
4609 034256 104003      ERROR   3

```

4610	034260	016237	000002	001126	25:	MOV	RKWC(R2), \$BDDAT	; STORE WORK COUNT REG.
4611	034266	001406				BEQ	35	; CHECK IF ZERO
4612	034270	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4613	034274	012737	067345	001322		MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
4614	034302	104003				ERROR	3	
4615	034304	016237	000006	001126	35:	MOV	RKCA(R2), \$BDDAT	; STORE DISK ADD REG
4616	034312	001406				BEQ	45	; CHECK IF ZERO
4617	034314	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4618	034320	012737	067422	001322		MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
4619	034326	104003				ERROR	3	
4620	034330	016237	000016	001126	45:	MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG
4621	034336	001406				BEQ	55	; CHECK IF ZERO
4622	034340	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4623	034344	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
4624	034352	104003				ERROR	3	
4625	034354	016237	000010	001126	55:	MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG. 2
4626	034362	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
4627	034370	001407				BEQ	65	; YES, CONTINUE
4628	034372	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
4629	034400	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
4630	034406	104003				ERROR	3	
4631	034410	016237	000012	001126	65:	MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG.
4632	034416	001406				BEQ	75	; CHECK IF ZERO
4633	034420	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4634	034424	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
4635	034432	104003				ERROR	3	
4636	034434	016237	000014	001126	75:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
4637	034442	001406				BEQ	85	; CHECK IF ZERO
4638	034444	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4639	034450	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
4640	034456	104003				ERROR	3	
4641	034460	016237	000020	001126	85:	MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
4642	034466	001406				BEQ	105	; CHECK IF ZERO
4643	034470	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED RESULTS
4644	034474	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
4645	034502	104003				ERROR	3	
4646	034504	016237	000026	001126	105:	MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG 1
4647	034512	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED CONTENTS
4648	034520	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4649	034526	001403				BEQ	115	
4650	034530	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4651	034536	023737	001124	001126	115:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT
4652	034544	001404				BEQ	125	; YES, CONTINUE TEST
4653	034546	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
4654	034554	104003				ERROR	3	
4655	034556	016237	000032	001126	125:	MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
4656	034564	001406				BEQ	135	; CHECK IF ZERO
4657	034566	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4658	034572	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
4659	034600	104003				ERROR	3	
4660	034602	016237	000030	001126	135:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG:
4661	034610	022737	004066	001126		CMP	#4066, \$BDDAT	; CHECK IF ECC POSITION CORRECT
4662	034616	001407				BEQ	TST44	; YES, GO TO NEXT TEST
4663	034620	012737	004066	001124		MOV	#4066, \$GDDAT	; LOAD EXPECTED CONTENTS
4664	034626	012737	067712	001322		MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE
4665	034634	104003				ERROR	3	


```

4666
4667
4668
4669
4670
4671
4672
4673
4674
4675 034636 000004
4676 034640 012737 000764 001200
4677 034646 013702 001270
4678 034652 012737 177777 0J2010
4679 034660 012737 064130 001320
4680 034666 012762 100000 000000
4681 034674 005062 000002
4682 034700 012762 177777 000034
4683 034706 016237 000000 001126
4684 034714 022737 000200 001126
4685 034722 001407
4686 034724 012737 000200 001124
4687 034732 012737 067324 001322
4688 034740 104003
4689 034742 016237 000004 001126 1$:
4690 034750 001406
4691 034752 005037 001124
4692 034756 012737 067375 001322
4693 034764 104003
4694 034766 016237 000002 001126 2$:
4695 034774 001406
4696 034776 005037 001124
4697 035002 012737 067345 001322
4698 035010 104003
4699 035012 016237 000006 001126 3$:
4700 035020 001406
4701 035022 005037 001124
4702 035026 012737 067422 001322
4703 035034 104003
4704 035036 016237 000016 001126 4$:
4705 035044 001406
4706 035046 005037 001124
4707 035052 012737 067566 001322
4708 035060 104003
4709 035062 016237 000010 001126 5$:
4710 035070 022737 000100 001126
4711 035076 001407
4712 035100 012737 000100 001124
4713 035106 012737 067460 001322
4714 035114 104003
4715 035116 016237 000012 001126 6$:
4716 035124 001406
4717 035126 005037 001124
4718 035132 012737 067501 001322
4719 035140 104003
4720 035142 016237 000014 001126 7$:
4721 035150 001406

```

```

*****
*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.
*****
↑ST44: SCOPE
MOV #500, $TIMES ; DO 500. ITERATIONS
MOV $BASE, R2 ; LOAD RK611 BASE ADDRESS
MOV #177777, CONFIG ; LOAD CONFIGURATION WORD
MOV #EM12, EM3N ; LOAD ERROR MESSAGE
MOV #CLR, RKCS1(R2) ; CLEAR RK611 WITH CONTROLLER CLEAR
CLR RKWC(R2) ; CLEAR WORD COUNT REG.
MOV #177777, RKMR2(R2) ; WRITE RKMR2 WITH 177777
MOV RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
CMP #RDY, $BDDAT ; CHECK IF CS1 CORRECT
BEQ 1$ ; YES, CONTINUE
MOV #RDY, $GDDAT ; LOAD EXPECTED RESULTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKBA(R2), $BDDAT ; STORE BUS ADD REG.
BEQ 2$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKWC(R2), $BDDAT ; STORE WORK COUNT REG.
BEQ 3$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKDA(R2), $BDDAT ; STORE DISK ADD REG
BEQ 4$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY/OFFSET REG
BEQ 5$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1024, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 2
CMP #IR, $BDDAT ; CHECK IF CS2 CORRECT
BEQ 6$ ; YES, CONTINUE
MOV #IR, $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1021, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKDS(R2), $BDDAT ; STORE DRIVE STATUS REG.
BEQ 7$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1022, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
MOV RKER(R2), $BDDAT ; STORE ERROR REG
BEQ 8$ ; CHECK IF ZERO

```



```

4722 035152 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4723 035156 012737 067537 001322 MOV #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
4724 035164 104003 ERROR 3
4725 03516E 016237 000020 001126 8$: MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
4726 035174 001406 BEQ 10$ ;CHECK IF ZERO
4727 035176 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
4728 035202 012737 067634 001322 MOV #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
4729 035210 104003 ERROR 3
4730 035212 016237 000026 001126 10$: MOV RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG 1
4731 035220 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED CONTENTS
4732 035226 032737 020000 001126 BIT #ECCW, $BDDAT
4733 035234 001403 BEQ 11$
4734 035236 052737 020000 001124 BIS #ECCW, $GDDAT
4735 035244 023737 001124 001126 11$: CMP $GDDAT, $D0DAT ;CHECK IF MRI CORRECT
4736 035252 001404 BEQ 12$ ;YES, CONTINUE TEST
4737 035254 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
4738 035262 104003 ERROR 3
4739 035264 016237 000032 001126 12$: MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
4740 035272 001406 BEQ 13$ ;CHECK IF ZERO
4741 035274 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4742 035300 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
4743 035306 104003 ERROR 3
4744 035310 016237 000030 001126 13$: MOV RKECPS(R2) $BDDAT ;STORE ECC POSITION REG:
4745 035316 022737 004066 001126 CMP #4066, $BDDAT ;CHECK IF ECC POSITION CORRECT
4746 035324 001407 BEQ TST45 ;YES, GO TO NEXT TEST
4747 035326 012737 004066 001124 MOV #4066, $GDDAT ;LOAD EXPECTED CONTENTS
4748 035334 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
4749 035342 104003 ERROR 3

```

4750

4751 *****

4752 *TEST 45 REGISTER INTERACTION USING MAINT REG 3

4753 *

4754 * ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.

4755 * WRITE THE WORD COUNT REGISTER TO ZERO, WRITE MAINTENANCE

4756 * REGISTER 3 TO 177777 AND NO INTERACTION TAKES PLACE.

4757 *

4758 *****

4759 TST45: SCOPE

4760 035344 000004 MOV #500, \$TIMES ;DO 500. ITERATIONS

4761 035346 012737 000764 001200 MOV \$BASE, R2 ;LOAD RK611 BASE ADDRESS

4762 035354 013702 001270 MOV #177777, CONFIG ;LOAD CONFIGURATION WORD

4763 035360 012737 177777 002010 MOV #EM13, EM3N ;LOAD ERROR MESSAGE

4764 035366 012737 064170 001320 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR

4765 035374 012762 100000 000000 CLR RKWC(R2) ;CLEAR WORD COUNT REG.

4766 035402 005062 000002 MOV #177777, RKMR3(R2) ;WRITE RKMR3 WITH 177777

4767 035406 012762 177777 000036 MOV RKCS1(R2), \$BDDAT ;STORE COMMAND AND STATUS REG.1

4768 035414 016237 000000 001126 CMP #RDY, \$BDDAT ;CHECK IF CS1 CORRECT

4769 035422 022737 000200 001126 BEQ 1\$;YES, CONTINUE

4770 035430 001407 MOV #RDY, \$GDDAT ;LOAD EXPECTED RESULTS

4771 035432 012737 000200 001124 MOV #RDY, \$GDDAT ;LOAD EXPECTED RESULTS

4772 035440 012737 067324 001322 MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE

4773 035446 104003 ERROR 3

4774 035450 016237 000004 001126 1\$: MOV RKBA(R2), \$BDDAT ;STORE BUS ADD REG.

4775 035456 001406 BEQ 2\$;CHECK IF ZERO

4776 035460 005037 001124 CLR \$GDDAT ;LOAD EXPECTED CONTENTS

4777 035464 012737 067375 001322 MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE

4778 035472 104003 ERROR 3

4778	035474	016237	000002	001126	25:	MOV	RKWC(R2), \$BDDAT	; STORE WORK COUNT REG.
4779	035502	001406				BEQ	35	; CHECK IF ZERO
4780	035504	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4781	035510	012737	067345	001322		MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
4782	035516	104003				ERROR	3	
4783	035520	016237	000006	001126	35:	MOV	RKDA(R2), \$BDDAT	; STORE DISK ADD REG
4784	035526	001406				BEQ	45	; CHECK IF ZERO
4785	035530	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4786	035534	012737	067422	001322		MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
4787	035542	104003				ERROR	3	
4788	035544	016237	000016	001126	45:	MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG
4789	035552	001406				BEQ	55	; CHECK IF ZERO
4790	035554	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4791	035560	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
4792	035566	104003				ERROR	3	
4793	035570	016237	000010	001126	55:	MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG. 2
4794	035576	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
4795	035604	001407				BEQ	65	; YES, CONTINUE
4796	035606	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
4797	035614	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
4798	035622	104003				ERROR	3	
4799	035624	016237	000012	001126	65:	MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG.
4800	035632	001406				BEQ	75	; CHECK IF ZERO
4801	035634	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4802	035640	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
4803	035646	104003				ERROR	3	
4804	035650	016237	000014	001126	75:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
4805	035656	001406				BEQ	85	; CHECK IF ZERO
4806	035660	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4807	035664	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
4808	035672	104003				ERROR	3	
4809	035674	016237	000020	001126	85:	MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
4810	035702	001406				BEQ	105	; CHECK IF ZERO
4811	035704	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED RESULTS
4812	035710	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
4813	035716	104003				ERROR	3	
4814	035720	016237	000026	001126	105:	MOV	RKMRI(R2), \$BDDAT	; STORE MAINTENANCE REG 1
4815	035726	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED CONTENTS
4816	035734	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4817	035742	001403				BEQ	115	
4818	035744	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4819	035752	023737	001124	001126	115:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MRI CORRECT
4820	035760	001404				BEQ	125	; YES, CONTINUE TEST
4821	035762	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
4822	035770	104003				ERROR	3	
4823	035772	016237	000032	001126	125:	MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
4824	036000	001406				BEQ	135	; CHECK IF ZERO
4825	036002	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
4826	036006	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
4827	036014	104003				ERROR	3	
4828	036016	016237	000030	001126	135:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG:
4829	036024	022737	004066	001126		CMP	#4066, \$BDDAT	; CHECK IF ECC POSITION CORRECT
4830	036032	001407				BEQ	TST46	; YES, GO TO NEXT TEST
4831	036034	012737	004066	001124		MOV	#4066, \$GDDAT	; LOAD EXPECTED CONTENTS
4832	036042	012737	067712	001322		MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE
4833	036050	104003				ERROR	3	

4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889

036052 000004
036054 012737 000764 001200
036062 012701 000021
036066 012737 000001 002010
036074 012737 064466 001320
036102 012737 100000 000000
036110 012737 036116 001110

036116
036116 005062 000002
036122 013762 002010 000020
036130 016237 000020 001126
036136 013737 002010 001124
036144 042737 176000 001124
036152 023737 001124 001126
036160 001404
036162 012737 067302 001322
036170 104003
036172
036172 016237 000000 001126
036200 022737 000200 001126
036206 001407
036210 012737 000100 001124
036216 012737 067324 001322
036224 104003
036226
036226 016237 000004 001126
036234 001406
036236 005037 001124
036242 012737 067375 001322
036250 104003
036252
036252 016237 000002 001126
036260 001406
036262 005037 001124
036266 012737 067345 001322
036274 104003
036276
036276 016237 000006 001126
036304 001406

*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. CHECK IF CONTENTS
* ARE CORRECT AND NO INTERACTION TAKES PLACE.
*
* 000000 000010 000200 004000 100000
* 000001 000020 000400 010000
* 000002 000040 001000 020000
* 000004 000100 002000 040000

TST46: SCOPE
MOV #500, \$TIMES ; DO 500. ITERATIONS
MOV #17, R1 ; LOAD NUMBER OF PATTERNS
MOV #0000C1, CONFIG ; LOAD INITIAL CONFIGURATION
MOV #EM19, EM3N ; LOAD ERROR MESSAGE
MOV #CCLR, RKCS1 ; 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, RKDCYL(R2) ; WRITE RKDCYL
MOV RKDCYL(R2), \$BDDAT ; STORE RKDCYL
MOV CONFIG, \$GDDAT ; PREPARE EXPECTED RESULTS
BIC #176000, \$GDDAT ; INITIALIZE READ ONLY BITS
CMP \$GDDAT, \$BDDAT ; CHECK IF RKDCYL CORRECT
BEQ 2\$; YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1016, 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 #IR, \$GDDAT ; LOAD EXPECTED RESULTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

3\$: MOV RKBA(R2), \$BDDAT ; STORE BUS AND REG
BEQ 4\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

4\$: MOV RKWC(R2), \$BDDAT ; STORE WORD COUNT REG
BEQ 5\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3

5\$: MOV RKDA(R2), \$BDDAT ; STORE DISK AVERAGE REG
BEQ 6\$; CHECK IF ZERO

4890	036306	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4891	036312	012737	067422	001322	MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
4892	036320	104003			ERROR	3	
4893	036322				6\$:		
4894	036322	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY OFFSET REG.
4895	036330	001406			BEQ	7\$;CHECK IF ZERO
4896	036332	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4897	036336	012737	067566	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
4898	036344	104003			ERROR	3	
4899	036346				7\$:		
4900	036346	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2
4901	036354	022737	000100	001126	CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
4902	036362	001407			BEQ	8\$;YES, CONTINUE
4903	036364	012737	000100	001124	MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
4904	036372	012737	067460	001322	MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
4905	036400	104003			ERROR	3	
4906	036402				8\$:		
4907	036402	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
4908	036410	001406			BEQ	9\$;CHECK IF ZERO
4909	036412	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4910	036416	012737	067501	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
4911	036424	104003			ERROR	3	
4912	036426	016237	000014	001126	9\$:		
4913	036434	001406			MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
4914	036436	005037	001124		BEQ	10\$;CHECK IF ZERO
4915	036442	012737	067537	001322	CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4916	036450	104003			MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
4917	036452				ERROR	3	
4918	036452	016237	000026	001126	10\$:		
4919	036460	012737	002000	001124	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
4920	036466	032737	020000	001126	MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MRI
4921	036474	001403			BIT	#ECCW, \$BDDAT	
4922	036476	052737	020000	001124	BEQ	13\$	
4923	036504	023737	001124	001126	BIS	#ECCW, \$GDDAT	
4924	036512	001404			13\$:		
4925	036514	012737	067661	001322	CMP	\$GDDAT, \$BDDAT	;CHECK IF MRI CORRECT
4926	036522	104003			BEQ	14\$;YES, ISSUE CONTROLLER CLEAR
4927	036524				MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
4928	036524	016237	000032	001126	ERROR	3	
4929	036532	001406			14\$:		
4930	036534	005037	001124		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
4931	036540	012737	067737	001322	BEQ	15\$;CHECK IF ZERO
4932	036546	104003			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4933	036550	016237	000030	001126	MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
4934	036556	012737	004066	001124	ERROR	3	
4935	036564	023737	001124	001126	15\$:		
4936	036572	001404			MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
4937	036574	012737	067712	001322	MOV	#4066, \$GDDAT	;USE 4066
4938	036602	104003			16\$:		
4939	036604	016237	000020	002014	17\$:		
4940	036612	012762	100000	000000	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT
4941	036620	016237	000020	001126	BEQ	18\$;YES, INITIALIZE RK611
4942	036626	005037	001124		MOV	#EM1029,EM3N+2	;LOAD ERROR MESSAGE
4943	036632	023737	001124	001126	ERROR	3	
4944	036640	001407			18\$:		
4945	036642	012737	063413	001330	MOV	RKDCYL(R2), PREREG	;GET PREVIOUS CONTENTS
					MOV	#CCLR, RKCS1(R2)	;CLEAR RK611 CONTROLLER
					MOV	RKDCYL(R2), \$BDDAT	;GET CURRENT VALUE
					CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
					CMP	\$GDDAT, \$BDDAT	;CHECK IF RKDCYL CORRECT
					BEQ	19\$;YES, CHECK IF FINISHED
					MOV	#EM3,EM4N	;LOAD ERROR MESSAGE

```

4946 036650 012737 067302 001332      MOV      #EM1016,EM4N+2
4947 036656 104004                      ERROR    4
4948 036660 104415      19$:      SCOPI      ;CHECK IF LOOP ON ERROR
4949 036662 000241                      CLC      ;SHIFT IN ZERO
4950 036664 006137 002010      ROL      CONFIG
4951 036670 005301                      DEC      R1      ;CHECK IF FINISHED
4952 036672 001402                      BEQ      TST47   ;;YES, GO ON TO NEXT TEST
4953 036674 000137 036116      JMP      15
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968

```

```

*****
*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. CHECK IF CONTENTS
*      ARE CORRECT AND NO INTERACTION TAKES PLACE.
*

```

```

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

```

```

*****
TST47:

```

```

4969 036700 000004      SCOPE
4970 036702 012737 000764 001200      MOV      #500, $TIMES      ;DO 500. ITERATIONS
4971 036710 012701 000021                      MOV      #17, R1          ;LOAD NUMBER OF PATTERNS
4972 036714 012737 177776 002010      MOV      #177776, CONFIG ;LOAD INITIAL CONFIGURATION
4973 036722 012737 064466 001320      MOV      #EM19,EM3N      ;LOAD ERROR MESSAGE
4974 036730 012737 100000 000000      MOV      #CCLR,RKCS1    ;CLEAR RK611 WITH CONTROLLER CLEAR
4975 036736 012737 036744 001110      MOV      #15,$LPERR     ;LOAD LOOP ON ERROR LOCATION FOR
*      ; SUBTEST LOOP
4976
4977

```

```

4978 036744      15:      CLR      RKWC(R2)      ;CLEAR WORD COUNT REG.
4979 036744 005062 000002                      MOV      CONFIG,RKDCYL(R2) ;WRITE RKDCYL
4980 036750 013762 002010 000020      MOV      RKDCYL(R2), $BDDAT ;STORE RKDCYL
4981 036756 016237 000020 001126      MOV      CONFIG,$GDDAT   ;PREPARE EXPECTED RESULTS
4982 036764 013737 002010 001124      BIC      #176000,$GDDAT  ;INITIALIZE READ ONLY BITS
4983 036772 042737 176000 001124      CMP      $GDDAT,$BDDAT  ;CHECK IF RKDCYL CORRECT
4984 037000 023737 001124 001126      BEQ      25            ;YES, TEST IF ANY OTHER REG MODIFIED
4985 037006 001404                      MOV      #EM1016,EM3N+2 ;LOAD ERROR MESSAGE
4986 037010 012737 067302 001322      ERROR    3
4987 037016 104003
4988 037020

```

```

4989 037020 016237 000000 001126      25:      MOV      RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG. 1
4990 037026 022737 000200 001126      CMP      #RDY,$BDDAT    ;CHECK IF CSI CORRECT
4991 037034 001407                      BEQ      35            ;YES, CONTINUE
4992 037036 012737 000100 001124      MOV      #IR,$GDDAT    ;LOAD EXPECTED RESULTS
4993 037044 012737 067324 001322      MOV      #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
4994 037052 104003      ERROR    3
4995 037054

```

```

4996 037054 016237 000004 001126      35:      MOV      RKBA(R2), $BDDAT ;STORE BUS AND REG
4997 037062 001406                      BEQ      45            ;CHECK IF ZERO
4998 037064 005037 001124                      CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
4999 037070 012737 067375 001322      MOV      #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
5000 037076 104003      ERROR    3
5001 037100

```

```

45:

```

5002	037100	016237	000002	001126	MOV	RKWC(R2), \$BDDAT	; STORE WORD COUNT REG
5003	037106	001406			BEQ	5\$; CHECK IF ZERO
5004	037110	005037	001124		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5005	037114	012737	067345	001322	MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
5006	037122	104003			ERROR	3	
5007	037124						5\$:
5008	037124	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	; STORE DISK AVERAGE REG
5009	037132	001406			BEQ	6\$; CHECK IF ZERO
5010	037134	005037	001124		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5011	037140	012737	067422	001322	MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
5012	037146	104003			ERROR	3	
5013	037150						6\$:
5014	037150	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG.
5015	037156	001406			BEQ	7\$; CHECK IF ZERO
5016	037160	005037	001124		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5017	037164	012737	067566	001322	MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
5018	037172	104003			ERROR	3	
5019	037174						7\$:
5020	037174	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG.2
5021	037202	022737	000100	001126	CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
5022	037210	001407			BEQ	8\$; YES, CONTINUE
5023	037212	012737	000100	001124	MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
5024	037220	012737	067460	001322	MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
5025	037226	104003			ERROR	3	
5026	037230						8\$:
5027	037230	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
5028	037236	001406			BEQ	9\$; CHECK IF ZERO
5029	037240	005037	001124		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5030	037244	012737	067501	001322	MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
5031	037252	104003			ERROR	3	
5032	037254	016237	000014	001126	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
5033	037262	001406			BEQ	10\$; CHECK IF ZERO
5034	037264	005037	001124		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5035	037270	012737	067537	001322	MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
5036	037276	104003			ERROR	3	
5037	037300						10\$:
5038	037300	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG.1
5039	037306	012737	002000	001124	MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MRI
5040	037314	032737	020000	001126	BIT	#ECCW, \$BDDAT	
5041	037322	001403			BEQ	13\$	
5042	037324	052737	020000	001124	BIS	#ECCW, \$GDDAT	
5043	037332	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	; CHECK IF MRI CORRECT
5044	037340	001404			BEQ	14\$; YES, ISSUE CONTROLLER CLEAR
5045	037342	012737	067661	001322	MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
5046	037350	104003			ERROR	3	
5047	037352						14\$:
5048	037352	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
5049	037360	001406			BEQ	15\$; CHECK IF ZERO
5050	037362	005037	001124		CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5051	037366	012737	067737	001322	MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
5052	037374	104003			ERROR	3	
5053	037376	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG.
5054	037404	012737	004066	001124	MOV	#4066, \$GDDAT	; USE 4066
5055	037412	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	; CHECK IF ECC POSITION CORRECT
5056	037420	001404			BEQ	18\$; YES, INITIALIZE RK611
5057	037422	012737	067712	001322	MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE


```

5058 037430 104003          ERROR 3
5059 037432 016237 000020 002014 18$: MOV RKDCYL(R2),PREREG ;GET PREVIOUS CONTENTS
5060 037440 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5061 037446 016237 000020 001126 MOV RKDCYL(R2),SBDDAT ;GET CURRENT VALUE
5062 037454 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5063 037460 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKDCYL CORRECT
5064 037466 001407 BEQ 19$ ;YES, CHECK IF FINISHED
5065 037470 012737 063413 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
5066 037476 012737 067302 001332 MOV #EM1016,EM4N+2
5067 037504 104004          ERROR 4
5068 037506 104415          19$: SCOPE1 ;CHECK IF LOOP ON ERROR
5069 037510 000261          SEC ;SHIFT IN ONE
5070 037512 006137 002010 ROL CONFIG
5071 037516 005301          DEC R1 ;CHECK IF FINISHED
5072 037520 001402          BEQ TST50 ;;YES, GO ON TO NEXT TEST
5073 037522 000137 036744          JMP 1$
5074
5075
5076 *****
5077 *TEST 50 REGISTER INTERACTION WITH DISK CYLINDER (PART 3)
5078 *
5079 * ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5080 * WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK
5081 * CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
5082 * ARE CORRECT AND NO INTERACTION TAKES PLACE.
5083 *
5084 * 000001 000037 000777 017777 000000
5085 * 000003 000077 001777 037777
5086 * 000007 000177 003777 077777
5087 * 000017 000377 007777 177777
5088 *
5089 *****
5090 037526 000004          †TST50: SCOPE
5091 037530 012737 000764 001200 MOV #500,$TIMES ;;DO 500. ITERATIONS
5092 037536 012701 000021 MOV #17,R1 ;LOAD NUMBER OF PATTERNS
5093 037542 005037 002010 CLR CONFIG ;LOAD INITIAL CONFIGURATION
5094 037546 012737 064466 001320 MOV #EM19,EM3N ;LOAD ERROR MESSAGE
5095 037554 012737 100000 000000 MOV #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
5096 037562 012737 037570 001110 MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
5097 ; SUBTEST LOOP
5098
5099 037570          1$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
5100 037574 005062 000002 MOV CONFIG,RKDCYL(R2) ;WRITE RKDCYL
5101 037602 013762 002010 000020 MOV RKDCYL(R2),SBDDAT ;STORE RKDCYL
5102 037610 013737 000020 001126 MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
5103 037616 042737 176000 001124 BIC #176000,$GDDAT ;INITIALIZE READ ONLY BITS
5104 037624 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKDCYL CORRECT
5105 037632 001404 BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
5106 037634 012737 067302 001322 MOV #EM1016,EM3N+2 ;LOAD ERROR MESSAGE
5107 037642 104003          ERROR 3
5108
5109 037644          2$: MOV RKCS1(R2),SBDDAT ;STORE COMMAND AND STATUS REG. 1
5110 037652 022737 000200 001126 CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT
5111 037660 001407 BEQ 3$ ;YES, CONTINUE
5112 037662 012737 000100 001124 MOV #IR,$GDDAT ;LOAD EXPECTED RESULTS
5113 037670 012737 067324 001322 MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE

```


5114	037676	104003				ERROR	3	
5115	037700				3\$:			
5116	037700	016237	000004	001126		MOV	RKBA(R2), \$BDDAT	; STORE BUS AND REG
5117	037706	001406				BEQ	4\$; CHECK IF ZERO
5118	037710	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5119	037714	012737	067375	001322		MOV	#EM1019, EM3N+2	; LOAD ERROR MESSAGE
5120	037722	104003				ERROR	3	
5121	037724				4\$:			
5122	037724	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	; STORE WORD COUNT REG
5123	037732	001406				BEQ	5\$; CHECK IF ZERO
5124	037734	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5125	037740	012737	067345	001322		MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
5126	037746	104003				ERROR	3	
5127	037750				5\$:			
5128	037750	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	; STORE DISK AVERAGE REG
5129	037756	001406				BEQ	6\$; CHECK IF ZERO
5130	037760	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5131	037764	012737	067422	001322		MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
5132	037772	104003				ERROR	3	
5133	037774				6\$:			
5134	037774	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG.
5135	040002	001406				BEQ	7\$; CHECK IF ZERO
5136	040004	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5137	040010	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
5138	040016	104003				ERROR	3	
5139	040020				7\$:			
5140	040020	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG.2
5141	040026	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
5142	040034	001407				BEQ	8\$; YES, CONTINUE
5143	040036	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
5144	040044	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
5145	040052	104003				ERROR	3	
5146	040054				8\$:			
5147	040054	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
5148	040062	001406				BEQ	9\$; CHECK IF ZERO
5149	040064	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5150	040070	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
5151	040076	104003				ERROR	3	
5152	040100	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
5153	040106	001406				BEQ	10\$; CHECK IF ZERO
5154	040110	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5155	040114	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
5156	040122	104003				ERROR	3	
5157	040124				10\$:			
5158	040124	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG.1
5159	040132	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MR1
5160	040140	032737	020000	001126		BIT	#ECCW, \$BDDAT	
5161	040146	001403				BEQ	13\$	
5162	040150	052737	020000	001124		BIS	#ECCW, \$GDDAT	
5163	040156	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT
5164	040164	001404				BEQ	14\$; YES, ISSUE CONTROLLER CLEAR
5165	040166	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
5166	040174	104003				ERROR	3	
5167	040176				14\$:			
5168	040176	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
5169	040204	001406				BEQ	15\$; CHECK IF ZERO

```

5170 04C206 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5171 040212 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5172 040220 104003 ERROR 3
5173 040222 016237 000030 001126 15$: MOV RKDCPS(R2), $BDDAT ;STORE ECC POSITION REG.
5174 040230 012737 0C4066 001124 16$: MOV #4066, $GDDAT ;USE 4066
5175 040236 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
5176 040244 001404 BEQ 18$ ;YES, INITIALIZE RK6!!
5177 040246 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5178 040254 104003 ERROR 3
5179 040256 016237 000020 002014 18$: MOV RKDCYL(R2), PREREG ;GET PREVIOUS CONTENTS
5180 040264 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
5181 040272 016237 000020 001126 MOV RKDCYL(R2), $BDDAT ;GET CURRENT VALUE
5182 040300 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5183 040304 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKDCYL CORRECT
5184 040312 001407 BEQ 19$ ;YES, CHECK IF FINISHED
5185 040314 012737 063413 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
5186 040322 012737 067302 001332 MOV #EM1016, EM4N+2
5187 040330 104004 ERROR 4
5188 040332 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
5189 040334 000261 SEC ;SHIFT IN ONE
5190 040336 006137 002010 ROL CONFIG
5191 040342 005301 DEC R1 ;CHECK IF FINISHED
5192 040344 001402 BEQ TST51 ;;YES, GO ON TO NEXT TEST
5193 040346 000137 037570 JMP 1$

*****
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
*
*****
TST51: SCOPE
MOV #500, $TIMES ;;DO 500. ITERATIONS
MOV #17, R1 ;LOAD NUMBER OF PATTERNS
CLR CONFIG ;LOAD INITIAL CONFIGURATION
MOV #EM19, EM3N ;LOAD ERROR MESSAGE
MOV #CCLR, RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV CONFIG, RKDCYL(R2) ;WRITE RKDCYL
MOV RKDCYL(R2), $BDDAT ;STORE RKDCYL
MOV CONFIG, $GDDAT ;PREPARE EXPECTED RESULTS
BIC #176000, $GDDAT ;INITIALIZE READ ONLY BITS
CMP $GDDAT, $BDDAT ;CHECK IF RKDCYL CORRECT
BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED

```

C10

 RA611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MAC11 27(732) 01-OCT-76 10:19 PAGE 97
 DZR6A8.P11 T51 REGISTER INTERACTION WITH DISK CYLINDER (PART 4)

SEG 0098

5226	040460	012737	067302	001322	MOV	#EM1016,EM3N+2	;LOAD ERROR MESSAGE
5227	040466	104003			ERROR	3	
5228	040470				25:		
5229	040470	016237	000000	001126	MOV	RKCS1(R2), \$BDDAT	;STORE COMMAND AND STATUS REG. 1
5230	040476	022737	000200	001126	CMP	#RDY, \$BDDAT	;CHECK IF CS1 CORRECT
5231	040504	001407			BEQ	35	;YES, CONTINUE
5232	040506	012737	000100	001124	MOV	#IR, \$GDDAT	;LOAD EXPECTED RESULTS
5233	040514	012737	067324	001322	MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
5234	040522	104003			ERROR	3	
5235	040524				35:		
5236	040524	016237	000004	001126	MOV	RKBA(R2), \$BDDAT	;STORE BUS AND REG
5237	040532	001406			BEQ	45	;CHECK IF ZERO
5238	040534	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5239	040540	012737	067375	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
5240	040546	104003			ERROR	3	
5241	040550				45:		
5242	040550	016237	000002	001126	MOV	RKWC(R2), \$BDDAT	;STORE WORD COUNT REG
5243	040556	001406			BEQ	55	;CHECK IF ZERO
5244	040560	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5245	040564	012737	067345	001322	MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
5246	040572	104003			ERROR	3	
5247	040574				55:		
5248	040574	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	;STORE DISK AVERAGE REG
5249	040602	001406			BEQ	65	;CHECK IF ZERO
5250	040604	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5251	040610	012737	067422	001322	MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
5252	040616	104003			ERROR	3	
5253	040620				65:		
5254	040620	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
5255	040626	001406			BEQ	75	;CHECK IF ZERO
5256	040630	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5257	040634	012737	067566	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
5258	040642	104003			ERROR	3	
5259	040644				75:		
5260	040644	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2
5261	040652	022737	000100	001126	CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
5262	040660	001407			BEQ	85	;YES, CONTINUE
5263	040662	012737	000100	001124	MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
5264	040670	012737	067460	001322	MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
5265	040676	104003			ERROR	3	
5266	040700				85:		
5267	040700	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
5268	040706	001406			BEQ	95	;CHECK IF ZERO
5269	040710	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5270	040714	012737	067501	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
5271	040722	104003			ERROR	3	
5272	040724	016237	000014	001126	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
5273	040732	001406			BEQ	105	;CHECK IF ZERO
5274	040734	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5275	040740	012737	067537	001322	MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
5276	040746	104003			ERROR	3	
5277	040750				105:		
5278	040750	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
5279	040756	012737	002000	001124	MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MRI
5280	040764	032737	020000	001126	BIT	#ECCW, \$BDDAT	
5281	040772	001403			BEQ	135	

```

5282 040774 052737 020000 001124      BIS      #ECCW,$GDDAT
5283 041002 023737 001124 001126 13$:    CMP      $GDDAT,$BDDAT ;CHECK IF MRI CORRECT
5294 041010 001404      BEQ      14$ ;YES,ISSUE CONTRCLLR CLEAR
5285 041012 012737 067661 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
5286 041020 104003      ERROR   3
5287 041022      14$:
5288 041022 016237 000032 001126      MOV      RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5289 041030 001406      BEQ      15$ ;CHECK IF ZERO
5290 041032 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5291 041036 012737 067737 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5292 041044 104003      ERROR   3
5293 041046 016237 000030 001126 15$:    MOV      RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
5294 041054 012737 004066 001124 16$:    MOV      #4066,$GDDAT ;USE 4066
5295 041062 023737 001124 001126 17$:    CMP      $GDDAT,$BDDAT ;CHECK IF ECC POSITION CORRECT
5296 041070 001404      BEQ      18$ ;YES,INITIALIZE RK611
5297 041072 012737 067712 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5298 041100 104003      ERROR   3
5299 041102 016237 000020 002014 18$:    MOV      RKDCYL(R2),PREREG ;GET PREVIOUS CONTENTS
5300 041110 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
5301 041116 016237 000020 001126      MOV      RKDCYL(R2),$BDDAT ;GET CURRENT VALUE
5302 041124 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5303 041130 023737 001124 001126      CMP      $GDDAT,$BDDAT ;CHECK IF RKDCYL CORRECT
5304 041136 001407      BEQ      19$ ;YES,CHECK IF FINISHED
5305 041140 012737 063413 001330      MOV      #EM3,EM4N ;LOAD ERROR MESSAGE
5306 041146 012737 067302 001332      MOV      #EM1016,EM4N+2
5307 041154 104004      ERROR   4
5308 041156 104415      19$:    SCOPE1 ;CHECK IF LOOP ON ERROR
5309 041160 000261      SEC      ;SHIFT IN ONE
5310 041162 006037 002010      ROR      CONFIG
5311 041166 005301      DEC      R1 ;CHECK IF FINISHED
5312 041170 001402      BEQ      TST52 ;;YES, GO ON TO NEXT TEST
5313 041172 000137 040414      JMP      1$

```

```

*****
;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 CONTENTS
; ARE CORRECT AND NO INTERACTION TAKES PLACE.

```

```

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

```

```

*****
;TST52: SCOPE

```

```

5329 041176 000004      MOV      #500,$TIMES ;DO 500. ITERATIONS
5330 041200 012737 000764 001200      MOV      #17,R1 ;LOAD NUMBER OF PATTERNS
5331 041206 012701 000021 002010      MOV      #000001,CONFIG ;LOAD INITIAL CONFIGURATION
5332 041212 012737 064527 001320      MOV      #EM20,EM3N ;LOAD ERROR MESSAGE
5333 041220 012737 064527 001320      MOV      #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
5334 041226 012737 100000 000000      MOV      #1$,SLPERR ;LOAD LOOP ON ERROR LOCATION FOR
5335 041234 012737 041242 001110      MOV      ; SUBTEST LOOP
5336
5337

```

E10

5338	041242						1\$:				
5339	041242	005062	000002					CLR	RKWC(R2)	;CLEAR WORD COUNT REG.	
5340	041246	013762	002010	000026				MOV	CONFIG,RKMR1(R2)	;WRITE RKMR1	
5341	041254	016237	003026	001126				MOV	RKMR1(R2), \$BDDAT	;STORE RKMR1	
5342	041262	013737	002010	001124				MOV	CONFIG, \$GDDAT	;PREPARE EXPECTED RESULTS	
5343	041270	042737	174000	001124				BIC	#PCA!PC!ECCW!WR!GAT!RDGATE, \$GDDAT	;INITIALIZE READ ONLY BITS	
5344	041276	052737	002000	001124				BIS	#MEWD, \$GDDAT		
5345	041304	032737	020000	001126				BIT	#ECCW, \$BDDAT		
5346	041312	001403						BEQ	30\$		
5347	041314	052737	020000	001124				BIS	#ECCW, \$GDDAT		
5348	041322						30\$:				
5349	041322	023737	001124	001126				CMP	\$GDDAT, \$BDDAT	;CHECK IF RKMR1 CORRECT	
5350	041330	001404						SEQ	2\$;YES TEST IF ANY OTHER REG MODIFIED	
5351	041332	012737	067102	001322				MOV	#EM1009, EM3N+2	;LOAD ERROR MESSAGE	
5352	041340	104003						ERROR	3		
5353	041342						2\$:				
5354	041342	016237	000000	001126				MOV	RKCS1(R2), \$BDDAT	;STORE COMMAND AND STATUS REG. 1	
5355	041350	022737	000200	001126				CMP	#RDY, \$BDDAT	;CHECK IF CS1 CORRECT	
5356	041356	001407						BEQ	3\$;YES, CONTINUE	
5357	041360	012737	000100	001124				MOV	#IR, \$GDDAT	;LOAD EXPECTED RESULTS	
5358	041366	012737	067324	001322				MOV	#EM1017, EM3N+2	;LOAD ERROR MESSAGE	
5359	041374	104003						ERROR	3		
5360	041376						3\$:				
5361	041376	016237	000004	001126				MOV	RKBA(R2), \$BDDAT	;STORE BUS AND REG	
5362	041404	001406						BEQ	4\$;CHECK IF ZERO	
5363	041406	005037	001124					CLR	\$GDDAT	;LOAD EXPECTED CONTENTS	
5364	041412	012737	067375	001322				MOV	#EM1019, EM3N+2	;LOAD ERROR MESSAGE	
5365	041420	104003						ERROR	3		
5366	041422						4\$:				
5367	041422	016237	000002	001126				MOV	RKWC(R2), \$BDDAT	;STORE WORD COUNT REG	
5368	041430	001406						BEQ	5\$;CHECK IF ZERO	
5369	041432	005037	001124					CLR	\$GDDAT	;LOAD EXPECTED CONTENTS	
5370	041436	012737	067345	001322				MOV	#EM1018, EM3N+2	;LOAD ERROR MESSAGE	
5371	041444	104003						ERROR	3		
5372	041446						5\$:				
5373	041446	016237	000006	001126				MOV	RKDA(R2), \$BDDAT	;STORE DISK AVERAGE REG	
5374	041454	001406						BEQ	6\$;CHECK IF ZERO	
5375	041456	005037	001124					CLR	\$GDDAT	;LOAD EXPECTED CONTENTS	
5376	041462	012737	067422	001322				MOV	#EM1020, EM3N+2	;LOAD ERROR MESSAGE	
5377	041470	104003						ERROR	3		
5378	041472						6\$:				
5379	041472	016237	000016	001126				MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.	
5380	041500	001406						BEQ	7\$;CHECK IF ZERO	
5381	041502	005037	001124					CLR	\$GDDAT	;LOAD EXPECTED CONTENTS	
5382	041506	012737	067566	001322				MOV	#EM1024, EM3N+2	;LOAD ERROR MESSAGE	
5383	041514	104003						ERROR	3		
5384	041516						7\$:				
5385	041516	016237	000010	001126				MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2	
5386	041524	022737	000100	001126				CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT	
5387	041532	001407						BEQ	8\$;YES, CONTINUE	
5388	041534	012737	000100	001124				MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS	
5389	041542	012737	067460	001322				MOV	#EM1021, EM3N+2	;LOAD ERROR MESSAGE	
5390	041550	104003						ERROR	3		
5391	041552						8\$:				
5392	041552	016237	000012	001126				MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG	
5393	041560	001406						BEQ	9\$;CHECK IF ZERO	

F10

5394	041562	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5395	041566	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
5396	041574	104003				ERROR	3	
5397	041576	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
5398	041604	001406				BEQ	10\$;CHECK IF ZERO
5399	041606	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5400	041612	012737	067537	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
5401	041620	104003				ERROR	3	
5402	041622				10\$:			
5403	041622	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
5404	041630	001406				BEQ	12\$;CHECK IF ZERO
5405	041632	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5406	041636	012737	067634	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
5407	041644	104003				ERROR	3	
5408	041646				12\$:			
5409	041646	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
5410	041654	001406				BEQ	15\$;CHECK IF ZERO
5411	041656	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5412	041662	012737	067737	001322		MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
5413	041670	104003				ERROR	3	
5414	041672	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
5415	041700	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	;USE 4066
5416	041706	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT
5417	041714	001404				BEQ	18\$;YES, INITIALIZE RK611
5418	041716	012737	067712	001322		MOV	#EM1029,EM3N+2	;LOAD ERROR MESSAGE
5419	041724	104003				ERROR	3	
5420	041726	016237	000026	002014	18\$:	MOV	RKMR1(R2), PREREG	;GET PREVIOUS CONTENTS
5421	041734	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	;CLEAR RK611 CONTROLLER
5422	041742	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	;GET CURRENT VALUE
5423	041750	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED CONTENTS
5424	041756	032737	020000	001126		BIT	#ECCW, \$BDDAT	
5425	041764	001403				BEQ	35\$	
5426	041766	052737	020000	001124		BIS	#ECCW, \$GDDAT	
5427	041774				35\$:			
5428	041774	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	;CHECK IF RKMR1 CORRECT
5429	042002	001407				BEQ	19\$;YES, CHECK IF FINISHED
5430	042004	012737	063413	001330		MOV	#EM3, EM4N	;LOAD ERROR MESSAGE
5431	042012	012737	067102	001332		MOV	#EM1009, EM4N+2	
5432	042020	104004				ERROR	4	
5433	042022	104415			19\$:	SCOP1		;CHECK IF LOOP ON ERROR
5434	042024	000241				CLC		;SHIFT IN ZERO
5435	042026	006137	002010			ROL	CONFIG	
5436	042032	005301				DEC	R1	;CHECK IF FINISHED
5437	042034	001402				BEQ	TST53	;YES, GO ON TO NEXT TEST
5438	042036	000137	041242			JMP	1\$	

```

*****
*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 CONTENTS
* ARE CORRECT AND NO INTERACTION TAKES PLACE.
*
* 177777 177767 177577 173777 077777
* 177776 177757 177377 167777

```

5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449

G10

```

5450          : *      177775 177737 176777 157777
5451          : *      177773 177677 175777 137777
5452          : *
5453          : *
5454          : *****
5454 042042 000004 ST53: SCOPE
5455 042044 012737 000764 001200 MOV #500, $TIMES ; DO 500. ITERATIONS
5456 042052 012701 000021 000026 MOV #17, R1 ; LOAD NUMBER OF PATTERNS
5457 042056 012737 177776 002010 MOV #177776, CONFIG ; LOAD INITIAL CONFIGURATION
5458 042064 012737 064527 001320 MOV #EM20, EM3N ; LOAD ERROR MESSAGE
5459 042072 012737 100000 000000 MOV #CCLR, RKCS1 ; CLEAR RK611 WITH CONTROLLER CLEAR
5460 042100 012737 042106 001110 MOV #15, $LPERR ; LOAD LOOP ON ERROR LOCATION FOR
5461          : ; SUBTEST LOOP
5462
5463 042106 15:
5464 042106 005062 000002 CLR RKWC(R2) ; CLEAR WORD COUNT REG.
5465 042112 013762 002010 000026 MOV CONFIG, RKMRI(R2) ; WRITE RKMRI
5466 042120 016237 000026 001126 MOV RKMRI(R2), $BDDAT ; STORE RKMRI
5467 042126 013737 002010 001124 MOV CONFIG, $GDDAT ; PREPARE EXPECTED RESULTS
5468 042134 042737 174000 001124 BIC #PCA!PCD!ECCW!WRTGAT!RDGATE, $GDDAT ; INITIALIZE READ ONLY BITS
5469 042142 052737 002000 001124 BIS #MEWD, $GDDAT
5470 042150 032737 020000 001126 BIT #ECCW, $BDDAT
5471 042156 001403 BEQ 30$
5472 042160 052737 020000 001124 BIS #ECCW, $GDDAT
5473 042166 30$:
5474 042166 023737 001124 001126 CMP $GDDAT, $BDDAT ; CHECK IF RKMRI CORRECT
5475 042174 001404 BEQ 2$ ; YES, TEST IF ANY OTHER REG MODIFIED
5476 042176 012737 067102 001322 MOV #EM1009, EM3N+2 ; LOAD ERROR MESSAGE
5477 042204 104003 ERROR 3
5478 042206 2$:
5479 042206 016237 000000 001126 MOV RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
5480 042214 022737 000200 001126 CMP #RDY, $BDDAT ; CHECK IF CS1 CORRECT
5481 042222 001407 BEQ 3$ ; YES, CONTINUE
5482 042224 012737 000100 001124 MOV #IR, $GDDAT ; LOAD EXPECTED RESULTS
5483 042232 012737 067324 001322 MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
5484 042240 104003 ERROR 3
5485 042242 3$:
5486 042242 016237 000004 001126 MOV RKBA(R2), $BDDAT ; STORE BUS AND REG
5487 042250 001406 BEQ 4$ ; CHECK IF ZERO
5488 042252 005037 001124 CLR $GDDAT ; LOAD EXPECTED CONTENTS
5489 042256 012737 067375 001322 MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
5490 042264 104003 ERROR 3
5491 042266 4$:
5492 042266 016237 000002 001126 MOV RKWC(R2), $BDDAT ; STORE WORD COUNT REG
5493 042274 001406 BEQ 5$ ; CHECK IF ZERO
5494 042276 005037 001124 CLR $GDDAT ; LOAD EXPECTED CONTENTS
5495 042302 012737 067345 001322 MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
5496 042310 104003 ERROR 3
5497 042312 5$:
5498 042312 016237 000006 001126 MOV RKDA(R2), $BDDAT ; STORE DISK AVERAGE REG
5499 042320 001406 BEQ 6$ ; CHECK IF ZERO
5500 042322 005037 001124 CLR $GDDAT ; LOAD EXPECTED CONTENTS
5501 042326 012737 067422 001322 MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
5502 042334 104003 ERROR 3
5503 042336 6$:
5504 042336 016237 000016 001126 MOV RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY/OFFSET REG.
5505 042344 001406 BEQ 7$ ; CHECK IF ZERO
    
```


H10

5506	042346	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5507	042352	012737	067566	001322	MOV	#EM1024,EM3N+2		;LOAD ERROR MESSAGE
5508	042360	104003			ERROR	3		
5509	042362						7\$:	
5510	042362	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT		;STORE COMMAND AND STATUS REG.2
5511	042370	022737	000100	001126	CMP	#IR, \$BDDAT		;CHECK IF CS2 CORRECT
5512	042376	001407			BEQ	8\$;YES, CONTINUE
5513	042400	012737	000100	001124	MOV	#IR, \$GDDAT		;LOAD EXPECTED CONTENTS
5514	042406	012737	067460	001322	MOV	#EM1021,EM3N+2		;LOAD ERROR MESSAGE
5515	042414	104003			ERROR	3		
5516	042416						8\$:	
5517	042416	016237	000012	001126	MOV	RKDS(R2), \$BDDAT		;STORE DRIVE STATUS REG
5518	042424	001406			BEQ	9\$;CHECK IF ZERO
5519	042426	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5520	042432	012737	067501	001322	MOV	#EM1022,EM3N+2		;LOAD ERROR MESSAGE
5521	042440	104003			ERROR	3		
5522	042442	016237	000014	001126	MOV	RKER(R2), \$BDDAT		;STORE ERROR REG
5523	042450	001406			BEQ	10\$;CHECK IF ZERO
5524	042452	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5525	042456	012737	067537	001322	MOV	#EM1023,EM3N+2		;LOAD ERROR MESSAGE
5526	042464	104003			ERROR	3		
5527	042466						10\$:	
5528	042466	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT		;STORE CYLINDER ADD REG
5529	042474	001406			BEQ	12\$;CHECK IF ZERO
5530	042476	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5531	042502	012737	067634	001322	MOV	#EM1025,EM3N+2		;LOAD ERROR MESSAGE
5532	042510	104003			ERROR	3		
5533	042512						12\$:	
5534	042512	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT		;STORE ECC PATTERN REG.
5535	042520	001406			BEQ	15\$;CHECK IF ZERO
5536	042522	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5537	042526	012737	067737	001322	MOV	#EM1030,EM3N+2		;LOAD ERROR MESSAGE
5538	042534	104003			ERROR	3		
5539	042536	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT		;STORE ECC POSITION REG.
5540	042544	012737	004066	001124	MOV	#4066, \$GDDAT		;USE 4066
5541	042552	023737	001124	001126	CMP	\$GDDAT, \$BDDAT		;CHECK IF ECC POSITION CORRECT
5542	042560	001404			BEQ	18\$;YES, INITIALIZE RK611
5543	042562	012737	067712	001322	MOV	#EM1029,EM3N+2		;LOAD ERROR MESSAGE
5544	042570	104003			ERROR	3		
5545	042572	016237	000026	002014	MOV	RKMR1(R2), PREREG		;GET PREVIOUS CONTENTS
5546	042600	012762	100000	000000	MOV	#CCLR, RKCS1(R2)		;CLEAR RK611 CONTROLLER
5547	042606	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT		;GET CURRENT VALUE
5548	042614	012737	002000	001124	MOV	#MEWD, \$GDDAT		;LOAD EXPECTED CONTENTS
5549	042622	032737	020000	001126	BIT	#ECCW, \$BDDAT		
5550	042630	001403			BEQ	35\$		
5551	042632	052737	020000	001124	BIS	#ECCW, \$GDDAT		
5552	042640						35\$:	
5553	042640	023737	001124	001126	CMP	\$GDDAT, \$BDDAT		;CHECK IF RKMR1 CORRECT
5554	042646	001407			BEQ	19\$;YES, CHECK IF FINISHED
5555	042650	012737	063413	001330	MOV	#EM3, EM4N		;LOAD ERROR MESSAGE
5556	042656	012737	067102	001332	MOV	#EM1009,EM4N+2		
5557	042664	104004			ERROR	4		
5558	042666	104415					19\$:	
5559	042670	000261			SCOP1			;CHECK IF LOOP ON ERROR
5560	042672	006137	002010		SEC			;SHIFT IN ONE
5561	042676	005301			ROL	CONFIG		
					DEC	R1		;CHECK IF FINISHED

5562 042700 001402 BEQ TST54 ;;YES, GO ON TO NEXT TEST
5563 042702 000137 042106 JMP 15

*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 CONTENTS
* ARE CORRECT AND NO INTERACTION TAKES PLACE.

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

*TST54: SCOPE

5578 042706 000004
5579 042710 012737 000764 001200 MOV #500, \$TIMES ;;DO 500. ITERATIONS
5580 042716 012701 000021 001200 MOV #17, R1 ;;LOAD NUMBER OF PATTERNS
5581 042722 005037 002010 001200 CLR CONFIG ;;LOAD INITIAL CONFIGURATION
5582 042726 012737 064527 001320 MOV #EM20, EM3N ;;LOAD ERROR MESSAGE
5583 042734 012737 100000 000000 MOV #CLR, RKCS1 ;;CLEAR RK611 WITH CONTROLLER CLEAR
5584 042742 012737 042750 001110 MOV #15, \$LPERR ;;LOAD LOOP ON ERROR LOCATION FOR
5585 042750 005062 000002 001126 15: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
5586 042750 013762 002010 000026 MOV CONFIG, RKMR1(R2) ;WRITE RKMR1
5587 042754 016237 000026 001126 MOV RKMR1(R2), \$BDDAT ;STORE RKMR1
5588 042762 013737 002010 001124 MOV CONFIG, \$GDDAT ;PREPARE EXPECTED RESULTS
5589 042770 042737 174000 001124 BIC #PCA!PCD!ECCW!WRTGAT!RDGATE, \$GDDAT ;INITIALIZE READ ONLY BITS
5590 042776 052737 002000 001124 BIS #MEWD, \$GDDAT
5591 043004 032737 020000 001124 BIT #ECCW, \$BDDAT
5592 043012 001403 BEQ 30\$
5593 043020 052737 020000 001124 BIS #ECCW, \$GDDAT
5594 043030 023737 001124 30\$: CMP \$GDDAT, \$BDDAT ;CHECK IF RKMR1 CORRECT
5595 043036 001404 BEQ 2\$;YES, TEST IF ANY OTHER REG MODIFIED
5596 043040 012737 067102 001322 MOV #EM1009, EM3N+2 ;LOAD ERROR MESSAGE
5597 043046 104003 ERROR 3
5598 043050 016237 000000 001126 2\$: MOV RKCS1(R2), \$BDDAT ;STORE COMMAND AND STATUS REG. 1
5599 043056 022737 000200 001126 CMP #RDY, \$BDDAT ;CHECK IF CS1 CORRECT
5600 043064 001407 BEQ 3\$;YES, CONTINUE
5601 043066 012737 000100 001124 MOV #IR, \$GDDAT ;LOAD EXPECTED RESULTS
5602 043074 012737 067324 001322 MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE
5603 043102 104003 ERROR 3
5604 043104 016237 000004 001126 3\$: MOV RKBA(R2), \$BDDAT ;STORE BUS AND REG
5605 043112 001406 BEQ 4\$;CHECK IF ZERO
5606 043114 005037 001124 CLR \$GDDAT ;LOAD EXPECTED CONTENTS
5607 043120 012737 067375 001322 MOV #EM1019, EM3N+2 ;LOAD ERROR MESSAGE
5608 043126 104003 ERROR 3
5609 043130 016237 000002 001126 4\$: MOV RKWC(R2), \$BDDAT ;STORE WORD COUNT REG
5610 043130

5618	043136	001406			BEQ	5\$;CHECK IF ZERO
5619	043140	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5620	043144	012737	067345	001322	MOV	#EM1018,EM3N+2		;LOAD ERROR MESSAGE
5621	043152	104003			ERROR	3		
5622	043154				5\$:			
5623	043154	C16237	000006	001126	MOV	RKDA(R2), \$BDDAT		;STORE DISK AVERAGE REG
5624	043162	001406			BEQ	6\$;CHECK IF ZERO
5625	043164	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5626	043170	012737	067422	001322	MOV	#EM1020,EM3N+2		;LOAD ERROR MESSAGE
5627	043176	104003			ERROR	3		
5628	043200				6\$:			
5629	043200	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT		;STORE ATTENTION SUMMARY/OFFSET REG.
5630	043206	001406			BEQ	7\$;CHECK IF ZERO
5631	043210	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5632	043214	012737	067566	001322	MOV	#EM1024,EM3N+2		;LOAD ERROR MESSAGE
5633	043222	104003			ERROR	3		
5634	043224				7\$:			
5635	043224	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT		;STORE COMMAND AND STATUS REG.2
5636	043232	022737	000100	001126	CMP	#IR, \$BDDAT		;CHECK IF CS2 CORRECT
5637	043240	001407			BEQ	8\$;YES, CONTINUE
5638	043242	012737	000100	001124	MOV	#IR, \$GDDAT		;LOAD EXPECTED CONTENTS
5639	043250	012737	067460	001322	MOV	#EM1021,EM3N+2		;LOAD ERROR MESSAGE
5640	043256	104003			ERROR	3		
5641	043260				8\$:			
5642	043260	016237	000012	001126	MOV	RKDS(R2), \$BDDAT		;STORE DRIVE STATUS REG
5643	043266	001406			BEQ	9\$;CHECK IF ZERO
5644	043270	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5645	043274	012737	067501	001322	MOV	#EM1022,EM3N+2		;LOAD ERROR MESSAGE
5646	043302	104003			ERROR	3		
5647	043304	016237	000014	001126	MOV	RKER(R2), \$BDDAT		;STORE ERROR REG
5648	043312	001406			BEQ	10\$;CHECK IF ZERO
5649	043314	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5650	043320	012737	067537	001322	MOV	#EM1023,EM3N+2		;LOAD ERROR MESSAGE
5651	043326	104003			ERROR	3		
5652	043330				10\$:			
5653	043330	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT		;STORE CYLINDER ADD REG
5654	043336	001406			BEQ	12\$;CHECK IF ZERO
5655	043340	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5656	043344	012737	067634	001322	MOV	#EM1025,EM3N+2		;LOAD ERROR MESSAGE
5657	043352	104003			ERROR	3		
5658	043354				12\$:			
5659	043354	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT		;STORE ECC PATTERN REG.
5660	043362	001406			BEQ	15\$;CHECK IF ZERO
5661	043364	005037	001124		CLR	\$GDDAT		;LOAD EXPECTED CONTENTS
5662	043370	012737	067737	001322	MOV	#EM1030,EM3N+2		;LOAD ERROR MESSAGE
5663	043376	104003			ERROR	3		
5664	043400	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT		;STORE ECC POSITION REG.
5665	043406	012737	004066	001124	MOV	#4066, \$GDDAT		;USE 4066
5666	043414	023737	001124	001126	CMP	\$GDDAT, \$BDDAT		;CHECK IF ECC POSITION CORRECT
5667	043422	001404			BEQ	18\$;YES, INITIALIZE RK611
5668	043424	012737	067712	001322	MOV	#EM1029,EM3N+2		;LOAD ERROR MESSAGE
5669	043432	104003			ERROR	3		
5670	043434	016237	000026	002014	MOV	RKMR1(R2), PREREG		;GET PREVIOUS CONTENTS
5671	043442	012762	100000	000000	MOV	#CCLR, RKCS1(R2)		;CLEAR RK611 CONTROLLER
5672	043450	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT		;GET CURRENT VALUE
5673	043456	012737	002000	001124	MOV	#MEWD, \$GDDAT		;LOAD EXPECTED CONTENTS

K10

```

5674 043464 032737 020000 001126 BIT #ECCW,$BDDAT
5675 043472 001403 BEQ 35$
5676 043474 052737 020000 001124 BIS #ECCW,$GDDAT
5677 043502 35$:
5678 043502 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKMRI CORRECT
5679 043510 001407 BEQ 19$ ;YES, CHECK IF FINISHED
5680 043512 012737 063413 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
5681 043520 012737 067102 001332 MOV #EM1009,EM4N+2
5682 043526 104004 ERROR 4
5683 043530 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
5684 043532 000261 SEC ;SHIFT IN ONE
5685 043534 006137 002010 ROL CONFIG
5686 043540 005301 DEC R1 ;CHECK IF FINISHED
5687 043542 001402 SEQ TST55 ;YES, GO ON TO NEXT TEST
5688 043544 000137 042750 JMP 1$
5689
5690 ;*****
5691 ;*TEST 55 REGISTER INTERACTION USING MAINT REG 1 (PART 4)
5692 ;*
5693 ;* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5694 ;* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
5695 ;* REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
5696 ;* ARE CORRECT AND NO INTERACTION TAKES PLACE.
5697 ;*
5698 ;* 100000 174000 177600 177770 000000
5699 ;* 140000 176000 177700 177774
5700 ;* 160000 177000 177740 177776
5701 ;* 170000 177400 177760 177777
5702 ;*
5703 ;*****
5704 043550 000004 TST55: SCOPE
5705 043552 012737 000764 001200 MOV #500,$TIMES ;DO 500. ITERATIONS
5706 043560 012701 000021 MOV #17,R1 ;LOAD NUMBER OF PATTERNS
5707 043564 005037 002010 CLR CONFIG ;LOAD INITIAL CONFIGURATION
5708 043570 012737 064527 001320 MOV #EM20,EM3N ;LOAD ERROR MESSAGE
5709 043576 012737 100000 000000 MOV #CCLR,RKCS1 ;CLEAR RK611 WITH CONTROLLER CLEAR
5710 043604 012737 043612 001110 MOV #1$,SLPERR ;LOAD LOOP ON ERROR LOCATION FOR
5711 ; SUBTEST LOOP
5712
5713 043612 1$:
5714 043612 005062 000002 CLR RKWC(R2) ;CLEAR WORD COUNT REG.
5715 043616 013762 002010 000026 MOV CONFIG,RKMRI(R2) ;WRITE RKMRI
5716 043624 016237 000026 001126 MOV RKMRI(R2),$BDDAT ;STORE RKMRI
5717 043632 013737 002010 001124 MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
5718 043640 042737 174000 001124 BIC #PCA!PCD!ECCW!WRTGAT!RDGATE,$GDDAT ;INITIALIZE READ ONLY BITS
5719 043646 052737 002000 001124 BIS #MEWD,$GDDAT
5720 043654 032737 020000 001126 BIT #ECCW,$BDDAT
5721 043662 001403 BEQ 30$
5722 043664 052737 020000 001124 BIS #ECCW,$GDDAT
5723 043672 30$:
5724 043672 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKMRI CORRECT
5725 043700 001404 BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
5726 043702 012737 067102 001322 MOV #EM1009,EM3N+2 ;LOAD ERROR MESSAGE
5727 043710 104003 ERROR 3
5728 043712 2$:
5729 043712 016237 000000 001126 MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1

```

5730	043720	022737	000200	001126		CMP	#RDY,\$BDDAT	;CHECK IF CS1 CORRECT
5731	043726	001407				BEQ	3\$;YES, CONTINUE
5732	043730	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED RESULTS
5733	043736	012737	067324	001322		MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
5734	043744	104003				ERROR	3	
5735	043746				3\$:			
5736	043746	016237	000004	001126		MOV	RKBA(R2),\$BDDAT	;STORE BUS AND REG
5737	043754	001406				BEQ	4\$;CHECK IF ZERO
5738	043756	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5739	043762	012737	067375	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
5740	043770	104003				ERROR	3	
5741	043772				4\$:			
5742	043772	016237	000002	001126		MOV	RKWC(R2),\$BDDAT	;STORE WORD COUNT REG
5743	044000	001406				BEQ	5\$;CHECK IF ZERO
5744	044002	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5745	044005	012737	067345	001322		MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
5746	044014	104003				ERROR	3	
5747	044016				5\$:			
5748	044016	016237	000006	001126		MOV	RKDA(R2),\$BDDAT	;STORE DISK AVERAGE REG
5749	044024	001406				BEQ	6\$;CHECK IF ZERO
5750	044026	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5751	044032	012737	067422	001322		MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
5752	044040	104003				ERROR	3	
5753	044042				6\$:			
5754	044042	016237	000016	001126		MOV	RKASOF(R2),\$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
5755	044050	001406				BEQ	7\$;CHECK IF ZERO
5756	044052	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5757	044056	012737	067566	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
5758	044064	104003				ERROR	3	
5759	044066				7\$:			
5760	044066	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT	;STORE COMMAND AND STATUS REG.2
5761	044074	022737	000100	001126		CMP	#IR,\$BDDAT	;CHECK IF CS2 CORRECT
5762	044102	001407				BEQ	8\$;YES, CONTINUE
5763	044104	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
5764	044112	012737	067460	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
5765	044120	104003				ERROR	3	
5766	044122				8\$:			
5767	044122	016237	000012	001126		MOV	RKDS(R2),\$BDDAT	;STORE DRIVE STATUS REG
5768	044130	001406				BEQ	9\$;CHECK IF ZERO
5769	044132	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5770	044136	012737	067501	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
5771	044144	104003				ERROR	3	
5772	044146	016237	000014	001126	9\$:	MOV	RKER(R2),\$BDDAT	;STORE ERROR REG
5773	044154	001406				BEQ	10\$;CHECK IF ZERO
5774	044156	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5775	044162	012737	067537	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
5776	044170	104003				ERROR	3	
5777	044172				10\$:			
5778	044172	016237	000020	001126		MOV	RKDCYL(R2),\$BDDAT	;STORE CYLINDER ADD REG
5779	044200	001406				BEQ	12\$;CHECK IF ZERO
5780	044202	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5781	044206	012737	067634	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
5782	044214	104003				ERROR	3	
5783	044216				12\$:			
5784	044216	016237	000032	001126		MOV	RKECPT(R2),\$BDDAT	;STORE ECC PATTERN REG.
5785	044224	001406				BEQ	15\$;CHECK IF ZERO

```

5786 044226 005037 001124          CLR    $GDDAT          ;LOAD EXPECTED CONTENTS
5787 044232 012737 067737 001322    MOV    #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5788 044240 104003          ERROR  3
5789 044242 016237 000030 001126 15$:  MOV    RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
5790 044250 012737 0C4066 001124 16$:  MOV    #4066, $GDDAT    ;USE 4066
5791 044256 023737 001124 001126 17$:  CMP    $GDDAT, $BDDAT  ;CHECK IF ECC POSITION CORRECT
5792 044264 001404          BEQ    18$            ;YES, INITIALIZE RK611
5793 044266 012737 067712 001322    MOV    #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5794 044274 104003          ERROR  3
5795 044276 016237 00C026 002014 18$:  MOV    RKMRI(R2), PREREG ;GET PREVIOUS CONTENTS
5796 044304 012762 100000 000000    MOV    #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
5797 044312 016237 000026 001126    MOV    RKMRI(R2), $BDDAT ;GET CURRENT VALUE
5798 044320 012737 002000 001124    MOV    #MEWD, $GDDAT   ;LOAD EXPECTED CONTENTS
5799 044326 032737 020000 001126    BIT    #ECCW, $BDDAT
5800 044334 001403          BEQ    35$
5801 044336 052737 020000 001124    BIS    #ECCW, $GDDAT
5802 044344          35$:
5803 044344 023737 001124 001126    CMP    $GDDAT, $BDDAT  ;CHECK IF RKMRI CORRECT
5804 044352 001407          BEQ    19$            ;YES, CHECK IF FINISHED
5805 044354 012737 063413 001330    MOV    #EM3, EM4N     ;LOAD ERROR MESSAGE
5806 044362 012737 067102 001332    MOV    #EM1009, EM4N+2
5807 044370 104004          ERROR  4
5808 044372 104415          19$:  SCOPI  ;CHECK IF LOOP ON ERROR
5809 044374 000261          SEC    ;SHIFT IN ONE
5810 044376 006037 002010    ROR    CONFIG
5811 044402 005301          DEC    R1             ;CHECK IF FINISHED
5812 044404 001402          BEQ    TST56         ;;YES, GO ON TO NEXT TEST
5813 044406 000137 043612          JMP    1$
5814
5815 ;*****
5816 ;*TEST 56 REGISTER INTERACTION WITH PATTERN REG.
5817 ;*
5818 ;* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5819 ;* WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE PATTERN
5820 ;* REGISTER TO 177777 AND MAKE SURE IT REMAINS 0 AND
5821 ;* NO INTERACTION TAKES PLACE.
5822 ;*
5823 ;*****
5824 044412 000004          †TST56: SCOPE
5825 044414 012737 000764 001200    MOV    #500, $TIMES   ;DO 500. ITERATIONS
5826 044422 013702 001270          MOV    $BASE, R2     ;LOAD RK611 BASE ADDRESS
5827 044426 012737 177777 002010    MOV    #177777, CONFIG ;LOAD CONFIGURATION WORD
5828 044434 012737 064230 001320    MOV    #EM14, EM3N   ;LOAD ERROR MESSAGE
5829 044442 012762 100000 000000    MOV    #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5830 044450 005062 000002          CLR    RKWC(R2)     ;CLEAR WORD COUNT REG.
5831 044454 012762 177777 000032    MOV    #177777, RKECPT(R2) ;WRITE RKECPT WITH 177777
5832 044462 016237 000000 001126    MOV    RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG.1
5833 044470 022737 000200 001126    CMP    #RDY, $BDDAT  ;CHECK IF CS1 CORRECT
5834 044476 001407          BEQ    1$            ;YES, CONTINUE
5835 044500 012737 000200 001124    MOV    #RDY, $GDDAT  ;LOAD EXPECTED RESULTS
5836 044506 012737 067324 001322    MOV    #EM1017, EM3N+2 ;LOAD ERROR MESSAGE
5837 044514 104003          ERROR  3
5838 044516 016237 000004 001126 1$:  MOV    RKBA(R2), $BDDAT ;STORE BUS ADD REG.
5839 044524 001406          BEQ    2$            ;CHECK IF ZERO
5840 044526 005037 001124          CLR    $GDDAT       ;LOAD EXPECTED CONTENTS
5841 044532 012737 067375 001322    MOV    #EM1019, EM3N+2 ;LOAD ERROR MESSAGE

```


5842	044540	104003				ERROR	3	
5843	044542	016237	000002	001126	2\$:	MOV	RKWC(R2), \$BDDAT	; STORE WORK COUNT REG.
5844	044550	001406				BEQ	3\$; CHECK IF ZERO
5845	044552	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5846	044556	012737	067345	001322		MOV	#EM1018, EM3N+2	; LOAD ERROR MESSAGE
5847	044564	104003				ERROR	3	
5848	044566	016237	000006	001126	3\$:	MOV	RKDA(R2), \$BDDAT	; STORE DISK ADD REG
5849	044574	001406				BEQ	4\$; CHECK IF ZERO
5850	044576	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5851	044602	012737	067422	001322		MOV	#EM1020, EM3N+2	; LOAD ERROR MESSAGE
5852	044610	104003				ERROR	3	
5853	044612	016237	000016	001126	4\$:	MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION SUMMARY/OFFSET REG
5854	044620	001406				BEQ	5\$; CHECK IF ZERO
5855	044622	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5856	044626	012737	067566	001322		MOV	#EM1024, EM3N+2	; LOAD ERROR MESSAGE
5857	044634	104003				ERROR	3	
5858	044636	016237	000010	001126	5\$:	MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG. 2
5859	044644	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
5860	044652	001407				BEQ	6\$; YES, CONTINUE
5861	044654	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
5862	044662	012737	067460	001322		MOV	#EM1021, EM3N+2	; LOAD ERROR MESSAGE
5863	044670	104003				ERROR	3	
5864	044672	016237	000012	001126	6\$:	MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG.
5865	044700	001406				BEQ	7\$; CHECK IF ZERO
5866	044702	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5867	044706	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
5868	044714	104003				ERROR	3	
5869	044716	016237	000014	001126	7\$:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REG
5870	044724	001406				BEQ	8\$; CHECK IF ZERO
5871	044726	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5872	044732	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
5873	044740	104003				ERROR	3	
5874	044742	016237	000020	001126	8\$:	MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
5875	044750	001406				BEQ	10\$; CHECK IF ZERO
5876	044752	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED RESULTS
5877	044756	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
5878	044764	104003				ERROR	3	
5879	044766	016237	000026	001126	10\$:	MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG 1
5880	044774	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED CONTENTS
5881	045002	032737	020000	001126		BIT	#ECCW, \$BDDAT	
5882	045010	001403				BEQ	11\$	
5883	045012	052737	020000	001124		BIS	#ECCW, \$GDDAT	
5884	045020	023737	001124	001126	11\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT
5885	045026	001404				BEQ	12\$; YES, CONTINUE TEST
5886	045030	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
5887	045036	104003				ERROR	3	
5888	045040	016237	000032	001126	12\$:	MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
5889	045046	001406				BEQ	13\$; CHECK IF ZERO
5890	045050	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
5891	045054	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
5892	045062	104003				ERROR	3	
5893	045064	016237	000030	001126	13\$:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG:
5894	045072	022737	004066	001126		CMP	#4066, \$BDDAT	; CHECK IF ECC POSITION CORRECT
5895	045100	001407				BEQ	TST57	; YES, GO TO NEXT TEST
5896	045102	012737	004066	001124		MOV	#4066, \$GDDAT	; LOAD EXPECTED CONTENTS
5897	045110	012737	067712	001322		MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE


```

5898 045116 104003
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909 045120 000004
5910 045122 012737 000764 001200
5911 045130 013702 001270
5912 045134 012737 177777 002010
5913 045142 012737 064270 001320
5914 045150 012762 100000 000000
5915 045156 005062 000002
5916 045162 012762 177777 000030
5917 045170 016237 000000 001126
5918 045176 022737 000200 001126
5919 045204 001407
5920 045206 012737 000200 001124
5921 045214 012737 067324 001322
5922 045222 104003
5923 045224 016237 000004 001126 1$:
5924 045232 001406
5925 045234 005037 001124
5926 045240 012737 067375 001322
5927 045246 104003
5928 045250 016237 000002 001126 2$:
5929 045256 001406
5930 045260 005037 001124
5931 045264 012737 067345 001322
5932 045272 104003
5933 045274 016237 000006 001126 3$:
5934 045302 001406
5935 045304 005037 001124
5936 045310 012737 067422 001322
5937 045316 104003
5938 045320 016237 000016 001126 4$:
5939 045326 001406
5940 045330 005037 001124
5941 045334 012737 067566 001322
5942 045342 104003
5943 045344 016237 000010 001126 5$:
5944 045352 022737 000100 001126
5945 045360 001407
5946 045362 012737 000100 001124
5947 045370 012737 067460 001322
5948 045376 104003
5949 045400 016237 000012 001126 6$:
5950 045406 001406
5951 045410 005037 001124
5952 045414 012737 067501 001322
5953 045422 104003

```

ERROR 3

```

*****
*TEST 57 REGISTER INTERACTION WITH POSITION REG.
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
* WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE POSITION
* REGISTER TO 177777 AND MAKE SURE IT STAYS AT THE
* INITIALIZED CONDITION AND NO INTERACTION TAKES PLACE.
*****

```

```

↑T57: SCOPE
MOV #500,$TIMES ;DO 500. ITERATIONS
MOV $BASE,R2 ;LOAD RK611 BASE ADDRESS
MOV #177777,CONFIG ;LOAD CONFIGURATION WORD
MOV #EM15,EM3N ;LOAD ERROR MESSAGE
MOV #CLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV #177777,RKECPS(R2) ;WRITE RKECPS WITH 177777
MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT
BEQ 1$ ;YES,CONTINUE
MOV #RDY,$GDDAT ;LOAD EXPECTED RESULTS
MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKBA(R2),$BDDAT ;STORE BUS ADD REG.
BEQ 2$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
BEQ 3$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKDA(R2),$BDDAT ;STORE DISK ADD REG
BEQ 4$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
BEQ 5$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1024,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 2
CMP #IR,$BDDAT ;CHECK IF CS2 CORRECT
BEQ 6$ ;YES,CONTINUE
MOV #IR,$GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1021,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG.
BEQ 7$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```

```

5954 045424 016237 000014 001126 7$: MOV RKER(R2),SBDDAT ;STORE ERROR REG
5955 045432 001406 BEQ 8$ ;CHECK IF ZERO
5956 045434 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5957 045440 012737 067537 001322 MOV #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
5958 045446 104003 ERROR 3
5959 045450 016237 000020 001126 8$: MOV RKDCYL(R2),SBDDAT ;STORE CYLINDER ADD REG
5960 045456 001406 BEQ 10$ ;CHECK IF ZERO
5961 045460 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
5962 045464 012737 067634 001322 MOV #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
5963 045472 104003 ERROR 3
5964 045474 016237 000026 001126 10$: MOV RKMR1(R2),SBDDAT ;STORE MAINTENANCE REG 1
5965 045502 012737 002000 001124 MOV #MEWD,$GDDAT ;LOAD EXPECTED CONTENTS
5966 045510 032737 020000 031126 BIT #ECCW,$BDDAT
5967 045516 001403 BEQ 11$
5968 045520 052737 020000 001124 BIS #ECCW,$GDDAT
5969 045526 023737 001124 001126 11$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
5970 045534 001404 BEQ 12$ ;YES,CONTINUE TEST
5971 045536 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
5972 045544 104003 ERROR 3
5973 045546 016237 000032 001126 12$: MOV RKECPT(R2),SBDDAT ;STORE ECC PATTERN REG.
5974 045554 001406 BEQ 13$ ;CHECK IF ZERO
5975 045556 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5976 045562 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5977 045570 104003 ERROR 3
5978 045572 016237 000030 001126 13$: MOV RKECPS(R2),SBDDAT ;STORE ECC POSITION REG:
5979 045600 022737 004066 001126 CMP #4066,$BDDAT ;CHECK IF ECC POSITION CORRECT
5980 045606 001407 BEQ TST60 ;:YES,GO TO NEXT TEST
5981 045610 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED CONTENTS
5982 045616 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5983 045624 104003 ERROR 3

```

.SBTTL **INTERRUPT TESTS

```

5984
5985
5986 .SBTTL **INTERRUPT TESTS
5987
5988 ::*****
5989 *TEST 60 RK611 INTERRUPT
5990 *
5991 * STORE LOCATIONS 0-776, LOAD LOCATIONS 0-776 TO TRAP ALL
5992 * POSSIBLE INTERRUPTS. LOWER PROCESSOR PRIORITY TO ZERO.
5993 * MAKE SURE THAT NO INTERRUPT OCCURS. NOW SET INTERRUPT
5994 * ENABLE AND READY. VERIFY THAT THE INTERRUPT OCCURS AT
5995 * PROPER VECTOR ADDRESS. MAKE SURE THAT INTERRUPT IS
5996 * CLEARED AFTER IT IS GIVEN.
5997 *
5998 ::*****
5999 TST60: SCOPE
6000 MOV #500,$TIMES ;:DO 500. ITERATIONS
6001 MOV #SAVVEC,R1 ;LOAD SAVED ADDRESS
6002 CLR R0 ;LOAD START OF VECTOR SPACE
6003 MOV #400,R3 ;LOAD COUNT
6004 1$: MOV (R0)+,(R1)+ ;SAVE TRAP CATCHER
6005 DEC R3
6006 BNE 1$
6007 MOV #-1,SAVFLG ;INDICATE TRAP CATCHER SAVED
6008 CLR R0 ;LOAD ADDRESS OF START OF VECTOR SPACE
6009 MOV #10$,R1 ;LOAD START OF VECTOR TABLE

```

D11

6010	045672	012703	000200			MOV	#200,R3	;LOAD COUNT
6011	045676	010120		25:		MOV	R1,(R0)+	;LOAD VECTOR SPACE
6012	045700	012720	000340			MOV	#PR7,(R0)+	; PRIORITY 7
6013	045704	062701	000002			ADD	#2,R1	
6014	045710	005303				DEC	R3	
6015	045712	001371				BNE	25	
6016	045714	012762	100000	000000		MOV	#CCLR,RKCS1(R2)	;CLEAR RK611 WITH CONTROLLER CLEAR
6017	045722	012705	046420			MOV	#115,R5	;LOAD ADDRESS FOR UNEXPECTED INTERRUPT
6018	045726	005046				CLR	-(SP)	;LOAD STACK TO ALLOW ALL INTERRUPTS
6019	045730	012746	045736			MOV	#645,-(SP)	;LOAD NEXT ADDRESS
6020	045734	000002				RTI		;CLEAR PSW
6021								
6022	045736			645:				
6023	045736	000240				NOP		;ALLOW INTERRUPT TO OCCUR
6024	045740	012705	046472			MOV	#155,R5	;LOAD ADDRESS FOR EXPECTED INTERRUPT
6025	045744	012762	000300	000000		MOV	#RDY!IE,RKCS1(R2)	;GENERATE INTERRUPT
6026	045752	000240				NOP		;ALLOW INTERRUPT TO OCCUR
6027	045754	012746	000340			MOV	#PR7,-(SP)	;LOCK OUT ALL INTERRUPTS
6028	045760	012746	045766			MOV	#35,-(SP)	
6029	045764	000002				RTI		
6030								
6031	045766	012701	067764	35:		MOV	#SAVVEC,R1	;LOAD SAVED TRAP CATCHER ADDRESS
6032	045772	005000				CLR	R0	;LOAD START OF VECTOR SPACE
6033	045774	012703	000400			MOV	#400,R3	;LOAD COUNT
6034	046000	012120		45:		MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
6035	046002	005303				DEC	R3	
6036	046004	001375				BNE	45	
6037	046006	005037	002016			CLR	SAVFLG	;INDICATE TRAP CATCHER RESTORED
6038	046012	104005				ERROR	5	;REPORT INTERRUPT DID NOT OCCUR
6039	046014	000137	046702			JMP	605	;GO ON TO NEXT TEST
6040								
6041	046020			105:				
6042	046020	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6043	046022	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6044	046024	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6045	046026	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6046	046030	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6047	046032	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6048	046034	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6049	046036	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6050	046040	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6051	046042	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6052	046044	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6053	046046	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6054	046050	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6055	046052	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6056	046054	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6057	046056	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6058	046060	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6059	046062	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6060	046064	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6061	046066	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6062	046070	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6063	046072	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6064	046074	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
6065	046076	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS

6066	046100	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6067	046102	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6068	046104	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6069	046106	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6070	046110	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6071	046112	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6072	046114	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6073	046116	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6074	046120	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6075	046122	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6076	046124	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6077	046126	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6078	046130	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6079	046132	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6080	046134	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6081	046136	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6082	046140	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6083	046142	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6084	046144	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6085	046146	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6086	046150	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6087	046152	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6088	046154	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6089	046156	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6090	046160	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6091	046162	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6092	046164	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6093	046166	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6094	046170	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6095	046172	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6096	046174	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6097	046176	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6098	046200	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6099	046202	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6100	046204	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6101	046206	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6102	046210	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6103	046212	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6104	046214	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6105	046216	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6106	046220	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6107	046222	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6108	046224	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6109	046226	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6110	046230	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6111	046232	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6112	046234	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6113	046236	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6114	046240	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6115	046242	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6116	046244	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6117	046246	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6118	046250	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6119	046252	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6120	046254	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS
6121	046256	004715	JSR	PC, (R5)	;LOAD STACK FOR VECTOR ADDRESS

6122	046260	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6123	046262	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6124	046264	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6125	046266	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6126	046270	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6127	046272	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6128	046274	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6129	046276	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6130	046300	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6131	046302	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6132	046304	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6133	046306	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6134	046310	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6135	046312	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6136	046314	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6137	046316	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6138	046320	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6139	046322	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6140	046324	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6141	046326	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6142	046330	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6143	046332	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6144	046334	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6145	046336	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6146	046340	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6147	046342	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6148	046344	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6149	046346	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6150	046350	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6151	046352	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6152	046354	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6153	046356	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6154	046360	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6155	046362	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6156	046364	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6157	046366	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6158	046370	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6159	046372	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6160	046374	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6161	046376	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6162	046400	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6163	046402	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6164	046404	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6165	046406	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6166	046410	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6167	046412	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6168	046414	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6169	046416	004715			JSR	PC, (R5)	; LOAD STACK FOR VECTOR ADDRESS
6170							
6171	046420	012637	001122		MOV	(SP)+, \$BDA0R	; DETERMINE VECTOR ADDRESS
6172	046424	162737	046022	001122	SUB	#105+2, \$BDA0R	
6173	046432	006337	001122		ASL	\$BDA0R	
6174	046436	062706	000004		ADD	#4, SP	; ADJUST STACK
6175	046442	012701	067764		MOV	#\$AVVEC, R1	; LOAD SAVED TRAP CATCHER ADDRESS
6176	046446	005000			CLR	R0	; LOAD START OF VECTOR SPACE
6177	046450	012703	000400		MOV	#400, R3	; LOAD COUNT

G11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 114
 DZR6A8.P11 T60 RK611 INTERRUPT

SEQ 0115

6178	046454	012120		12\$:	MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
6179	046456	005303			DEC	R3	
6180	046460	001375			BNE	12\$	
6181	046462	005037	002016		CLR	SAVFLG	;INDICATE TRAP CATCHER RESTORED
6182	046466	104006			ERROR	6	;REPORT UNEXPECTED INTERRUPT
6183	046470	000504			BR	TST61	;GO ON TO NEXT TEST
6184							
6185	046472	012637	001122	15\$:	MOV	(SP)+,\$BDADR	;DETERMINE VECTOR ADDRESS
6186	046476	162737	046022	001122	SUB	#10\$+2,\$BDADR	
6187	046504	006337	001122		ASL	\$BDADR	
6188	046510	062706	000004		ADD	#4,SP	;ADJUST STACK
6189	046514	023737	002000	001122	CMP	RKVEC,\$BDADR	;CHECK IF VECTOR ADDRESS CORRECT
6190	046522	001414			BEG	20\$;YES, CONTINUE
6191	046524	012701	067764		MOV	#SAVVEC,R1	;LOAD SAVED TRAP CATCHER ADDRESS
6192	046530	005000			CLR	R0	;START OF VECTOR SPACE
6193	046532	012703	000400		MOV	#400,R3	;LOAD COUNT
6194	046536	012120		16\$:	MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
6195	046540	005303			DEC	R3	
6196	046542	001375			BNE	16\$	
6197	046544	005037	002016		CLR	SAVFLG	;INDICATE TRAP CATCHER RESTORED
6198	046550	104007			ERROR	7	;REPORT INCORRECT VECTOR ADDRESS
6199	046552	000453			BR	TST61	;GO ON TO NEXT TEST
6200							
6201	046554	012705	046632	20\$:	MOV	#25\$,R5	;LOAD ADDRESS FOR UNEXPECTED INTERRUPT
6202	046560	005046			CLR	-(SP)	;LOAD STACK TO ALLOW ALL INTERRUPTS
6203	046562	012746	046570		MOV	#65\$,-(SP)	;LOAD NEXT ADDRESS
6204	046566	000002			RTI		;CLEAR PSW
6205							
6206	046570			65\$:			
6207	046570	012746	000340		MOV	#PR7,-(SP)	;LOCK OUT RK611 INTERRUPTS
6208							
6209	046574	012746	046602		MOV	#21\$,-(SP)	
6210	046600	000002			RTI		
6211							
6212	046602	000240		21\$:	NOP		;ALLOW INTERRUPT TO OCCUR
6213	046604	012701	067764		MOV	#SAVVEC,R1	;LOAD SAVE TRAP CATCHER ADDRESS
6214	046610	005000			CLR	R0	;START OF VECTOR SPACE
6215	046612	012703	000400		MOV	#400,R3 ;LOAD COUNT	
6216	046616	012120		22\$:	MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
6217	046620	005303			DEC	R3	
6218	046622	001375			BNE	22\$	
6219	046624	005037	002016		CLR	SAVFLG	;INDICATE TRAP CATCHER RESTORED
6220	046630	000424			BR	TST61	;GO ON TO NEXT TEST
6221							
6222	046632	012637	001122	25\$:	MOV	(SP)+,\$BDADR	;DETERMINE VECTOR ADDRESS
6223	046636	162737	046022	001122	SUB	#10\$+2,\$BDADR	
6224	046644	006337	001122		ASL	\$BDADR	
6225	046650	062706	000004		ADD	#4,SP	;ADJUST STACK
6226	046654	012701	067764		MOV	#SAVVEC,R1	;LOAD SAVE TRAP CATCHER ADDRESS
6227	046660	005000			CLR	R0	;START OF VECTOR SPACE
6228	046662	012703	000400		MOV	#400,R3	;LOAD COUNT
6229	046666	012120		26\$:	MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
6230	046670	005303			DEC	R3	
6231	046672	001375			BNE	26\$	
6232	046674	005037	002016		CLR	SAVFLG	;INDICATE THAT TRAP CATCHER RESTART
6233							

H11

6234 046700 104010 ERROR 10 ;REPORT ATTENTION DID NOT CLEAR

6235 046702 60\$:

6236
6237 :*****
6238 *TEST 61 INTERRUPT PRIORITY
6239 *

6240 * SET UP PRIORITY TO 1 LESS THAN INTERRUPT PRIORITY.
6241 * WRITE READY WITH INTERRUPT ENABLE. MAKE SURE INTERRUPT OCCURS.
6242 *

6243 * NOW SET UP PRIORITY EQUAL TO INTERRUPT PRIORITY.
6244 * WRITE INTERRUPT ENABLE WITH READY. MAKE SURE INTERRUPT
6245 * DOES NOT OCCUR. NOW LOWER PRIORITY AND MAKE
6246 * INTERRUPT HAS BEEN STORED.
6247 *

6248 :*****

6249 046702 000004 1\$T61: SCOPE

6250 046704 012737 000764 001200 MOV #500,\$TIMES ;DO 500. ITERATIONS

6251 046712 013701 002000 MOV RKVEC,R1 ;LOAD RK611 VECTOR ADDRESS FOR INTERRUPT

6252 046716 012721 047006 MOV #10\$,(R1)+ ; PRIORITY 7

6253 046722 013746 002002 MOV RKPRI,-(SP) ;SET PROCESSOR PRIORITY =

6254 046726 162716 000040 SUB #40,(SP) ; RK611 PRIORITY -1

6255 046732 011646 MOV (SP),-(SP)

6256 046734 006216 ASR (SP)

6257 046736 006216 ASR (SP)

6258 046740 006216 ASR (SP)

6259 046742 006216 ASR (SP)

6260 046744 006216 ASR (SP)

6261 046746 012637 002020 MOV (SP)+,PRIOR

6262 046752 012746 046760 MOV #1\$,-(SP)

6263 046756 000002 RTI

6264
6265 046760 012762 000300 000000 1\$: MOV #RDY!IE,RKCS1(R2) ;GENERATE RK611 INTERRUPT

6266 046766 000240 NOP ;WAIT FOR INTERRUPT

6267 046770 012746 000340 MOV #PR7,-(SP) ;LOCK OUT INTERRUPTS

6268 046774 012746 047002 MOV #2\$,-(SP)

6269 047000 000002 RTI

6270
6271 047002 104011 2\$: ERROR 11 ;EXPECTED INTERRUPT DID NOT OCCUR AT

6272 ; PROCESSOR PRIORITY

6273 047004 000462 BR 60\$;RESTORE TRAP CATCHER

6274
6275 047006 062706 000004 10\$: ADD #4,SP ;ADJUST STACK

6276 047012 012777 047136 132760 MOV #20\$,RKVEC ;LOAD RK611 VECTOR FOR UNEXPECTED

6277 ; INTERRUPT

6278 047020 013737 002002 002020 MOV RKPRI,PRIOR ;STORE PRIORITY PRINT OUT

6279 047026 006237 002020 ASR PRIOR

6280 047032 006237 002020 ASR PRIOR

6281 047036 006237 002020 ASR PRIOR

6282 047042 006237 002020 ASR PRIOR

6283 047046 006237 002020 ASR PRIOR

6284 047052 013746 002002 MOV RKPRI,-(SP) ;SET PROCESSOR PRIORITY

6285 047056 012746 047064 MOV #11\$,-(SP) ; RK611 PRIORITY

6286 047062 000002 RTI

6287
6288 047064 012762 000300 000000 11\$: MOV #RDY!IE,RKCS1(R2) ;GENERATE RK611 INTERRUPT

6289 047072 000240 NOP ;ALLOW INTERRUPT TO OCCUR


```

6290 047074 005037 002020          CLR      PRIOR      ;LOAD PRIORITY FOR PRINT OUT
6291 047100 012777 047146 132672    MOV      #25$,@RKVEC ;LOAD RK611 VECTOR FOR INTERRUPT
6292 047106 005046          CLR      -(SP)      ;LOAD STACK TO ALLOW ALL INTERRUPTS
6293 047110 012746 047116    MOV      #64$,-(SP) ;LOAD NEXT ADDRESS
6294 047114 000002          RTI           ;CLEAR PSW
6295
6296 047116          64$:
6297 047116 000240          NOP           ;ALLOW INTERRUPT TO OCCUR
6298 047120 012746 000340    MOV      #PR7,-(SP) ;LOCK OUT INTERRUPTS
6299 047124 012746 047132    MOV      #12$,-(SP)
6300 047130 000002          RTI
6301
6302 047132 104013          12$: ERROR 13      ; INTERRUPT DID NOT OCCUR WHEN
6303                                ; PRIORITY LOWERED
6304 047134 000406          BR      60$      ; RESTORE TRAP CATCHER
6305
6306 047136 062706 000004          20$: ADD      #4,SP ; ADJUST STACK
6307 047142 104012          ERROR 12      ; UNEXPECTED INTERRUPT OCCURRED
6308                                ; AT PROCESSOR PRIORITY
6309 047144 000402          BR      60$      ; RESTORE TRAP CATCHER
6310
6311 047146 062706 000004          25$: ADD      #4,SP ; ADJUST STACK
6312
6313 047152 013701 002000          60$: MOV      RKVEC,R1 ; RESTORE TRAP CATCHER
6314 047156 010111          MOV      R1,(R1)
6315 047160 062721 000002          ADD      #2,(R1)+
6316 047164 005011          CLR      (R1)
6317
6318                                ;*****
6319                                ;*TEST 62      SETTING INTERRUPT ENABLE
6320                                ;*
6321                                ;*      CLEAR RK611 CONTROLLER WITH CONTROLLER CLEAR. ALLOW RK611
6322                                ;*      INTERRUPTS BY SETTING PROCESSOR PRIORITY TO ZERO.
6323                                ;*      SET INTERRUPT ENABLE AND MAKE SURE NO INTERRUPTS OCCUR.
6324                                ;*
6325                                ;*****
6326 047166 000004          TST62: SCOPE
6327 047170 012737 000764 001200    MOV      #500,$TIMES ;:DO 500. ITERATIONS
6328 047176 012762 100000 000000    MOV      #CLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
6329 047204 013701 002000          MOV      RKVEC,R1 ;LOAD RK611 VECTOR ADDRESS FOR
6330 047210 012721 047252          MOV      #10$, (R1)+ ; UNEXPECTED INTERRUPT
6331 047214 012711 000340          MOV      #PR7,(R1) ; PRIORITY 7
6332 047220 005046          CLR      -(SP) ;LOAD STACK TO ALLOW ALL INTERRUPTS
6333 047222 012746 047230          MOV      #64$,-(SP) ;LOAD NEXT ADDRESS
6334 047226 000002          RTI           ;CLEAR PSW
6335
6336 047230          64$:
6337 047230 012762 000100 000000    MOV      #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
6338 047236 000240          NOP           ;ALLOW INTERRUPT TO OCCUR
6339 047240 012746 000340          MOV      #PR7,-(SP) ;LOCK OUT ALL INTERRUPTS
6340 047244 012746 047260          MOV      #15$,-(SP) ;RESTORE TRAP CATCHER
6341 047250 000002          RTI
6342
6343 047252 062706 000004          10$: ADD      #4,SP ;ADJUST STACK
6344 047256 104014          ERROR 14      ;REPORT ERROR
6345 047260 013701 002000          15$: MOV      RKVEC,R1 ;RESTORE TRAP CATCHER

```

```

6346 047264 010111
6347 047266 012721 000002
6348 047272 005011
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359 047274 000004
6360 047276 012737 000764 001200
6361 047304 012762 100000 000000
6362 047312 012762 000300 000000
6363 047320 012762 100000 000000
6364 047326 012762 000100 000000
6365 047334 013701 002000
6366 047340 012721 047374
6367 047344 012711 000340
6368 047350 005046
6369 047352 012746 047360
6370 047356 000002
6371
6372 047360
6373 047360 000240
6374 047362 012746 000340
6375 047366 012746 047402
6376 047372 000002
6377
6378 047374 062706 000004
6379 047400 104015
6380 047402 013701 002000
6381 047406 010111
6382 047410 062721 000002
6383 047414 005011
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395 047416 000004
6396 047420 012737 000764 001200
6397 047426 013702 001270
6398 047432 012762 100000 000000
6399 047440 005762 000024
6400 047444 016237 000010 001710
6401 047452 016237 000000 001700

```

```

*****
*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.
*
*****
†ST63:  SCOPE
MOV      #500, $TIMES      ;; DO 500. ITERATIONS
MOV      #CCLR, RKCS1(R2)  ;; CLEAR RK611
MOV      #IE!RDY, RKCS1(R2) ;; GENERATE INTERRUPT
MOV      #CCLR, RKCS1(R2)  ;; CLEAR INTERRUPT
MOV      #IE, RKCS1(R2)    ;; GET INTERRUPT ENABLE
MOV      RKVEC, R1         ;; LOAD RK611 VECTOR ADDRESS FOR
                          ;; UNEXPECTED INTERRUPT
MOV      #10$, (R1)+       ;; PRIORITY 7
MOV      #PR7, (R1)
CLR      -(SP)             ;; LOAD STACK TO ALLOW ALL INTERRUPTS
MOV      #64$, -(SP)       ;; LOAD NEXT ADDRESS
RTI
64$:
NOP
MOV      #PR7, -(SP)       ;; ALLOW INTERRUPT TO OCCUR
MOV      #15$, -(SP)      ;; LOCK OUT INTERRUPTS
RTI
10$:  ADD      #4, SP       ;; ADJUST STACK
      ERROR   15          ;; REPORT ERROR
15$:  MOV      RKVEC, R1   ;; RESTORE TRAP CATCHER
      MOV      R1, (R1)
      ADD      #2, (R1)+
      CLR      (R1)

.SBTTL  **SILO TESTS
*****
*TEST 64      READ SILO WHEN EMPTY
*
*      READ SILO WHEN EMPTY.  CHECK FOR DATA LATE AND CONTROLLER
*      ERROR.  ISSUE CONTROLLER CLEAR AND CHECK IF ERROR RESET.
*
*****
†ST64:  SCOPE
MOV      #500, $TIMES      ;; DO 500. ITERATIONS
MOV      $BASE, R2         ;; LOAD RK611 BASE
MOV      #CCLR, RKCS1(R2)  ;; CLEAR RK611 CONTROLLER
TST      RKDB(R2)         ;; READ DATA BUFFER
MOV      RKCS2(R2), T.CS2  ;; STORE COMMAND AND STATUS REG.2
MOV      RKCS1(R2), T.CS1  ;; STORE COMMAND AND STATUS REG. 1

```

49

```

6402 047460 012737 100100 001750      MOV      #DLT!IR,E.CS2      ;LOAD EXPECTED CS2
6403 047466 012737 100200 001740      MOV      #CERR!RDY,E.CS1   ;LOAD EXPECTED CS1
6404 047474 023737 001750 001710      CMP      E.CS2,T.CS2      ;CHECK FOR DAT LATE SET
6405 047502 001401                BEQ      15                 ;YES, CHECK FOR CONTROLLER ERROR
6406 047504 104016                ERROR    16                 ;CS2 INCORRECT AFTER READING
6407                                ;EMPTY SILO
6408 047506 023737 001740 001700 15:    CMP      E.CS1,T.CS1      ;CHECK FOR CONTROLLER ERROR
6409 047514 001401                BEQ      25                 ;YES, CLEAR DATA LATE
6410 047516 104017                ERROR    17                 ;CS1 INCORRECT AFTER REQUIRE
6411                                ;EMPTY SILO
6412 047520 012762 100000 000000 25:    MOV      #CCLR,RKCS1(R2)   ;CLEAR RK611 CONTROLLER
6413 047526 016237 000000 001700      MOV      RKCS1(R2),T.CS1   ;STORE COMMAND AND STATUS REG. 1
6414 047534 016237 000010 001710      MOV      RKCS2(R2),T.CS2   ;STORE COMMAND AND STATUS REG.2
6415 047542 012737 000200 001740      MOV      #RDY,E.CS1       ;LOAD EXPECTED CS1
6416 047550 012737 000100 001750      MOV      #IR,E.CS2        ;LOAD EXPECTED CS2
6417 047556 023737 001740 001700      CMP      E.CS1,T.CS1      ;CHECK TO CONTROLLER ERROR RESET
6418 047564 001401                BEQ      35                 ;YES, CHECK IF DATA LATE RESET
6419 047566 104020                ERROR    20                 ;CS1 INCORRECT AFTER ATTEMPTING
6420                                ;TO CLEAR DATA LATE
6421 047570 023737 001750 001710 35:    CMP      E.CS2,T.CS2      ;CHECK IF DATA LATE RESET
6422 047576 001401                BEQ      TST65             ;YES, GO ON TO NEXT TEST
6423 047600 104021                ERROR    21                 ;CS2 INCORRECT AFTER ATTEMPTING
6424                                ;TO OCCUR DATA LATE

```

: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 17777 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 17777. CHECK FOR NO CONTROLLER
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.

TST65: SCOPE

```

6444 047602 000004                MOV      #500,$TIMES      ;;DO 500. ITERATIONS
6445 047604 012737 000764 001200      MOV      #CCLR,RKCS1(R2)  ;CLEAR RK611 CONTROLLER
6446 047612 012762 100000 000000      CLR      RKWC(R2)         ;CLEAR WORD COUNT REG
6447 047620 005062 000002                MOV      #17777,CONFIG    ;LOAD CONFIGURATION FOR PRINT OUT
6448 047624 012737 177777 002010      MOV      #17777,RKDB(R2)  ;WRITE DATA BUFFER WITH ALL 1'S
6449 047632 012762 177777 000024      MOV      RKCS1(R2),%BDDAT ;STORE COMMAND AND STATUS REG 1
6450 047640 016237 000000 001126      CMP      #RDY,%BDDAT      ;CHECK IF CS1 CORRECT
6451 047646 022737 000200 001126      BEQ      15                 ;YES, CHECK BUS AND REG
6452 047654 001407                MOV      #RDY,%GDDAT      ;LOAD EXPECTED CONTENTS
6453 047656 012737 000200 001124      MOV      #EM1017,EM3N+2  ;LOAD ERROR MESSAGE
6454 047664 012737 067324 001322      ERROR    3
6455 047672 104003                MOV      RKBA(R2),%BDDAT  ;STORE BUS AND
6456 047674 016237 000004 001126 15:    BEQ      25                 ;CHECK IF ZERO
6457 047702 001406

```

6458	047704	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6459	047710	012737	067375	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
6460	047716	104003			ERROR	3	
6461	047720	016237	000002	001126	2\$:	MOV	RKWC(R2), \$BDDAT ;STORE WORD COUNT REG
6462	047726	001406			BEQ	3\$;CHECK IF ZERO
6463	047730	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6464	047734	012737	067345	001322	MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
6465	047742	104003			ERROR	3	
6466	047744	016237	000006	001126	3\$:	MOV	RKDA(R2), \$BDDAT ;STORE DESK ADDRESS REG
6467	047752	001406			BEQ	4\$;CHECK IF ZERO
6468	047754	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6469	047760	012737	067422	001322	MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
6470	047766	104003			ERROR	3	
6471	047770	016237	000016	001126	4\$:	MOV	RKASOF(R2), \$BDDAT ;STORE ATTENTION SUMMARY AND OFFSET
6472	047776	001406			BEQ	5\$;CHECK IF ZERO
6473	050000	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6474	050004	012737	067712	001322	MOV	#EM1029,EM3N+2	;LOAD ERROR MESSAGE
6475	050012	104003			ERROR	3	
6476	050014	012700	000005		5\$:	MOV	#5, R0 ;LOAD COUNTER TO WAIT FOR OUTPUT READY
6477	050020	005300			6\$:	DEC	R0
6478	050022	001376			BNE	6\$	
6479	050024	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG. 2
6480	050032	022737	000300	001126	CMP	#IR!OR, \$BDDAT	;CHECK IF CS2 CORRECT
6481	050040	001407			BEQ	7\$;YES, CONTINUE TEST
6482	050042	012737	000300	001124	MOV	#IR!OR, \$GDDAT	;LOAD EXPECTED CONTENTS
6483	050050	012737	066714	001322	MOV	#EM1004,EM3N+2	;LOAD ERROR MESSAGE
6484	050056	104003			ERROR	3	
6485	050060	016237	000012	001126	7\$:	MOV	RKDS(R2), \$BDDAT ;STORE DRIVE STATUS REG.
6486	050066	001406			BEQ	8\$;CHECK IF ZERO
6487	050070	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6488	050074	012737	067501	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
6489	050102	104003			ERROR	3	
6490	050104	016237	000014	001126	8\$:	MOV	RKER(R2), \$BDDAT ;STORE ERROR REGISTER
6491	050112	001406			BEQ	9\$;CHECK IF ZERO
6492	050114	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6493	050120	012737	067537	001322	MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
6494	050126	104003			ERROR	3	
6495	050130	016237	000020	001126	9\$:	MOV	RKDCYL(R2), \$BDDAT ;STORE CYLINDER ADDRESS REG.
6496	050136	001406			BEQ	11\$;CHECK IF ZERO
6497	050140	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6498	050144	012737	067634	001322	MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
6499	050152	104003			ERROR	3	
6500	050154	016237	000026	001126	11\$:	MOV	RKMR1(R2), \$BDDAT ;STORE MAINTENANCE REG. 1
6501	050162	012737	002000	001124	MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MRI
6502	050170	032737	020000	001126	BIT	#ECCW, \$BDDAT	
6503	050176	001403			BEQ	12\$	
6504	050200	052737	020000	001124	BIS	#ECCW, \$GDDAT	
6505	050206	023737	001124	001126	12\$:	CMP	\$GDDAT, \$BDDAT ;CHECK IF MRI CORRECT
6506	050214	001404			BEQ	13\$;YES, CONTINUE
6507	050216	012737	067661	001322	MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
6508	050224	104003			ERROR	3	
6509	050226	016237	000032	001126	13\$:	MOV	RKECPT(R2), \$BDDAT ;STORE ECC PATTERN REG.
6510	050234	001513			BEQ	19\$;CHECK IF ZERO
6511	050236	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6512	050242	012737	067737	001322	MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
6513	050250	104003			ERROR	3	

M11

```

6514 050252 016237 000030 001126 14$: MOV RKECP5(R2), $BDDAT ;STORE REC POSITION REC
6515 050260 022737 004066 001126 CMP #4066, $BDDAT ;CHECK IF POSITION CORRECT
6516 050266 001407 BEQ 15$ ;YES, CONTINUE
6517 050270 012737 004066 001124 MOV #4066, $GDDAT ;LOAD EXPECTED CONTENTS
6518 050276 012737 067712 001322 MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
6519 050304 104003 ERROR 3
6520 050306 016237 000024 001126 15$: MOV RKDB(R2), $BDDAT ;STORE DATA BUFFER
6521 050314 022737 177777 001126 CMP #177777, $BDDAT ;CHECK CONTENTS CORRECT
6522 050322 001407 BEQ 16$ ;YES, CHECK TO MAKE SURE
6523 ; DATA LATE RESET AND
6524 ; CONTROLLER ERROR RESET
6525 050324 012737 177777 001124 MOV #177777, $GDDAT ;LOAD EXPECTED CONTENTS
6526 050332 012737 067054 001322 MOV #EM1008, EM3N+2 ;LOAD ERROR MESSAGE
6527 050340 104003 ERROR 3
6528 050342 016237 000000 001700 16$: MOV RKCS1(R2), T.CS1 ;STORE COMMAND AND STATUS REG. 1
6529 050350 016237 000010 001710 MOV RKCS2(R2), T.CS2 ;STORE COMMAND AND STATUS REG. 2
6530 050356 012737 000200 001740 MOV #RDY, E.CS1 ;LOAD EXPECTED CS1
6531 050364 012737 000100 001750 MOV #IR, E.CS2 ;LOAD EXPECTED CS2
6532 050372 023737 001740 001700 CMP E.CS1, T.CS1 ;CHECK IF CS1 CORRECT
6533 050400 001401 BEQ 17$ ;YES CHECK OUTPUT READY RESET
6534 050402 104022 ERROR 22 ;ATTEMPTING TO READ SILO CONTAINING
6535 ; ONE WORD - CS1 INCORRECT
6536 050404 023737 001750 001710 17$: CMP E.CS2, T.CS2 ;CHECK IN OUTPUT READY RESET
6537 050412 001401 BEQ 18$ ;YES, READ AN EXTRA WORD
6538 050414 104023 ERROR 23 ;ATTEMPTING TO READ SILO COMMAND
6539 ; ONE WORD - CS2 INCORRECT
6540 050416 005762 000024 18$: TST RKDB(R2) ;READ DATA BUFFER
6541 050422 016237 000000 001700 MOV RKCS1(R2), T.CS1 ;STORE COMMAND AND STATUS REG. 7
6542 050430 016237 000010 001710 MOV RKCS2(R2), T.CS2 ;STORE COMMAND AND STATUS REG. 2
6543 050436 012737 100200 001740 MOV #CERR!RDY, E.CS1 ;LOAD EXPECTED CS1
6544 050444 012737 100100 001750 MOV #DLT!IR, E.CS2 ;LOAD EXPECTED CS2
6545 050452 023737 001750 001710 CMP E.CS2, T.CS2 ;CHECK IF DATA LATE SET
6546 050460 001401 BEQ 19$ ;YES, CHECK IF CONTROLLER ERROR
6547 ; SET
6548 050462 104016 ERROR 16 ;ATTEMPTING TO READ SILO WHEN
6549 ; EMPTY - CS2 INCORRECT
6550 050464 023737 001740 001700 19$: CMP E.CS1, T.CS1 ;CHECK IF CONTROLLER ERROR SET
6551 050472 001401 BEQ 20$ ;YES, CLEAR CONTROLLER
6552 050474 104017 ERROR 17 ;ATTEMPTING TO READ SILO WHEN
6553 ; EMPTY - CS1 INCORRECT
6554 050476 012762 100000 000000 20$: MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
    
```

```

6555
6556 *****
6557 *TEST 66 ONE WORD SILO WRITE AND REG. INTERACTION (PART 1)
6558 *
6559 * ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
6560 * ZERO.
6561 *
6562 * WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERACTION
6563 * PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND
6564 * STATUS REGISTER 2. IF NOT WAIT.
6565 *
6566 * IF OUTPUT READY SETS READ BACK CONTENTS, AND CHECK
6567 * IF CORRECT.
6568 *
6569 * THE FOLLOWING CONFIGURATIONS ARE USED:
    
```

N11

```

6570          : *
6571          : *      000000  000010  000200  004000  100000
6572          : *      000001  000020  000400  010000
6573          : *      000002  000040  001000  020000
6574          : *      000004  000100  002000  040000
6575          : *
6576          : *****
6577 050504 000004          TS66: SCOPE
6578 050506 012737 000764 001200      MOV      #500, $TIMES      ;; DO 500. ITERATIONS
6579 050514 012737 065575 001320      MOV      #EM33, EM3N      ;; LOAD ERROR MESSAGE FOR PRINT OUT
6580 050522 012737 000001 002010      MOV      #000001, CONFIG ;; LOAD INITIAL CONFIGURATION
6581 050530 012701 000021          MOV      #17, R1         ;; LOAD CONFIGURATION COUNT
6582 050534 012737 050542 001110      MOV      #15, $LPERR     ;; LOAD LOOP ON ERROR LOCATION FOR
6583                                     ;; SUBTEST LOOP
6584
6585 050542          15:
6586 050542 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ; CLEAR RK611 CONTROLLER
6587 050550 005062 000002          CLR      RKWC(R2)        ; CLEAR WORD COUNT REG.
6588 050554 013762 002010 000024      MOV      CONFIG, RKDB(R2) ; WRITE DATA BUFFER
6589 050562 016237 000000 001126      MOV      RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
6590 050570 022737 000200 001126      CMP      #RDY, $BDDAT    ; CHECK IF CS1 CORRECT
6591 050576 001407          BEQ      2$              ; YES, CHECK OTHER REGISTERS
6592 050600 012737 000200 001124      MOV      #RDY, $GDDAT    ; LOAD EXPECTED CONTENTS
6593 050606 012737 067324 001322      MOV      #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
6594 050614 104003          ERROR    3
6595 050616 016237 000004 001126      MOV      RKBA(R2), $BDDAT ; STORE BUS ADDRESS
6596 050624 001406          BEQ      3$              ; CHECK IF ZERO
6597 050626 005037 001124          CLR      $GDDAT         ; LOAD EXPECTED CONTENTS
6598 050632 012737 067375 001322      MOV      #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
6599 050640 104003          ERROR    3
6600 050642 016237 000002 001126      MOV      RKWC(R2), $BDDAT ; STORE WORD COUNT REG.
6601 050650 001406          BEQ      4$              ; CHECK IF ZERO
6602 050652 005037 001124          CLR      $GDDAT         ; LOAD EXPECTED CONTENTS
6603 050656 012737 067345 001322      MOV      #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
6604 050664 104003          ERROR    3
6605 050666 016237 000006 001126      MOV      RKDA(R2), $BDDAT ; STORE DISK ADDRESS REG
6606 050674 001406          BEQ      5$              ; CHECK IF ZERO
6607 050676 005037 001124          CLR      $GDDAT         ; LOAD EXPECTED CONTENTS
6608 050702 012737 067422 001322      MOV      #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
6609 050710 104003          ERROR    3
6610 050712 016237 000016 001126      MOV      RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY AND OFFSET
6611 050720 001406          BEQ      6$              ; CHECK IF ZERO
6612 050722 005037 001124          CLR      $GDDAT         ; LOAD EXPECTED CONTENTS
6613 050726 012737 067566 001322      MOV      #EM1024, EM3N+2 ; LOAD ERROR MESSAGE
6614 050734 104003          ERROR    3
6615 050736 012700 000005          6$: MOV      #5, R0          ; LOAD COUNTER TO WAIT FOR
6616 050742 005300          7$: DEC      R0          ; OUTPUT READY
6617 050744 001376          BNE     7$
6618 050746 016237 000010 001126      MOV      RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG 2
6619 050754 022737 000300 001126      CMP      #IR!OR, $BDDAT  ; CHECK IF CS2 CORRECT
6620 050762 001407          BEQ      8$              ; YES, CONTINUE TEST
6621 050764 012737 000300 001124      MOV      #IR!OR, $GDDAT  ; LOAD EXPECTED CONTENTS
6622 050772 012737 066714 001322      MOV      #EM1004, EM3N+2 ; LOAD ERROR MESSAGE
6623 051000 104003          ERROR    3
6624 051002 016237 000012 001126      MOV      RKDS(R2), $BDDAT ; STORE PRIVE STATUS REG
6625 051010 001406          BEQ      9$              ; CHECK IF ZERO
    
```

```

6626 051012 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
6627 051016 012737 067501 001322 MOV #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
6628 051024 104003 ERROR 3
6629 051026 016237 000014 001126 9$: MOV RKR(R2), $BDDAT ;STORE ERROR REGISTER
6630 051034 001406 BEQ 10$ ;CHECK IF ZERO
6631 051036 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
6632 051042 012737 067537 001322 MOV #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
6633 051050 104003 ERROR 3
6634 051052 016237 000020 001126 10$: MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADDRESS REG
6635 051060 001406 BEQ 12$ ;CHECK IF ZERO
6636 051062 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
6637 051066 012737 067634 001322 MOV #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
6638 051074 104003 ERROR 3
6639 051076 016237 000026 001126 12$: MOV RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG. 1
6640 051104 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED MRI
6641 051112 032737 020000 001126 BIT #ECCW, $BDDAT
6642 051120 001403 BEQ 13$
6643 051122 052737 020000 001124 BIS #ECCW, $GDDAT
6644 051130 023737 001124 001126 13$: CMP $GDDAT, $BDDAT ;CHECK IF MRI CORRECT
6645 051136 001404 BEQ 14$ ;YES, CONTINUE
6646 051140 012737 067661 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
6647 051146 104003 ERROR 3
6648 051150 016237 000032 001126 14$: MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
6649 051156 001406 BEQ 15$ ;CHECK IF ZERO
6650 051160 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
6651 051164 012737 067737 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
6652 051172 104003 ERROR 3
6653 051174 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
6654 051202 022737 004066 001126 CMP #4066, $BDDAT ;CHECK IF ECC POSITION REG CORRECT
6655 051210 001407 BEQ 16$ ;YES, CHECK DATA BUFFER
6656 051212 012737 004066 001124 MOV #4066, $GDDAT ;LOAD EXPECTED CONTENTS
6657 051220 012737 067712 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
6658 051226 104003 ERROR 3
6659 051230 016237 000024 001126 16$: MOV RKDB(R2), $BDDAT ;STORE DATA BUFFER
6660 051236 023737 002010 001126 CMP CONFIG, $BDDAT ;CHECK FOR CORRECT CONTENTS
6661 051244 001407 BEQ 17$ ;YES, CHECK IF FINISHED
6662 051246 013737 002010 001124 MOV CONFIG, $GDDAT ;LOAD EXPECTED CONTENTS
6663 051254 012737 067054 001322 MOV #EM1008,EM3N+2 ;LOAD ERROR MESSAGE
6664 051262 104003 ERROR 3
6665 051264 104415 17$: SCOP1 ;CHECK IF LOOP ON ERROR
6666 051266 000241 CLC ;SHIFT IN ZERO
6667 051270 006137 002010 ROL CONFIG
6668 051274 005301 DEC R1 ;CHECK IF FINISHED
6669 051276 001402 BEQ TST67 ;;YES, GO ON TO NEXT TEST
6670 051300 000137 050542 JMP 1$

```

```

6671
6672 *****
6673 *TEST 67 ONE WORD SILO WRITE AND REG. INTERACTION (PART 2)
6674 *
6675 *
6676 * ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
6677 * ZERO.
6678 *
6679 * WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERACTION
6680 * PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND
6681 * STATUS REGISTER 2. IF NOT WAIT.

```



```

6682      *      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
6683      *      IF CORRECT.
6684      *
6685      *      THE FOLLOWING CONFIGURATIONS ARE USED:
6686      *
6687      *      177777 177767 177577 173777 077777
6688      *      177776 177757 177377 167777
6689      *      177775 177737 176777 157777
6690      *      177773 177677 175777 137777
6691      *

```

```

*****
TST67:

```

```

051304 000004          SCOPE
051306 012737 000764 001200  MOV      #500, $TIMES      ; DO 500. ITERATIONS
051314 012737 065575 001320  MOV      #EM33, EM3N    ; LOAD ERROR MESSAGE FOR PRINT OUT
051322 012737 177776 002010  MOV      #177776, CONFIG ; LOAD INITIAL CONFIGURATION
051330 012701 000021          MOV      #17, R1       ; LOAD CONFIGURATION COUNT
051334 012737 051342 001110  MOV      #15, $LPERR   ; LOAD LOOP ON ERROR LOCATION FOR
                          ; SUBTEST LOOP

```

```

15:

```

```

051342 012762 100000 000000  MOV      #CLR, RKCS1(R2) ; CLEAR RK611 CONTROLLER
051350 005062 000002          CLR      RKWC(R2)      ; CLEAR WORD COUNT REG.
051354 013762 002010 000024  MOV      CONFIG, RKDB(R2) ; WRITE DATA BUFFER
051362 016237 000000 001126  MOV      RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG. 1
051370 022737 000200 001126  CMP      #RDY, $BDDAT   ; CHECK IF CS1 CORRECT
051376 001407          BEQ      25             ; YES, CHECK OTHER REGISTERS
051400 012737 000200 001124  MOV      #RDY, $GDDAT   ; LOAD EXPECTED CONTENTS
051406 012737 067324 001322  MOV      #EM1017, EM3N+2 ; LOAD ERROR MESSAGE

```

```

051414 104003          ERROR      3
051416 016237 000004 001126 25:  MOV      RKBA(R2), $BDDAT ; STORE BUS ADDRESS
051424 001406          BEQ      35             ; CHECK IF ZERO
051426 005037 001124          CLR      $GDDAT       ; LOAD EXPECTED CONTENTS
051432 012737 067375 001322  MOV      #EM1019, EM3N+2 ; LOAD ERROR MESSAGE

```

```

051440 104003          ERROR      3
051442 016237 000002 001126 35:  MOV      RKWC(R2), $BDDAT ; STORE WORD COUNT REG.
051450 001406          BEQ      45             ; CHECK IF ZERO
051452 005037 001124          CLR      $GDDAT       ; LOAD EXPECTED CONTENTS
051456 012737 067345 001322  MOV      #EM1018, EM3N+2 ; LOAD ERROR MESSAGE

```

```

051464 104003          ERROR      3
051466 016237 000006 001126 45:  MOV      RKDA(R2), $BDDAT ; STORE DISK ADDRESS REG
051474 001406          BEQ      55             ; CHECK IF ZERO
051476 005037 001124          CLR      $GDDAT       ; LOAD EXPECTED CONTENTS
051502 012737 067422 001322  MOV      #EM1020, EM3N+2 ; LOAD ERROR MESSAGE

```

```

051510 104003          ERROR      3
051512 016237 000016 001126 55:  MOV      RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY AND OFFSET
051520 001406          BEQ      65             ; CHECK IF ZERO
051522 005037 001124          CLR      $GDDAT       ; LOAD EXPECTED CONTENTS
051526 012737 067566 001322  MOV      #EM1024, EM3N+2 ; LOAD ERROR MESSAGE

```

```

051534 104003          ERROR      3
051536 012700 000005 65:  MOV      #5, R0        ; LOAD COUNTER TO WAIT FOR
051542 005300 75:  DEC      R0           ; OUTPUT READY
051544 001376          BNE      75

```

```

051546 016237 000010 001126  MOV      RKCS2(R2), $BDDAT ; STORE COMMAND AND STATUS REG 2
051554 022737 000300 001126  CMP      #IR!OR, $BDDAT  ; CHECK IF CS2 CORRECT
051562 001407          BEQ      85             ; YES, CONTINUE TEST
051564 012737 000300 001124  MOV      #IR!OR, $GDDAT  ; LOAD EXPECTED CONTENTS

```

```

6738 051572 012737 066714 001322      MOV      #EM1004,EM3N+2 ;LOAD ERROR MESSAGE
6739 051600 104003      ERROR   3
6740 051602 016237 000012 001126 8$:  MOV      RKDS(R2), $BDDAT ;STORE PRIVE STATUS REG
6741 051610 001406      BEQ     9$ ;CHECK IF ZERO
6742 051612 005037 001124      CLR     $GDDAT ;LOAD EXPECTED CONTENTS
6743 051616 012737 067501 001322      MOV      #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
6744 051624 104003      ERROR   3
6745 051626 016237 000014 001126 9$:  MOV      RPER(R2), $BDDAT ;STORE ERROR REGISTER
6746 051634 001406      BEQ     10$ ;CHECK IF ZERO
6747 051636 005037 001124      CLR     $GDDAT ;LOAD EXPECTED CONTENTS
6748 051642 012737 067537 001322      MOV      #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
6749 051650 104003      ERROR   3
6750 051652 016237 000020 001126 10$: MOV      RKDCYL(R2), $BDDAT ;STORE CYLINDER ADDRESS REG
6751 051660 001406      BEQ     12$ ;CHECK IF ZERO
6752 051662 005037 001124      CLR     $GDDAT ;LOAD EXPECTED CONTENTS
6753 051666 012737 067634 001322      MOV      #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
6754 051674 104003      ERROR   3
6755 051676 016237 000026 001126 12$: MOV      PKMRI(R2), $BDDAT ;STORE MAINTENANCE REG. 1
6756 051704 012737 002000 001124      MOV      #EM1026,$GDDAT ;LOAD EXPECTED MRI
6757 051712 032737 020000 001126      BIT     #ECCW,$BDDAT
6758 051720 001403      BEQ     13$
6759 051722 052737 020000 001124      BIS     #ECCW,$GDDAT
6760 051730 023737 001124 001126 13$:  CMP     $GDDAT,$BDDAT ;CHECK IF MRI CORRECT
6761 051736 001404      BEQ     14$ ;YES, CONTINUE
6762 051740 012737 067661 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
6763 051746 104003      ERROR   3
6764 051750 016237 000032 001126 14$: MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
6765 051756 001406      BEQ     15$ ;CHECK IF ZERO
6766 051760 005037 001124      CLR     $GDDAT ;LOAD EXPECTED CONTENTS
6767 051764 012737 067737 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
6768 051772 104003      ERROR   3
6769 051774 016237 000030 001126 15$: MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
6770 052002 022737 004066 001126      CMP     #4066,$BDDAT ;CHECK IF ECC POSITION REG CORRECT
6771 052010 001407      BEQ     16$ ;YES, CHECK DATA BUFFER
6772 052012 012737 004066 001124      MOV      #4066,$GDDAT ;LOAD EXPECTED CONTENTS
6773 052020 012737 067712 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
6774 052026 104003      ERROR   3
6775 052030 016237 000024 001126 16$: MOV      RKDB(R2), $BDDAT ;STORE DATA BUFFER
6776 052036 023737 002010 001126      CMP     CONFIG,$BDDAT ;CHECK FOR CORRECT CONTENTS
6777 052044 001407      BEQ     17$ ;YES, CHECK IF FINISHED
6778 052046 013737 002010 001124      MOV      CONFIG,$GDDAT ;LOAD EXPECTED CONTENTS
6779 052054 012737 067054 001322      MOV      #EM1008,EM3N+2 ;LOAD ERROR MESSAGE
6780 052062 104003      ERROR   3
6781 052064 104415      SCOPI  ;CHECK IF LOOP ON ERROR
6782 052066 000261      SEC     ;SHIFT IN ONE
6783 052070 006137 002010      ROL     CONFIG
6784 052074 005301      DEC     R1 ;CHECK IF FINISHED
6785 052076 001402      BEQ     TST70 ;;YES, GO ON TO NEXT TEST
6786 052100 000137 051342      JMP     1$

```

```

6787
6788
6789 *****
6790 *TEST 70 ONE WORD SILO WRITE AND REG. INTERACTION (PART 3)
6791 *
6792 * ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
6793 * ZERO.

```

E12

6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809 052104 000004
6810 052106 012737 000764 001200
6811 052114 012737 065575 001320
6812 052122 005037 002010
6813 052126 012701 000021
6814 052132 012737 052140 001110
6815
6816
6817 052140
6818 052140 012762 100000 000000
6819 052146 005062 000002
6820 052152 013762 002010 000024
6821 052160 016237 000000 001126
6822 052166 022737 000200 001126
6823 052174 001407
6824 052176 012737 000200 001124
6825 052204 012737 067324 001322
6826 052212 104003
6827 052214 016237 000004 001126 2\$:
6828 052222 001406
6829 052224 005037 001124
6830 052230 012737 067375 001322
6831 052236 104003
6832 052240 016237 000002 001126 3\$:
6833 052246 001406
6834 052250 005037 001124
6835 052254 012737 067345 001322
6836 052262 104003
6837 052264 016237 000006 001126 4\$:
6838 052272 001406
6839 052274 005037 001124
6840 052300 012737 067422 001322
6841 052306 104003
6842 052310 016237 000016 001126 5\$:
6843 052316 001406
6844 052320 005037 001124
6845 052324 012737 067566 001322
6846 052332 104003
6847 052334 012700 000005 6\$:
6848 052340 005300 7\$:
6849 052342 001376

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

T\$T70: SCOPE
MOV #500, \$TIMES ; DO 500. ITERATIONS
MOV #EM33, EM3N ; LOAD ERROR MESSAGE FOR PRINT OUT
CLR CONFIG ; LOAD INITIAL CONFIGURATION
MOV #17, R1 ; LOAD CONFIGURATION COUNT
MOV #1\$, \$LPERR ; LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1\$:
MOV #CCLR, RKCS1(R2) ; CLEAR RK611 CONTROLLER
CLR RKWC(R2) ; CLEAR WORD COUNT REG.
MOV CONFIG, RKDB(R2) ; WRITE DATA BUFFER
MOV RKCS1(R2), \$BODAT ; STORE COMMAND AND STATUS REG. 1
CMP #RDY, \$BODAT ; CHECK IF CSI CORRECT
BEQ 2\$; YES, CHECK OTHER REGISTERS
MOV #RDY, \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
2\$:
MOV RKBA(R2), \$BODAT ; STORE BUS ADDRESS
BEQ 3\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
3\$:
MOV RKWC(R2), \$BODAT ; STORE WORD COUNT REG.
BEQ 4\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
4\$:
MOV RKDA(R2), \$BODAT ; STORE DISK ADDRESS REG
BEQ 5\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
5\$:
MOV RKASOF(R2), \$BODAT ; STORE ATTENTION SUMMARY AND OFFSET
BEQ 6\$; CHECK IF ZERO
CLR \$GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1024, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
6\$:
MOV #5, R0 ; LOAD COUNTER TO WAIT FOR
7\$:
DEC R0 ; OUTPUT READY
BNE 7\$

F12

6850	052344	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG 2
6851	052352	022737	000300	001126		CMP	#IR!OR, \$BDDAT	; CHECK IF CS2 CORRECT
6852	052360	001407				BEQ	8\$; YES, CONTINUE TEST
6853	052362	012737	000300	001124		MOV	#IR!OR, \$GDDAT	; LOAD EXPECTED CONTENTS
6854	052370	012737	066714	001322		MOV	#EM1004, EM3N+2	; LOAD ERROR MESSAGE
6855	052376	104003				ERROR	3	
6856	052400	016237	000012	001126	8\$:	MOV	RKDS(R2), \$BDDAT	; STORE PRIVE STATUS REG
6857	052406	001406				BEQ	9\$; CHECK IF ZERO
6858	052410	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
6859	052414	012737	067501	001322		MOV	#EM1022, EM3N+2	; LOAD ERROR MESSAGE
6860	052422	104003				ERROR	3	
6861	052424	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	; STORE ERROR REGISTER
6862	052432	001406				BEQ	10\$; CHECK IF ZERO
6863	052434	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
6864	052440	012737	067537	001322		MOV	#EM1023, EM3N+2	; LOAD ERROR MESSAGE
6865	052446	104003				ERROR	3	
6866	052450	016237	000020	001126	10\$:	MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADDRESS REG
6867	052456	001406				BEQ	12\$; CHECK IF ZERO
6868	052460	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
6869	052464	012737	067634	001322		MOV	#EM1025, EM3N+2	; LOAD ERROR MESSAGE
6870	052472	104003				ERROR	3	
6871	052474	016237	000026	001126	12\$:	MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG. 1
6872	052502	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MR1
6873	052510	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6874	052516	001403				BEQ	13\$	
6875	052520	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6876	052526	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT
6877	052534	001404				BEQ	14\$; YES, CONTINUE
6878	052536	012737	067661	001322		MOV	#EM1026, EM3N+2	; LOAD ERROR MESSAGE
6879	052544	104003				ERROR	3	
6880	052546	06237	000032	001126	14\$:	MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
6881	052554	001406				BEQ	15\$; CHECK IF ZERO
6882	052556	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
6883	052562	012737	067737	001322		MOV	#EM1030, EM3N+2	; LOAD ERROR MESSAGE
6884	052570	104003				ERROR	3	
6885	052572	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG.
6886	052600	022737	004066	001126		CMP	#4066, \$BDDAT	; CHECK IF ECC POSITION REG CORRECT
6887	052606	001407				BEQ	16\$; YES, CHECK DATA BUFFER
6888	052610	012737	004066	001124		MOV	#4066, \$GDDAT	; LOAD EXPECTED CONTENTS
6889	052616	012737	067712	001322		MOV	#EM1029, EM3N+2	; LOAD ERROR MESSAGE
6890	052624	104003				ERROR	3	
6891	052626	016237	000024	001126	16\$:	MOV	RKDB(R2), \$BDDAT	; STORE DATA BUFFER
6892	052634	023737	002010	001126		CMP	CONFIG, \$BDDAT	; CHECK FOR CORRECT CONTENTS
6893	052642	001407				BEQ	17\$; YES, CHECK IF FINISHED
6894	052644	013737	002010	001124		MOV	CONFIG, \$GDDAT	; LOAD EXPECTED CONTENTS
6895	052652	012737	067054	001322		MOV	#EM1008, EM3N+2	; LOAD ERROR MESSAGE
6896	052660	104003				ERROR	3	
6897	052662	104415			17\$:	SCOPI		; CHECK IF LOOP ON ERROR
6898	052664	000261				SEC		; SHIFT IN ONE
6899	052666	006137	002010			ROL	CONFIG	
6900	052672	005301				DEC	R1	; CHECK IF FINISHED
6901	052674	001402				BEQ	TST71	; YES, GO ON TO NEXT TEST
6902	052676	000137	052140			JMP	1\$	

6903
 6904
 6905
 ;*****
 ;*TEST 71 ONE WORD SILO WRITE AND REG. INTERACTION (PART 4)

```

6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925 052702 000004
6926 052704 012737 000764 001200
6927 052712 012737 065575 001320
6928 052720 005037 002010
6929 052724 012701 000021
6930 052730 012737 052736 001110
6931
6932
6933 052736
6934 052736 012762 100000 000000
6935 052744 005062 000002
6936 052750 013762 002010 000024
6937 052756 016237 000000 001126
6938 052764 022737 000200 001126
6939 052772 001407
6940 052774 012737 000200 001124
6941 053002 012737 067324 001322
6942 053010 104003
6943 053012 016237 000004 001126
6944 053020 001406
6945 053022 005037 001124
6946 053026 012737 067375 001322
6947 053034 104003
6948 053036 016237 000002 001126
6949 053044 001406
6950 053046 005037 001124
6951 053052 012737 067345 001322
6952 053060 104003
6953 053062 016237 000006 001126
6954 053070 001406
6955 053072 005037 001124
6956 053076 012737 067422 001322
6957 053104 104003
6958 053106 016237 000016 001126
6959 053114 001406
6960 053116 005037 001124
6961 053122 012737 067566 001322

```

```

*****
:
: 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
:
:*****
†ST71: SCOPE
MOV #500, $TIMES ; DO 500. ITERATIONS
MOV #EM33, EM3N ; LOAD ERROR MESSAGE FOR PRINT OUT
CLR CONFIG ; LOAD INITIAL CONFIGURATION
MOV #17, R1 ; LOAD CONFIGURATION COUNT
MOV #15, $LPERR ; LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

15:
MOV #CCLR, RKCS1(R2) ; CLEAR RK611 CONTROLLER
CLR RKWC(R2) ; CLEAR WORD COUNT REG.
MOV CONFIG, RKDB(R2) ; WRITE DATA BUFFER
MOV RKCS1(R2), $BDDAT ; STORE COMMAND AND STATUS REG.1
CMP #RDY, $BDDAT ; CHECK IF CS1 CORRECT
BEQ 2$ ; YES, CHECK OTHER REGISTERS
MOV #RDY, $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1017, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
2$:
MOV RKBA(R2), $BDDAT ; STORE BUS ADDRESS
BEQ 3$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1019, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
3$:
MOV RKWC(R2), $BDDAT ; STORE WORD COUNT REG.
BEQ 4$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1018, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
4$:
MOV RKDA(R2), $BDDAT ; STORE DISK ADDRESS REG
BEQ 5$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1020, EM3N+2 ; LOAD ERROR MESSAGE
ERROR 3
5$:
MOV RKASOF(R2), $BDDAT ; STORE ATTENTION SUMMARY AND OFFSET
BEQ 6$ ; CHECK IF ZERO
CLR $GDDAT ; LOAD EXPECTED CONTENTS
MOV #EM1024, EM3N+2 ; LOAD ERROR MESSAGE

```

H12

6962	053130	104003				ERROR	3	
6963	053132	012700	000005		6\$:	MOV	#5,RO	;LOAD COUNTER TO WAIT FOR
6964	053136	005300			7\$:	DEC	RO	; OUTPUT READY
6965	053140	001376				BNE	7\$	
6966	053142	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG 2
6967	053150	022737	000300	001126		CMP	#IR!OR, \$BDDAT	;CHECK IF CS2 CORRECT
6968	053156	001407				REQ	8\$;YES, CONTINUE TEST
6969	053160	012737	000300	001124		MOV	#IR!OR, \$GDDAT	;LOAD EXPECTED CONTENTS
6970	053166	012737	066714	001322		MOV	#EM1004, EM3N+2	;LOAD ERROR MESSAGE
6971	053174	104003				ERROR	3	
6972	053176	016237	000012	001126	8\$:	MOV	RKDS(R2), \$BDDAT	;STORE PRIVE STATUS REG
6973	053204	001406				BEQ	9\$;CHECK IF ZERO
6974	053206	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6975	053212	012737	067501	001322		MOV	#EM1022, EM3N+2	;LOAD ERROR MESSAGE
6976	053220	104003				ERROR	3	
6977	053222	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REGISTER
6978	053230	001406				BEQ	10\$;CHECK IF ZERO
6979	053232	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6980	053236	012737	067537	001322		MOV	#EM1023, EM3N+2	;LOAD ERROR MESSAGE
6981	053244	104003				ERROR	3	
6982	053246	016237	000020	001126	10\$:	MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADDRESS REG
6983	053254	001406				BEQ	12\$;CHECK IF ZERO
6984	053256	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6985	053262	012737	067634	001322		MOV	#EM1025, EM3N+2	;LOAD ERROR MESSAGE
6986	053270	104003				ERROR	3	
6987	053272	016237	000026	001126	12\$:	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG. 1
6988	053300	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MR1
6989	053306	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6990	053314	001403				BEQ	13\$	
6991	053316	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6992	053324	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
6993	053332	001404				BEQ	14\$;YES, CONTINUE
6994	053334	012737	067661	001322		MOV	#EM1026, EM3N+2	;LOAD ERROR MESSAGE
6995	053342	104003				ERROR	3	
6996	053344	016237	000032	001126	14\$:	MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
6997	053352	001406				BEQ	15\$;CHECK IF ZERO
6998	053354	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6999	053360	012737	067737	001322		MOV	#EM1030, EM3N+2	;LOAD ERROR MESSAGE
7000	053366	104003				ERROR	3	
7001	053370	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
7002	053376	022737	004066	001126		CMP	#4066, \$BDDAT	;CHECK IF ECC POSITION REG CORRECT
7003	053404	001407				BEQ	16\$;YES, CHECK DATA BUFFER
7004	053406	012737	004066	001124		MOV	#4066, \$GDDAT	;LOAD EXPECTED CONTENTS
7005	053414	012737	067712	001322		MOV	#EM1029, EM3N+2	;LOAD ERROR MESSAGE
7006	053422	104003				ERROR	3	
7007	053424	016237	000024	001126	16\$:	MOV	RKDB(R2), \$BDDAT	;STORE DATA BUFFER
7008	053432	023737	002010	001126		CMP	CONFIG, \$BDDAT	;CHECK FOR CORRECT CONTENTS
7009	053440	001407				BEQ	17\$;YES, CHECK IF FINISHED
7010	053442	013737	002010	001124		MOV	CONFIG, \$GDDAT	;LOAD EXPECTED CONTENTS
7011	053450	012737	067054	001322		MOV	#EM1008, EM3N+2	;LOAD ERROR MESSAGE
7012	053456	104003				ERROR	3	
7013	053460	104415			17\$:	SCOPI		;CHECK IF LOOP ON ERROR
7014	053462	000261				SEC		;SHIFT IN ONE
7015	053464	006037	002010			ROR	CONFIG	
7016	053470	005301				DEC	R1	;CHECK IF FINISHED
7017	053472	001402				BEQ	TST72	;YES, GO ON TO NEXT TEST


```

7018 053474 000137 052736          JMP      1$
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031 053500 000004
7032 053502 012737 000764 001200
7033 053510 013702 001270
7034 053514 012762 100000 000000
7035 053522 012703 070764
7036 053526 005037 002022
7037 053532 012362 000024
7038 053536 012700 000005
7039 053542 005300          1$:
7040 053544 001376          DEC      R0
7041 053546 016237 000000 001700      BNE     1$
7042 053554 016237 000010 001710      MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
7043 053562 012737 000200 001740      MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
7044 053570 012737 000300 001750      MOV     #RDY,E.CS1 ;LOAD EXPECTED COMMAND AND STATUS REG. 1
7045 053576 023737 001740 001700      MOV     #IR!OR,E.CS2 ;LOAD EXPECTED COMMAND AND STATUS REG 2
7046 053604 001401          CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
7047 053606 104024          BEQ     2$ ;YES, CONTINUE
7048 053610 023737 001750 001710      ERROR  24 ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT
7049 053616 001401          CMP     E.CS2,T.CS2 ;CHECK IF CS2 CORRECT
7050 053620 104025          BEQ     3$ ;YES, CONTINUE
7051 053622 005237 002022          ERROR  25 ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
7052 053626 012362 000024          INC     SILCNT ;INCREMENT SILO COUNT
7053 053632 016237 000000 001700      MOV     (R3)+,RKDB(R2) ;LOAD SILO
7054 053640 016237 000010 001710      MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7055 053646 023737 001740 001700      MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
7056 053654 001401          CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
7057 053656 104024          BEQ     4$ ;YES, CONTINUE
7058 053660 023737 001750 001710      ERROR  24 ;ATTEMPTING TO WRITE SILO- CS1 INCORRECT
7059 053666 001401          CMP     E.CS2,T.CS2 ;CHECK IF CS2 CORRECT
7060 053670 104025          BEQ     5$ ;YES, CONTINUE
7061 053672 022737 000100 002022      ERROR  25 ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
7062 053700 001350          CMP     #64.,SILCNT ;CHECK IF ALL EXCEPT LAST WORD WRITTEN
7063 053702 005237 002022          BNE     3$ ;YES, GO WRITE NEXT WORD
7064 053706 011362 000024          INC     SILCNT ;INCREMENT SILO COUNT
7065 053712 016237 000000 001700      MOV     (R3),RKDB(R2) ;LOAD SILO
7066 053720 016237 000010 001710      MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7067 053726 012737 000200 001750      MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
7068 053734 023737 001740 001700      MOV     #OR,E.CS2 ;LOAD EXPECTED CS2
7069 053742 001401          CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
7070 053744 104024          BEQ     6$ ;YES, CONTINUE
7071 053746 023737 001750 001710      ERROR  24 ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT
7072 053754 001401          CMP     E.CS2,T.CS2 ;CHECK IF CS2 CORRECT
7073 053756 104025          BEQ     7$ ;YES, UNLOAD SILO
          ERROR  25 ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT

```

```

*****
*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 CHECKING
* CONTENTS, INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH
* WORD READ. AN EXTRA READ IS THEN DONE TO MAKE SURE THE
* SILO IS EMPTY.
*
*****

```

```

TST72: SCOPE
MOV     #500.,$TIMES ;DO 500. ITERATIONS
MOV     $BASE,R2 ;LOAD RK611 BASE
MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
MOV     #SILO,R3 ;LOAD ADDRESS OF INPUT DATA
CLR     SILCNT ;CLEAR SILO COUNT
MOV     (R3)+,RKDB(R2) ;LOAD SILO
MOV     #5,R0 ;WAIT FOR OUTPUT READY
1$: DEC     R0
BNE     1$
MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
MOV     #RDY,E.CS1 ;LOAD EXPECTED COMMAND AND STATUS REG. 1
MOV     #IR!OR,E.CS2 ;LOAD EXPECTED COMMAND AND STATUS REG 2
CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
BEQ     2$ ;YES, CONTINUE
ERROR  24 ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT
2$: CMP     E.CS2,T.CS2 ;CHECK IF CS2 CORRECT
BEQ     3$ ;YES, CONTINUE
ERROR  25 ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
3$: INC     SILCNT ;INCREMENT SILO COUNT
MOV     (R3)+,RKDB(R2) ;LOAD SILO
MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
BEQ     4$ ;YES, CONTINUE
ERROR  24 ;ATTEMPTING TO WRITE SILO- CS1 INCORRECT
4$: CMP     E.CS2,T.CS2 ;CHECK IF CS2 CORRECT
BEQ     5$ ;YES, CONTINUE
ERROR  25 ;ATTEMPTING TO WRITE SILO-CS2 INCORRECT
5$: CMP     #64.,SILCNT ;CHECK IF ALL EXCEPT LAST WORD WRITTEN
BNE     3$ ;YES, GO WRITE NEXT WORD
INC     SILCNT ;INCREMENT SILO COUNT
MOV     (R3),RKDB(R2) ;LOAD SILO
MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
MOV     #OR,E.CS2 ;LOAD EXPECTED CS2
CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
BEQ     6$ ;YES, CONTINUE
ERROR  24 ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT
6$: CMP     E.CS2,T.CS2 ;CHECK IF CS2 CORRECT
BEQ     7$ ;YES, UNLOAD SILO
ERROR  25 ;ATTEMPTING TO WRITE SILO-CS1 INCORRECT

```



```

7074 053760 005037 002022          7$: CLR      SILCNT      ;CLEAR SILO COUNT
7075 053764 012703 070764          MOV      #SILO,R3      ;LOAD ADDRESS OF INPUT DATA
7076 053770 012737 000300 001750    MOV      #IR!OR,E.CS2  ;LOAD EXPECTED CS2
7077 053776 016237 000024 001126    MOV      RKDB(R2), $BDDAT ;STORE NEXT WORD ON SILO
7078 054004 012700 000005          MOV      #5,R0        ;WAIT FOR INPUT READY
7079 054010 005300          8$: DEC      R0
7080 054012 001376          SNE     8$
7081 054014 000403          BR      10$          ;CONTINUE WITH TEST
7082
7083 054016 016237 000024 001126    9$: MOV      RKDB(R2), $BDDAT ;STORE NEXT WORD ON SILO
7084 054024 016237 000000 001700    10$: MOV     RKCS1(R2), T.CS1 ;STORE COMMAND AND STATUS REG 1
7085 054032 016237 000010 001710    MOV     RKCS2(R2), T.CS2 ;STORE COMMAND AND STATUS REG 2
7086 054040 012337 001124          MOV     (R3)+, $GDDAT  ;LOAD EXPECTED DATA
7087 054044 023737 001740 001700    CMP     E.CS1, T.CS1  ;CHECK IF CONTROLLER ERROR RESET
7088 054052 001401          BEQ     11$          ;YES, CONTINUE
7089 054054 104026          ERROR  26          ;ATTEMPTING TO READ SILO-CS1 INCORRECT
7090 054056 023737 001750 001710    11$: CMP     E.CS2, T.CS2  ;CHECK IF OUTPUT READY AND INPUT READY SET
7091 054064 001401          BEQ     12$          ;YES, CHECK DATA
7092 054066 104027          ERROR  27          ;ATTEMPTING TO READ SILO-CS2 INCORRECT
7093 054070 023737 001124 001126    12$: CMP     $GDDAT, $BDDAT ;CHECK IF SILO CONTENTS CORRECT
7094 054076 001401          BEQ     13$          ;YES, CHECK IF LAST WORD
7095 054100 104030          ERROR  30          ;ATTEMPTING TO READ SILO-RKDB INCORRECT
7096 054102 005237 002022          13$: INC     SILCNT      ;SET UP FOR NEXT WORD
7097 054106 022737 000101 002022    CMP     #65., SILCNT  ;CHECK READY FOR LAST WORD
7098 054114 101340          BHI     9$          ;NO, READ NEXT WORD
7099 054116 103404          BLO     14$          ;CHECK IF SILO EMPTY
7100 054120 012737 000100 001750    MOV     #IR,E.CS2    ;LOAD EXPECTED CS2
7101 054126 000733          BR      9$          ;READ LAST WORD
7102
7103 054130 005762 000024          14$: TST     RKDB(R2)    ;READ DATA BUFFER
7104 054134 016237 000010 001710    MOV     RKCS2(R2), T.CS2 ;STORE COMMAND AND STATUS REG. 2
7105 054142 016237 000000 001700    MOV     RKCS1(R2), T.CS1 ;STORE COMMAND AND STATUS REG. 1
7106 054150 012737 100100 001750    MOV     #DLT!IR,E.CS2  ;LOAD EXPECTED CS2
7107 054156 012737 100200 001740    MOV     #CERR!RDY,E.CS1 ;LOAD EXPECTED CS1
7108 054164 023737 001750 001710    CMP     E.CS2, T.CS2  ;CHECK FOR DATA LATE SET
7109 054172 001401          BEQ     15$          ;YES, CHECK FOR CONTROLLER ERROR
7110 054174 104006          ERROR  6          ;CS2 INCORRECT AFTER LEAVING EMPTY
7111                                     SILO
7112 054176 023737 001740 001700    15$: CMP     E.CS1, T.CS1  ;CHECK FOR CONTROLLER ERROR
7113 054204 001401          BEQ     16$          ;YES, CLEAR DATA LATE
7114 054206 104017          ERROR  17          ;CS1 INCORRECT AFTER READING EMPTY
7115                                     SILO
7116 054210 012762 100000 000000    16$: MOV     #CLR,RKCS1(R2) ;CLEAR RK611
7117
7118                                     ;*****
7119                                     ;*TEST 73          SILO CAPACITY DATA LATE
7120                                     ;*
7121                                     ;*
7122                                     ;*   WRITE 67 WORDS IN THE SILO AND MAKE SURE DATA LATE ONLY
7123                                     ;*   OCCURS ON THE 67TH WORD.  CLEAR RK611 WITH CONTROLLER CLEAR.
7124                                     ;*   CHECK INPUT READY AND OUTPUT READY FOR INITIALIZED STATE.
7125                                     ;*****
7126 054216 000004          †ST73: SCOPE
7127 054220 012737 000764 001200    MOV     #500., $TIMES  ;;DO 500. ITERATIONS
7128 054226 013702 001270          MOV     $BASE,R2      ;LOAD RK611 BASE
7129 054232 012762 100000 000000    MOV     #CLR,RKCS1(R2) ;CLEAR RK611

```

K12

```

7130 054240 012700 000103          MOV      #67,R0          ;LOAD 67 WORDS INTO SILO CAUSING
7131 054244 012762 177777 000024 1$:  MOV      #177777,RKDB(R2) ; DATA LATE
7132 054252 005300          DEC      R0
7133 054254 001373          BNE     1$
7134 054256 016237 000010 001710      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
7135
7136 054264 016237 000000 001700      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
7137 054272 012737 100200 001750      MOV      #DLT!OR,E.CS2   ;LOAD EXPECTED CS2
7138 054300 012737 100200 001740      MOV      #CERR!RDY,E.CS1 ;LOAD EXPECTED CS1
7139 054306 023737 001750 001710      CMP      E.CS2,T.CS2    ;CHECK IF CS2 CORRECT (DLT)
7140 054314 001401          BEQ     2$              ;YES, CONTINUE
7141 054316 104031          ERROR   31              ;CS2 INCORRECT WHEN LOADING FULL SILO
7142 054320 023737 001740 001700 2$:  CMP      E.CS1,T.CS1    ;CHECK IF CS1 CORRECT (CERR)
7143 054326 001401          BEQ     3$              ;YES, CLEAR CONTROLLER
7144 054330 104032          ERROR   32              ;CS1 INCORRECT WHEN LOADING FULL SILO
7145 054332 012762 100000 000000 3$:  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7146 054340 016237 000000 001700      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
7147 054346 016237 000010 001710      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
7148 054354 012737 000200 001740      MOV      #RDY,E.CS1     ;LOAD EXPECTED CS1
7149 054362 012737 000100 001750      MOV      #IR,E.CS2      ;LOAD EXPECTED CS2
7150 054370 023737 001740 001700      CMP      E.CS1,T.CS1    ;CHECK IF CONTROLLER ERROR RESET
7151 054376 001401          BEQ     4$              ;YES, CHECK IF DATA LATE RESET
7152 054400 104020          ERROR   20              ;CS1 INCORRECT AFTER ATTEMPTING
7153                                     TO CLEAR DATA LATE
7154 054402 023737 001750 001710 4$:  CMP      E.CS2,T.CS2    ;CHECK IF DATA LATE RESET
7155 054410 001401          BEQ     TST74          ;YES GO ON TO NEXT TEST
7156 054412 104021          ERROR   21              ;CS2 INCORRECT AFTER ATTEMPTING
7157                                     TO CLEAR DATA LATE
7158
7159                                     ;*****
7160                                     ;*TEST 74          INTERRUPT DUE TO DATA LATE
7161                                     ;*
7162                                     ;*      ALLOW RK611 INTERRUPTS.  SET INTERRUPT ENABLE.
7163                                     ;*      NOW READ ONE WORD FROM DATA BUFFER AND MAKE SURE THAT
7164                                     ;*      DATA LATE CAUSES INTERRUPT.  BEFORE CLEARING ERROR ALLOW
7165                                     ;*      RK611 INTERRUPTS AND MAKE SURE IT DOES NOT OCCUR AGAIN.
7166                                     ;*      NOW CLEAR CONTROLLER WITH A CONTROLLER CLEAR.
7167                                     ;*
7168                                     ;*****
7169 054414 000004      TST74: SCOPE
7170 054416 012737 000764 001200      MOV      #500,$TIMES    ;DO 500. ITERATIONS
7171 054424 013702 001270          MOV      $BASE,R2       ;LOAD RK611 BASE
7172 054430 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
7173 054436 013701 002000          MOV      RKVEC,R1       ;LOAD VECTOR FOR EXPECTED INTERRUPT
7174 054442 012721 054514          MOV      #10$,(R1)+
7175 054446 012711 000340          MOV      #PR7,(R1)
7176 054452 005046          CLR      -(SP)          ;LOAD STACK TO ALLOW ALL INTERRUPTS
7177 054454 012746 054462          MOV      #64$,-(SP)    ;LOAD NEXT ADDRESS
7178 054460 000002          RTI                    ;CLEAR PSW
7179
7180 054462          64$:
7181 054462 012762 000100 000000      MOV      #IE,RKCS1(R2)  ;SET INTERRUPT ENABLE
7182 054470 005762 000024          TST     RKDB(R2)       ;READ DATA BUFFER
7183 054474 000240          NOP                    ;ALLOW INTERRUPT TO OCCUR
7184 054476 012746 000340          MOV      #PR7,-(SP)    ;LOCK OUT INTERRUPTS
7185 054502 012746 054510          MOV      #1$,-(SP)

```

```

7186 054506 000002 RTI
7187
7188 054510 104033 1$: ERROR 33 ;DATA LATE DID NOT CAUSE EXPECTED INTERRUPT
7189 054512 000422 BR 60$ ;CLEAR UP FOR NEXT TEST
7190
7191 054514 062706 000004 10$: ADD #4,SP ;ADJUST STACK
7192 054520 012777 054552 125252 MOV #15$,RKVEC ;LOAD VECTOR FOR EXPECTED INTERRUPT
7193 054526 005046 CLR -(SP) ;LOAD STACK TO ALLOW ALL INTERRUPTS
7194 054530 012746 054536 MOV #65$,-(SP) ;LOAD NEXT ADDRESS
7195 054534 000002 RTI ;CLEAR PSW
7196
7197 054536 65$: NOP ;ALLOW INTERRUPT TO OCCUR
7198 054536 000240 MOV #PR7,-(SP) ;LOCK OUT INTERRUPTS
7199 054540 012746 000340 MOV #60$,-(SP) ;CLEAN UP FOR NEXT TEST
7200 054544 012746 054560 RTI
7201 054550 000002
7202
7203 054552 062706 000004 15$: ADD #4,SP ;ADJUST STACK
7204 054556 104034 ERROR 34 ;UNEXPECTED INTERRUPT DUE TO
7205 ;UNCLEARED CONTROLLER ERROR
7206 054560 012762 100000 000000 60$: MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7207 054566 013701 002000 MOV RKVEC,R1 ;RESTORE TRAP CATCHER
7208 054572 010111 MOV R1,(R1)
7209 054574 062721 000002 ADD #2,(R1)+
7210 054600 005011 CLR (R1)
7211
7212 ;*****
7213 ;*TEST 75 INTERRUPT CLEARING AND DATA LATE
7214 ;*
7215 ;* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. CREATE
7216 ;* A CONTROLLER ERROR (DATA LATE) BY READING THE DATA BUFFER
7217 ;* WHEN EMPTY. CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
7218 ;* SET INTERRUPT ENABLE AND LOWER PROCESSOR PRIORITY.
7219 ;* MAKE SURE AN INTERRUPT DOES NOT OCCUR.
7220 ;*
7221 ;*****
7222 054602 000004 TST75: SCOPE
7223 054604 012737 000144 001200 MOV #100,$TIMES ;DO 100. ITERATIONS
7224 054612 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE
7225 054616 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7226 054624 005762 000024 TST RKDB(R2) ;CREATE DATA LATE
7227 054630 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR DATA LATE
7228 054636 013701 002000 MOV RKVEC,R1 ;LOAD VECTOR FOR UNEXPECTED INTERRUPT
7229 054642 012721 054674 MOV #5$,(R1)+
7230 054646 012711 000340 MOV #PR7,(R1)
7231 054652 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
7232 054660 005046 CLR -(SP) ;LOAD STACK TO ALLOW ALL INTERRUPTS
7233 054662 012746 054670 MOV #64$,-(SP) ;LOAD NEXT ADDRESS
7234 054666 000002 RTI ;CLEAR PSW
7235
7236 054670 64$: NOP ;ALLOW INTERRUPT TO OCCUR
7237 054670 000240 BR 60$ ;CLEAN UP FOR NEXT TEST
7238 054672 000403
7239
7240 054674 062706 000004 5$: ADD #4,SP ;ADJUST STACK
7241 054700 104035 ERROR 35 ;CONTROLLER CLEAR DID NOT CLEAR

```

```

7242                                     ; PENDING INTERRUPT DUE
7243                                     ; TO CONTROLLER ERROR
7244 054702 013701 002000 60$: MOV   RKVEC,R1      ;RESTORE TRAP CATCHER
7245 054706 010111          MOV   R1,(R1)
7246 054710 062721 000002          ADD   #2,(R1)+
7247 054714 005011          CLR   (R1)
7248
7249                                     ;*****
7250                                     ;*TEST 76      INTERRUPT ENABLE AND DATA LATE
7251                                     ;*
7252                                     ;* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. ALLOW
7253                                     ;* RK611 INTERRUPTS. READ DATA BUFFER TO GENERATE INTERRUPT
7254                                     ;* PENDING. MAKE SURE INTERRUPT DOES NOT OCCUR.
7255                                     ;*
7256                                     ;* NOW SET INTERRUPT ENABLE AND MAKE SURE INTERRUPTS OCCURS.
7257                                     ;*
7258                                     ;*****
7259 054716 000004          TST76: SCOPE
7260 054720 012737 000764 001200          MOV   #500.,$TIMES      ;;DO 500. ITERATIONS
7261 054726 013702 001270          MOV   $BASE,R2         ;LOAD RK611 BASE
7262 054732 012762 100000 000000          MOV   #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7263 054740 013701 002000          MOV   RKVEC,R1       ;LOAD VECTOR FOR UNEXPECTED INTERRUPT
7264 054744 012721 055026          MOV   #10$, (R1)+
7265 054750 012711 000340          MOV   #PR7,(R1)
7266 054754 005046          CLR   -(SP)          ;LOAD STACK TO ALLOW ALL INTERRUPTS
7267 054756 012746 054764          MOV   #64$,-(SP)     ;LOAD NEXT ADDRESS
7268 054762 000002          RTI                ;CLEAR PSW
7269
7270                                     64$:
7271 054764 005762 000024          TST   RKDB(R2)        ;READ DATA BUFFER (GENERATE DATA LATE)
7272 054770 000240          NOP
7273 054772 012777 055036 125000          MOV   #15$,RKVEC     ;ALLOW INTERRUPT TO OCCUR
7274 055000 012762 000100 000000          MOV   #IE,RKCS1(R2) ;LOAD VECTOR FOR EXPECTED INTERRUPT
7275 055006 000240          NOP                 ;SET INTERRUPT ENABLE
7276 055010 012746 000340          MOV   #PR7,-(SP)    ;ALLOW INTERRUPT TO OCCUR
7277 055014 012746 055022          MOV   #1$,-(SP)    ;LOCK OUT INTERRUPTS
7278 055020 000002          RTI
7279
7280                                     1$:
7281 055022 104037          ERROR 37            ; INTERRUPT DID NOT OCCUR WHEN
7282                                     ; INTERRUPT ENABLE SET
7283 055024 000406          BR    20$           ; CLEAN UP FOR NEXT TEST
7284
7285 055026 062706 000004          10$: ADD   #4,SP       ; ADJUST STACK
7286 055032 104036          ERROR 36           ; CONTROLLER ERROR CAUSED INTERRUPT
7287                                     ; WITH INTERRUPT ENABLE RESET
7288
7289 055034 000402          BR    20$           ; CLEAN UP FOR NEXT TEST
7290
7291 055036 062706 000004          15$: ADD   #4,SP       ; ADJUST STACK
7292 055042 012762 100000 000000          20$: MOV   #CCLR,RKCS1(R2) ; CLEAR RK611
7293 055050 013701 002000          MOV   RKVEC,R1     ; RESTORE TRAP CATCHER
7294 055054 010111          MOV   R1,(R1)
7295 055056 062721 000002          ADD   #2,(R1)+
7296 055062 005011          CLR   (R1)
    
```

```

7297 .SBTTL END OF PASS ROUTINE
7298
7299
7300 ;*****
7301 ;*INCREMENT THE PASS NUMBER ($PASS)
7302 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
7303 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
7304 ;*IF THERES A MONITOR GO TO IT
7305 ;*IF THERE ISN'T JUMP TO NEWPAS
7306 055064 $EOP:
7307 055064 000004 SCOPE
7308 055066 005037 001102 CLR $TSTNM ;;ZERO THE TEST NUMBER
7309 055072 005037 001200 CLR $TIMES ;;ZERO THE NUMBER OF ITERATIONS
7310 055076 005237 001222 INC $PASS ;;INCREMENT THE PASS NUMBER
7311 055102 042737 100000 001222 BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
7312 055110 005327 DEC (PC)+ ;;LOOP?
7313 055112 000001 $EOPCT: .WORD 1
7314 055114 003063 BGT $DOAGN ;;YES
7315 055116 012737 MOV (PC)+,@(PC)+ ;;RESTORE COUNTER
7316 055120 000001 $ENDCT: .WORD 1
7317 055122 055112 $EOPCT
7318 055124 104401 055132 TYPE ,65$ ;;TYPE ASCIZ STRING
7319 055130 000407 BR ,64$ ;;GET OVER THE ASCIZ
7320 ;;65$: .ASCIZ <12><15>/END PASS #/
7321 055150 64$:
7322 055150 013746 001222 MOV $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
7323 ;;TYPE PASS NUMBER
7324 055154 104405 TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
7325 055156 104401 055164 TYPE ,67$ ;;TYPE ASCIZ STRING
7326 055162 000421 BR ,66$ ;;GET OVER THE ASCIZ
7327 ;;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
7328 66$:
7329 055226 013746 001112 MOV $ERTTL,-(SP) ;;SAVE $ERTTL FOR TYPEOUT
7330 ;;TOTAL NUMBER OF ERRORS
7331 055232 104405 TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
7332 055234 104401 001211 TYPE $CRLF ;;TYPE CARRIAGE RETURN, LINE FEED
7333 055240 005037 001112 CLR $ERTTL ;;CLEAR ERROR TOTAL
7334 055244 013700 000042 $GET42: MOV @#42,RO ;;GET MONITOR ADDRESS
7335 055250 001405 REQ $DOAGN ;;BRANCH IF NO MONITOR
7336 055252 000005 RESET ;;CLEAR THE WORLD
7337 055254 004710 $ENDAD: JSR PC,(RO) ;;GO TO MONITOR
7338 055256 000240 NOP ;;SAVE ROOM
7339 055260 000240 NOP ;;FOR
7340 055262 000240 NOP ;;ACT11
7341 055264 $DOAGN:
7342 055264 000137 JMP @(PC)+ ;;RETURN
7343 055266 003054 $RTNAD: .WORD NEWPAS
7344 055270 377 000 000 $ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
7345 055274 .EVEN
7346
7347 .SBTTL CHECK FOR MEMORY CHECK ENABLE
7348
7349 055274 012737 055346 000004 CHKPAR: MOV #20$,ERRVEC ;;SET VECTOR ;FOR MEMORY PARTITY CHECK
7350 055302 012737 000340 000006 MOV #PR7,ERRVEC+2
7351 055310 012703 172100 MOV #MEMBAS,R3 ;;LOAD REGISTER TO DETERMINE IF
7352 ;; MEMORY CHECK ENABLE AVAILIZGLE

```

```

7353 055314 012704 000020          MOV      #16,R4          ;LOAD COUNT
7354 055320 012723 000001          MOV      #PAR.EN,(R3)+ ;ENABLE MEMORY CHECK
7355 055324 013737 055364 000114          MOV      MEMERR,MEMVEC ;LOAD MEMORY CHECK VECTOR
7356 055332 012737 000340 000116          MOV      #PR7,MEMVEC+2
7357 055340 005304          DEC      R4          ;CHECK IF FINISHED
7358 055342 001366          BNE     16$          ;NO, SET UP NEXT MEMORY PARITY MODULE
7359 055344 000401          BR      22$          ;RESTORE TRAP VECTOR
7360
7361 055346 022626          20$:  CMP      (SP)+,(SP)+ ;ADJUST STACK
7362 055350 012737 000006 000004          22$:  MOV      #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
7363 055356 005037 000006          CLR     ERRVEC+2
7364 055362 000207          RTS     PC          ;RETURN
7365
7366          .SBTTL MEMORY CHECK ENABLE TRAP
7367
7368 055364 012737 055400 001202 MEMERR: MOV      #10$,SESCAPE ;LOAD ESCAPE
7369 055372 011637 002024          MOV      (SP),TRAPPC ;STORE PC
7370 055376 104040          ERROR   40          ;REPORT MEM PARITY ERROR
7371 055400 005037 001202          10$:  CLR     $ESCAPE ;CLEAR ESCAPE
7372 055404 032777 001000 123526          BIT     #SW9,$SWR ;CHECK IF LOOP ON ERROR
7373 055412 001001          BNE     15$          ;YES, FORCE STACK AND TRY AGAIN
7374 055414 000002          RTI     ;NO, RETURN
7375
7376 055416 012706 001100          15$:  MOV      #STACK,SP ;INITIALIZE STACK
7377 055422 000177 123462          JMP     $SLPERR ;LOOP ON ERROR
7378
7379          .SBTTL SCOPE HANDLER ROUTINE
7380
7381          ;*****
7382          ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
7383          ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
7384          ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
7385          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7386          ;*SW14=1 LOOP ON TEST
7387          ;*SW11=1 INHIBIT ITERATIONS
7388          ;*SW09=1 LOOP ON ERROR
7389          ;*SW08=1 LOOP ON TEST IN SWR<7:0>
7390          ;*CALL
7391          ;* SCOPE ;:SCOPE=IOT
7392
7393          $SCOPE:
7394 055426 104407          CKSWR
7395 055430 032777 040000 123502          1$:  BIT     #BIT14,$SWR ;:TEST FOR CHANGE IN SOFT-SWR
7396 055436 001131          BNE     $OVER ;:LOOP ON PRESENT TEST?
7397          ;*****START OF CODE FOR THE XOR TESTER***** ;:YES IF SW14=1
7398 055440 000416          $XTSTR: BR      6$ ;:IF RUNNING ON THE "XOR" TESTER CHANGE
7399          ;THIS INSTRUCTION TO A "NOP" (NOP=240)
7400 055442 013746 000004          MOV     2#ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
7401 055446 012737 055466 000004          MOV     #5,2#ERRVEC ;:SET FOR TIMEOUT
7402 055454 005737 177060          TST    2#177060 ;:TIME OUT ON XOR?
7403 055460 012637 000004          MOV     (SP)+,2#ERRVEC ;:RESTORE THE ERROR VECTOR
7404 055464 000500          BR     $SVLAD ;:GO TO THE NEXT TEST
7405 055466 022626          5$:  CMP     (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
7406 055470 012637 000004          MOV     (SP)+,2#ERRVEC ;:RESTORE THE ERROR VECTOR
7407 055474 000440          BR     7$ ;:LOOP ON THE PRESENT TEST
7408 055476          6$: ;*****END OF CODE FOR THE XOR TESTER*****

```

7409	055476	032777	000400	123434		BIT	#BIT08,\$SWR	:: LOOP ON SPEC. TEST?
7410	055504	001421				BEQ	2\$:: BR IF NO
7411	055506	005046				CLR	-(SP)	:: CLEAR A TEMP. LOCATION
7412	055510	117716	123424			MOVB	\$SWR,(SP)	:: PICKUP THE DESIRED TEST NUMBER
7413	055514	001414				BEQ	8\$:: BRANCH IF BAD TEST NUMBER IN SWR
7414	055516	022716	000076			CMP	#76,(SP)	:: CHECK THE NUMBER IN THE SWR
7415	055522	002411				BLT	8\$:: BRANCH IF TEST NUMBER IS OUT OF RANGE
7416	055524	011637	001102			MOV	(SP),\$STNM	:: UPDATE THE TEST NUMBER
7417	055530	005316				DEC	(SP)	:: BACKUP BY ONE
7418	055532	006316				ASL	(SP)	:: SCALE THE TEST NUMBER AS AN INDEX
7419	055534	062716	055740			ADD	\$\$SW08TBL,(SP)	:: FORM THE ADDRESS OF TEST POINTER
7420	055540	013637	001106			MOV	\$(\$SP)+,\$LPADR	:: SET LOOP ADDRESS TO DESIRED TEST
7421	055544	000466				BR	\$OVER	:: GO LOOP ON THE TEST
7422	055546	005726			8\$:	TST	(SP)+	:: CLEAN THE BAD TEST NUMBER OFF OF THE STACK
7423	055550	105737	001103		2\$:	TSTB	\$ERFLG	:: HAS AN ERROR OCCURRED?
7424	055554	001421				BEQ	3\$:: BR IF NO
7425	055556	123737	001115	001103		CMPB	\$ERMAX,\$ERFLG	:: MAX. ERRORS FOR THIS TEST OCCURRED?
7426	055564	101015				BHI	3\$:: BR IF NO
7427	055566	032777	001000	123344		BIT	#BIT09,\$SWR	:: LOOP ON ERROR?
7428	055574	001404				BEQ	4\$:: BR IF NO
7429	055576	013737	001110	001106	7\$:	MOV	\$LPERR,\$LPADR	:: SET LOOP ADDRESS TO LAST SCOPE
7430	055604	000446				BR	\$OVER	
7431	055606	105037	001103		4\$:	CLRB	\$ERFLG	:: ZERO THE ERROR FLAG
7432	055612	005037	001200			CLR	\$TIMES	:: CLEAR THE NUMBER OF ITERATIONS TO MAKE
7433	055616	000415				BR	1\$:: ESCAPE TO THE NEXT TEST
7434	055620	032777	004000	123312	3\$:	BIT	#BIT11,\$SWR	:: INHIBIT ITERATIONS?
7435	055626	001011				BNE	1\$:: BR IF YES
7436	055630	005737	001222			TST	\$PASS	:: IF FIRST PASS OF PROGRAM
7437	055634	001406				BEQ	1\$:: INHIBIT ITERATIONS
7438	055636	005237	001104			INC	\$ICNT	:: INCREMENT ITERATION COUNT
7439	055642	023737	001200	001104		CMP	\$TIMES,\$ICNT	:: CHECK THE NUMBER OF ITERATIONS MADE
7440	055650	002024				BGE	\$OVER	:: BR IF MORE ITERATION REQUIRED
7441	055652	012737	000001	001104	1\$:	MOV	#1,\$ICNT	:: REINITIALIZE THE ITERATION COUNTER
7442	055660	013737	055736	001200		MOV	\$MXCNT,\$TIMES	:: SET NUMBER OF ITERATIONS TO DC
7443	055666	105237	001102		\$SVLAD:	INCB	\$STNM	:: COUNT TEST NUMBERS
7444	055672	113737	001102	001220		MOVB	\$STNM,\$STNM	:: SET TEST NUMBER IN APT MAILBOX
7445	055700	011637	001106			MOV	(SP),\$LPADR	:: SAVE SCOPE LOOP ADDRESS
7446	055704	011637	001110			MOV	(SP),\$LPERR	:: SAVE ERROR LOOP ADDRESS
7447	055710	005037	001202			CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
7448	055714	112737	000001	001115		MOVB	#1,\$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7449	055722	013777	001102	123212	\$OVER:	MOV	\$STNM,\$DISPLAY	:: DISPLAY TEST NUMBER
7450	055730	013716	001106			MOV	\$LPADR,(SP)	:: FUDGE RETURN ADDRESS
7451	055734	000002				RTI		:: FIXES PS
7452	055736	003720			\$MXCNT:	2000.		:: MAX. NUMBER OF ITERATIONS
7453	055740				\$SW08TBL:			
7454	055740	003074			.WORD	TST1+2		:: STARTING ADDRESS OF TEST 1
7455	055742	003166			.WORD	TST2+2		:: STARTING ADDRESS OF TEST 2
7456	055744	003726			.WORD	TST3+2		:: STARTING ADDRESS OF TEST 3
7457	055746	004466			.WORD	TST4+2		:: STARTING ADDRESS OF TEST 4
7458	055750	005222			.WORD	TST5+2		:: STARTING ADDRESS OF TEST 5
7459	055752	005320			.WORD	TST6+2		:: STARTING ADDRESS OF TEST 6
7460	055754	006130			.WORD	TST7+2		:: STARTING ADDRESS OF TEST 7
7461	055756	006714			.WORD	TST10+2		:: STARTING ADDRESS OF TEST 10
7462	055760	007542			.WORD	TST11+2		:: STARTING ADDRESS OF TEST 11
7463	055762	010370			.WORD	TST12+2		:: STARTING ADDRESS OF TEST 12
7464	055764	011214			.WORD	TST13+2		:: STARTING ADDRESS OF TEST 13

7465	055766	012040	.WORD	TST14+2	:::STARTING ADDRESS OF TEST 14
7466	055770	012726	.WORD	TST15+2	:::STARTING ADDRESS OF TEST 15
7467	055772	013616	.WORD	TST16+2	:::STARTING ADDRESS OF TEST 16
7468	055774	014502	.WORD	TST17+2	:::STARTING ADDRESS OF TEST 17
7469	055776	015366	.WORD	TST20+2	:::STARTING ADDRESS OF TEST 20
7470	056000	016074	.WORD	TST21+2	:::STARTING ADDRESS OF TEST 21
7471	056002	016712	.WORD	TST22+2	:::STARTING ADDRESS OF TEST 22
7472	056004	017530	.WORD	TST23+2	:::STARTING ADDRESS OF TEST 23
7473	056006	020344	.WORD	TST24+2	:::STARTING ADDRESS OF TEST 24
7474	056010	021160	.WORD	TST25+2	:::STARTING ADDRESS OF TEST 25
7475	056012	022006	.WORD	TST26+2	:::STARTING ADDRESS OF TEST 26
7476	056014	022634	.WORD	TST27+2	:::STARTING ADDRESS OF TEST 27
7477	056016	023460	.WORD	TST30+2	:::STARTING ADDRESS OF TEST 30
7478	056020	024304	.WORD	TST31+2	:::STARTING ADDRESS OF TEST 31
7479	056022	025132	.WORD	TST32+2	:::STARTING ADDRESS OF TEST 32
7480	056024	025760	.WORD	TST33+2	:::STARTING ADDRESS OF TEST 33
7481	056026	026604	.WORD	TST34+2	:::STARTING ADDRESS OF TEST 34
7482	056030	027430	.WORD	TST35+2	:::STARTING ADDRESS OF TEST 35
7483	056032	030300	.WORD	TST36+2	:::STARTING ADDRESS OF TEST 36
7484	056034	031150	.WORD	TST37+2	:::STARTING ADDRESS OF TEST 37
7485	056036	032014	.WORD	TST40+2	:::STARTING ADDRESS OF TEST 40
7486	056040	032660	.WORD	TST41+2	:::STARTING ADDRESS OF TEST 41
7487	056042	033424	.WORD	TST42+2	:::STARTING ADDRESS OF TEST 42
7488	056044	034132	.WORD	TST43+2	:::STARTING ADDRESS OF TEST 43
7489	056046	034640	.WORD	TST44+2	:::STARTING ADDRESS OF TEST 44
7490	056050	035346	.WORD	TST45+2	:::STARTING ADDRESS OF TEST 45
7491	056052	036054	.WORD	TST46+2	:::STARTING ADDRESS OF TEST 46
7492	056054	036702	.WORD	TST47+2	:::STARTING ADDRESS OF TEST 47
7493	056056	037530	.WORD	TST50+2	:::STARTING ADDRESS OF TEST 50
7494	056060	040354	.WORD	TST51+2	:::STARTING ADDRESS OF TEST 51
7495	056062	041200	.WORD	TST52+2	:::STARTING ADDRESS OF TEST 52
7496	056064	042044	.WORD	TST53+2	:::STARTING ADDRESS OF TEST 53
7497	056066	042710	.WORD	TST54+2	:::STARTING ADDRESS OF TEST 54
7498	056070	043552	.WORD	TST55+2	:::STARTING ADDRESS OF TEST 55
7499	056072	044414	.WORD	TST56+2	:::STARTING ADDRESS OF TEST 56
7500	056074	045122	.WORD	TST57+2	:::STARTING ADDRESS OF TEST 57
7501	056076	045630	.WORD	TST60+2	:::STARTING ADDRESS OF TEST 60
7502	056100	046704	.WORD	TST61+2	:::STARTING ADDRESS OF TEST 61
7503	056102	047170	.WORD	TST62+2	:::STARTING ADDRESS OF TEST 62
7504	056104	047276	.WORD	TST63+2	:::STARTING ADDRESS OF TEST 63
7505	056106	047420	.WORD	TST64+2	:::STARTING ADDRESS OF TEST 64
7506	056110	047604	.WORD	TST65+2	:::STARTING ADDRESS OF TEST 65
7507	056112	050506	.WORD	TST66+2	:::STARTING ADDRESS OF TEST 66
7508	056114	051306	.WORD	TST67+2	:::STARTING ADDRESS OF TEST 67
7509	056116	052106	.WORD	TST70+2	:::STARTING ADDRESS OF TEST 70
7510	056120	052704	.WORD	TST71+2	:::STARTING ADDRESS OF TEST 71
7511	056122	053502	.WORD	TST72+2	:::STARTING ADDRESS OF TEST 72
7512	056124	054220	.WORD	TST73+2	:::STARTING ADDRESS OF TEST 73
7513	056126	054416	.WORD	TST74+2	:::STARTING ADDRESS OF TEST 74
7514	056130	054604	.WORD	TST75+2	:::STARTING ADDRESS OF TEST 75
7515	056132	054720	.WORD	TST76+2	:::STARTING ADDRESS OF TEST 76
7516			;:*****		
7517			.SBTTL LOOP ON INTERNAL ERROR		
7518					
7519					
7520	056134	032777	001000	122776	SCOPI\$: BIT #SW9,2SWR ;CHECK IF LOOP IN ERROR

E13

```

7521 056142 001405          BEQ      5$          ;NO, CONTINUE TEST
7522 056144 105737 001103    TSTB    $ERFLG      ;CHECK IF ERROR OCCURRED
7523 056150 001402          BEQ      5$          ;NO, CONTINUE
7524 056152 013716 001110    MOV     $LPERR, (SP) ;LOAD ERROR RETURN
7525 056156 000002          SS:     RTI          ;RETURN
7526                                     .SBTTL  APT COMMUNICATIONS ROUTINE
7527
7528                                     ;*****
7529 056160 112737 000001 056424  SATY1:  MOVB    #1, $FFLG  ;; TO REPORT FATAL ERROR
7530 056166 112737 000001 056422  SATY3:  MOVB    #1, $MFLG  ;; TO TYPE A MESSAGE
7531 056174 000403          BR      SATYC
7532 056176 112737 000001 056424  SATY4:  MOVB    #1, $FFLG  ;; TO ONLY REPORT FATAL ERROR
7533 056204          SATYC:
7534 056204 010046          MOV     RO, -(SP)    ;; PUSH RO ON STACK
7535 056206 010146          MOV     RI, -(SP)    ;; PUSH RI ON STACK
7536 056210 105737 056422    TSTB    $MFLG      ;; SHOULD TYPE A MESSAGE?
7537 056214 001450          BEQ     5$          ;; IF NOT: BR
7538 056216 122737 000001 001234  CMPB    #APTENV, $ENV ;; OPERATING UNDER APT?
7539 056224 001031          BNE    3$          ;; IF NOT: BR
7540 056226 132737 000100 001235  BITB    #APTSPOOL, $ENVM ;; SHOULD SPOOL MESSAGES?
7541 056234 001425          BEQ     3$          ;; IF NOT: BR
7542 056236 017600 000004          MOV     @4(SP), RO   ;; GET MESSAGE ADDR.
7543 056242 062766 000002 000004  ADD     #2, 4(SP)    ;; BUMP RETURN ADDR.
7544 056250 005737 001214          1$:    TST     $MSGTYPE  ;; SEE IF DONE W/ LAST XMISSION?
7545 056254 001375          BNE    1$          ;; IF NOT: WAIT
7546 056256 010037 001230    MOV     RO, $MSGAD  ;; PUT ADDR IN MAILBOX
7547 056262 105720          2$:    TSTB    (RO)+    ;; FIND END OF MESSAGE
7548 056264 001376          BNE    2$
7549 056266 163700 001230    SUB     $MSGAD, RO   ;; SUB START OF MESSAGE
7550 056272 006200          ASR    RO          ;; GET MESSAGE LNTH IN WORDS
7551 056274 010037 001232    MOV     RO, $MSGGLT ;; PUT LENGTH IN MAILBOX
7552 056300 012737 000004 001214  MOV     #4, $MSGTYPE ;; TELL APT TO TAKE MSG.
7553 056306 000413          BR      5$
7554 056310 017637 000004 056334  3$:    MOV     @4(SP), 4$  ;; PUT MSG ADDR IN JSR LINKAGE
7555 056316 062766 000002 000004  ADD     #2, 4(SP)    ;; BUMP RETURN ADDRESS
7556 056324 013746 177776          MOV     177776, -(SP) ;; PUSH 177776 ON STACK
7557 056330 004737 051110          JSR    PC, $TYPE    ;; CALL TYPE MACRO
7558 056334 000000          4$:    .WORD    0
7559                                     5$:
7560 056336 105737 056424          10$:   TSTB    $FFLG      ;; SHOULD REPORT FATAL ERROR?
7561 056342 001416          BEQ    12$         ;; IF NOT: BR
7562 056344 005737 001234          TST     $ENV        ;; RUNNING UNDER APT?
7563 056350 001413          BEQ    12$         ;; IF NOT: BR
7564 056352 005737 001214          11$:   TST     $MSGTYPE    ;; FINISHED LAST MESSAGE?
7565 056356 001375          BNE    11$        ;; IF NOT: WAIT
7566 056360 017637 000004 001216  MOV     @4(SP), $FATAL ;; GET ERROR #
7567 056366 062766 000002 000004  ADD     #2, 4(SP)    ;; BUMP RETURN ADDR.
7568 056374 005237 001214          INC     $MSGTYPE    ;; TELL APT TO TAKE ERROR
7569 056400 105037 056424          12$:   CLRB    $FFLG      ;; CLEAR FATAL FLAG
7570 056404 105037 056423          CLRB    $LFLG      ;; CLEAR LOG FLAG
7571 056410 105037 056422          CLRB    $MFLG      ;; CLEAR MESSAGE FLAG
7572 056414 012601          MOV     (SP)+, R1   ;; POP STACK INTO R1
7573 056416 012600          MOV     (SP)+, R0   ;; POP STACK INTO R0
7574 056420 000207          RTS     PC          ;; RETURN
7575 056422          000          $MFLG: .BYTE    0   ;; MESSG. FLAG
7576 056423          000          $LFLG: .BYTE    0   ;; LOG FLAG
  
```

F13

```

7577 056424 000 $FFLG: .BYTE 0 ;;FATAL FLAG
7578 056426 056426 .EVEN
7579 000200 APTSIZE=200
7580 000001 APTENV=001
7581 000100 APTSPOOL=100
7582 000040 APTCSUP=040
7583 .SBTTL ERROR HANDLER ROUTINE
7584
7585 ;:*****
7586 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7587 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7588 ;*AND GO TO TYPERR ON ERROR
7589 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7590 ;*SW15=1 HALT ON ERROR
7591 ;*SW13=1 INHIBIT ERROR TYPEOUTS
7592 ;*SW10=1 BELL ON ERROR
7593 ;*SW09=1 LOOP ON ERROR
7594 ;*CALL
7595 ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
7596
7597 056426 $ERROR:
7598 056426 104407 CKSWR ;; TEST FOR CHANGE IN SOFT-SWR
7599 056430 105237 001103 7$: INCB $ERFLG ;; SET THE ERROR FLAG
7600 056434 001775 BEQ 7$ ;; DON'T LET THE FLAG GO TO ZERO
7601 056436 013777 001102 122476 MOV $TSTNM,$DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
7602 056444 032777 002000 122466 BIT #BIT10,$SWR ;; BELL ON ERROR?
7603 056452 001402 BEQ 1$ ;; NO - SKIP
7604 056454 104401 001204 TYPE $BELL ;; RING BELL
7605 056460 005237 001112 1$: INC $ERTTL ;; COUNT THE NUMBER OF ERRORS
7606 056464 011637 001116 MOV (SP),$ERRPC ;; GET ADDRESS OF ERROR INSTRUCTION
7607 056470 162737 000002 001116 SUB #2,$ERRPC
7608 056476 117737 122414 001114 MOVB $ERRPC,$ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
7609 056504 032777 020000 122426 BIT #BIT13,$SWR ;; SKIP TYPEOUT IF SET
7610 056512 001004 BNE 20$ ;; SKIP TYPEOUTS
7611 056514 004737 056626 JSR PC,TYPERR ;; GO TO USER ERROR ROUTINE
7612 056520 104401 001211 TYPE , $CRLF
7613 056524
7614 056524 122737 000001 001234 20$: CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
7615 056532 001007 BNE 2$ ;; NO SKIP APT ERROR REPORT
7616 056534 113737 001114 056546 MOVB $ITEMB,21$ ;; SET ITEM NUMBER AS ERROR NUMBER
7617 056542 004737 056176 JSR PC,$ATY4 ;; REPORT FATAL ERROR TO APT
7618 056546 000 .BYTE 0
7619 056547 000 .BYTE 0
7620 056550 000777 BR 22$ ;; APT ERROR LOOP
7621 056552 005777 122362 2$: TST $SWR ;; HALT ON ERROR
7622 056556 100002 BPL 3$ ;; SKIP IF CONTINUE
7623 056560 000000 HALT ;; HALT ON ERROR!
7624 056562 104407 CKSWR
7625 056564 032777 001000 122346 3$: BIT #BIT09,$SWR ;; TEST FOR CHANGE IN SOFT-SWR
7626 056572 001402 BEQ 4$ ;; LOOP ON ERROR SWITCH SET?
7627 056574 013716 001110 MOV $LPERR,(SP) ;; BR IF NO
7628 056600 005737 001202 4$: TST $ESCAPE ;; FUDGE RETURN FOR LOOPING
7629 056604 001402 BEQ 5$ ;; CHECK FOR AN ESCAPE ADDRESS
7630 056606 013716 001202 MOV $ESCAPE,(SP) ;; BR IF NONE
7631 056612
7632 056612 022737 055254 000042 5$: CMP #SENDAD,$42 ;; FUDGE RETURN ADDRESS FOR ESCAPE
;;ACT-11 AUTO-ACCEPT?

```

```

7633 056620 001001          BNE      6$          ;;BRANCH IF NO
7634 056622 000000          HALT           ;;YES
7635 056624          6$:          RTI              ;;RETURN
7636 056624 000002
7637
7638 ;*****
7639 ;SBTTL TYPE ERROR ROUTINE
7640 ;*ENTRY JSR PC,TYPERR
7641 ;*RETURN RTS PC
7642 ;*
7643 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
7644 ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
7645 ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
7646 ;*THE ERROR.
7647 ;*****
7648 056626 104413          TYPERR: SAVREG
7649 056630 113700 001114      MOV      $ITEMB,R0      ;ENTER ERROR NUMBER
7650 056634 042700 177400      BIC      #177400,R0     ;CLEAR UNUSED BITS
7651 056640 005300          DEC      R0            ;FORM INDEX FOR ERROR TABLE
7652 056642 006300          ASL      R0
7653 056644 006300          ASL      R0
7654 056646 006300          ASL      R0
7655 056650 062700 001300      1$:      ADD      #$ERRTB,R0   ;FORM ADDRESS OF ERROR ENTRY
7656 056654 012037 056670      MOV      (R0)+,2$      ;GET EM POINTER
7657 056660 001404          BEQ      3$            ;BRANCH IF THERE ISN'T ONE
7658 056662 104401 001211      TYPE    ,SCRLF        ;TYPE CARRIAGE RETURN LINE FEED
7659 056666 104401          TYPE    ;TYPE ERROR MESSAGE (EM)
7660 056670 000000          .WORD   0            ;EM POINTER GOES HERE
7661 056672 012037 056706      3$:      MOV      (R0)+,4$      ;GET DH POINTER
7662 056676 001404          BEQ      5$            ;BRANCH IF THERE ISN'T ONE
7663 056700 104401 001211      TYPE    ,SCRLF        ;TYPE CR-LF
7664 056704 104401          TYPE    ;TYPE DATA HEADER
7665 056706 000000          .WORD   0            ;DH POINTER GOES HERE
7666 056710 012001          5$:      MOV      (R0)+,R1      ;GET DT POINTER
7667 056712 001445          BEQ      20$           ;BRANCH IF THERE ARE NONE
7668 056714 005004          CLR      R4           ;RESET INDENT SWITCH
7669 056716 012000          MOV      (R0)+,R0     ;GET DF POINTER
7670 056720 012002          MOV      (R0)+,R2     ;STORE NUMBER OF DH'S
7671 056722 104401 001211      TYPE    ,SCRLF        ;TYPE <CR><LF>
7672 056726 112003          10$:     MOV      (R0)+,R3      ;GET & STORE NUMBER OF DATA WORDS
7673 056730 105720          TST      (R0)+       ;BUMP PAST FORMAT WORD
7674 056732 005703          TST      R3          ;TEST IF ANY DATA FOR THIS HEADER
7675 056734 001416          BEQ      14$         ;NO - SKIP DATA PRINT
7676 056736 005704          TST      R4          ;CHECK IF INDENT WORDS
7677 056740 001004          BNE      12$         ;YES, GO INDENT
7678 056742 013146          11$:     MOV      2(R1)+,-(SP)  ;PUT FIRST DATA WORD ON STACK
7679 056744 104402          TYPOC          ;TYPE IT
7680 056746 005303          DEC      R3          ;MORE DATA WORDS
7681 056750 001403          BEQ      13$         ;NO-BRANCH
7682 056752 104401 062171      12$:     TYPE    ,SPACE2   ;TYPE SEPARATORS
7683 056756 000771          BR       11$         ;LOOP
7684 056760 104401 001211      13$:     TYPE    ,SCRLF        ;TYPE <CR><LF>
7685 056764 005710          TST      (R0)        ;CHECK IF NEXT HEADER AVAILABLE
7686 056766 001401          BEQ      14$         ;NO, DO NOT CHANGE INDENT
7687 056770 005104          COM      R4          ;CHANGE INDENT
7688 056772 005302          14$:     DEC      R2          ;MORE DH'S?

```

```

7689 056774 003414          BLE      20$      ;NO-BRANCH
7690 056776 012037 057016    15$:     MOV      (R0)+,18$ ;GET NEXT DH POINTER
7691 057002 001751          BEQ      10$      ;IF NO HEADER GET DATA
7692 057004 005704          TST      R4       ;INDENT?
7693 057006 001402          BEQ      17$      ;NO-BRANCH
7694 057010 104401 062171          TYPE    ,SPACE2  ;INDENT
7695 057014 104401          17$:     TYPE    ;TYPE DH
7696 057016 000000          18$:     .WORD   0       ;DH POINTER GOES HERE
7697 057020 104401 001211          TYPE    $CRLF    ;
7698 057024 000740          BR       10$      ;LOOP
7699 057026 104414          20$:     RESREG
7700 057030 005237 002006          INC     ERRCNT   ;INCREMENT ERROR COUNT
7701 057034 032777 010000 122076          BIT     #SW12,$SWR ;CHECK IF ABORT AFTER 20 ERRORS
7702 057042 001421          BEQ     25$      ;NO, TYPE OUT ERROR
7703 057044 022737 000024 002006          CMP     #20.,ERRCNT ;CHECK IF ERROR THRESHOLD EXCEEDED
7704 057052 103015          BHIS   25$      ;NO, PRINT ERROR
7705 057054 104401 062174          TYPE    ,ABORT   ;TYPE "PROGRAM HAS BEEN ABORTED BECAUSE
7706                                ; ERROR THRESHOLD EXCEEDED"
7707 057060 005737 000042          TST     42       ;CHECK IF IN CHAIN MODE
7708 057064 001407          BEQ     30$      ;NO, HALT
7709 057066 012737 000001 055112          MOV     #1,$EOPCT ;FORCE END OF PASS COUNT TO ONE FOR ABORT
7710 057074 012706 001100          MOV     #STACK,SP ;INITIALIZE STACK
7711 057100 000137 055064          JMP     $EOP     ;BRING IN NEXT PROGRAM IN CHAIN
7712 057104 000000          30$:     HALT
7713 057106 000207          25$:     RTS     PC
7714                                .SBTTL  TYPE ROUTINE
7715
7716                                ;*****
7717                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
7718                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
7719                                ;*NOTE1:          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
7720                                ;*NOTE2:          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
7721                                ;*NOTE3:          $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
7722                                ;*
7723                                ;*CALL:
7724                                ;*1) USING A TRAP INSTRUCTION
7725                                ;*      TYPE    ,MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
7726                                ;*OR
7727                                ;*      TYPE
7728                                ;*      MESADR
7729                                ;*
7730
7731 057110 105737 001157          $TYPE:  TSTB   $TFPLG ;: IS THERE A TERMINAL?
7732 057114 100002          BPL     1$       ;: BR IF YES
7733 057116 000000          HALT
7734 057120 000430          BR     3$       ;: HALT HERE IF NO TERMINAL
7735 057122 010046          1$:     MOV     RO,-(SP) ;: LEAVE
7736 057124 017600 000002          MOV     @2(SP),RO ;: SAVE RO
7737 057130 122737 000001 001234          CMPB   #APTENV,$ENV ;: GET ADDRESS OF ASCIZ STRING
7738 057136 001011          BNE    62$      ;: RUNNING IN APT MODE
7739 057140 132737 000100 001235          BITB   #APTPOOL,$ENVM ;: NO GO CHECK FOR APT CONSOLE
7740 057146 001405          BEQ    62$      ;: SPOOL MESSAGE TO APT
7741 057150 010037 057160          MOV     RO,61$  ;: NO GO CHECK FOR CONSOLE
7742 057154 004737 056166          JSR    PC,$ATY3 ;: SETUP MESSAGE ADDRESS FOR APT
7743 057160 000000          61$:     .WORD   0   ;: SPOOL MESSAGE TO APT
7744 057162 132737 000040 001235          62$:     BITB   #APTCSUP,$ENVM ;: MESSAGE ADDRESS
;: APT CONSOLE SUPPRESSED

```

```

7745 057170 001003          BNE      60$          ;; YES, SKIP TYPE OUT
7746 057172 112046          2$:     MOVB     (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
7747 057174 001005          BNE      4$          ;; BR IF IT ISN'T THE TERMINATOR
7748 057176 005726          TST     (SP)+        ;; IF TERMINATOR POP IT OFF THE STACK
7749 057200 012600          60$:    MOV      (SP)+, RO    ;; RESTORE RO
7750 057202 062716 000002        3$:     ADD      #2, (SP)    ;; ADJUST RETURN PC
7751 057206 000002          RTI                    ;; RETURN
7752 057210 122716 000011        4$:     CMPB     #HT, (SP)    ;; BRANCH IF <HT>
7753 057214 001430          BEQ     8$          ;; BRANCH IF NOT <CRLF>
7754 057216 122716 000200          CMPB     #CRLF, (SP)
7755 057222 001006          BNE     5$          ;; BRANCH IF NOT <CR><LF> EQUIV
7756 057224 005726          TST     (SP)+        ;; TYPE A CR AND LF
7757 057226 104401          TYPE
7758 057230 001211          $CRLF
7759 057232 105037 057366          CLRB    $CHARCNT    ;; CLEAR CHARACTER COUNT
7760 057236 000755          BR      2$          ;; GET NEXT CHARACTER
7761 057240 004737 057322        5$:     JSR     PC, $TYPEC    ;; GO TYPE THIS CHARACTER
7762 057244 123726 001156        6$:     CMPB     $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
7763 057250 001350          BNE     2$          ;; IF NO GO GET NEXT CHAR.
7764 057252 013746 001154          MOV     $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
7765                                ;; AND THE NULL CHAR.
7766 057256 105366 000001        7$:     DECB    1(SP)      ;; DOES A NULL NEED TO BE TYPED?
7767 057262 002770          BLT     6$          ;; BR IF NO--GO POP THE NULL OFF OF STACK
7768 057264 004737 057322          JSR     PC, $TYPEC    ;; GO TYPE A NULL
7769 057270 105337 057366          DECB    $CHARCNT    ;; DO NOT COUNT AS A COUNT
7770 057274 000770          BR      7$          ;; LOOP

```

; HORIZONTAL TAB PROCESSOR

```

7771
7772
7773
7774 057276 112716 000040        8$:     MOVB     #' (SP)    ;; REPLACE TAB WITH SPACE
7775 057302 004737 057322        9$:     JSR     PC, $TYPEC    ;; TYPE A SPACE
7776 057306 132737 000007 057366  BITB     #7, $CHARCNT ;; BRANCH IF NOT AT
7777 057314 001372          BNE     9$          ;; TAB STOP
7778 057316 005726          TST     (SP)+        ;; POP SPACE OFF STACK
7779 057320 000724          BR      2$          ;; GET NEXT CHARACTER
7780 057322 105777 121622        $TYPEC: TSTB    2$STPS    ;; WAIT UNTIL PRINTER IS READY
7781 057326 100375          BPL     $TYPEC
7782 057330 116677 000002 121614  MOVB     2(SP), 2$STPB ;; LOAD CHAR TO BE TYPED INTO DATA REG.
7783 057336 122766 000015 000002  CMPB     #CR, 2(SP)    ;; IS CHARACTER A CARRIAGE RETURN?
7784 057344 001003          BNE     1$          ;; BRANCH IF NO
7785 057346 105037 057366          CLRB    $CHARCNT    ;; YES--CLEAR CHARACTER COUNT
7786 057352 000406          BR      $TYPEX
7787 057354 122766 000012 000002  1$:     CMPB     #LF, 2(SP) ;; IS CHARACTER A LINE FEED?
7788 057362 001402          BEQ     $TYPEX      ;; BRANCH IF YES
7789 057364 105227          INCB    (PC)+        ;; COUNT THE CHARACTER
7790 057366 000000          $CHARCNT: .WORD    0 ;; CHARACTER COUNT STORAGE
7791 057370 000207          $TYPEX:  RTS     PC

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

7792
7793
7794
7795 *****
7796 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7797 *OCTAL (ASCII) NUMBER AND TYPE IT.
7798 *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7799 *CALL:
7800 *      MOV     NUM, -(SP) ;; NUMBER TO BE TYPED

```

```

7801          :*      TYPOS          ;;CALL FOR TYPEOUT
7802          :*      .BYTE      N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7803          :*      .BYTE      M          ;;M=1 OR 0
7804          :*                                  ;;1=TYPE LEADING ZEROS
7805          :*                                  ;;0=SUPPRESS LEADING ZEROS
7806          :*
7807          :*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
7808          :*$TYPOS OR $TYPOC
7809          :*CALL:
7810          :*      MOV          NUM,-(SP)      ;;NUMBER TO BE TYPED
7811          :*      TYPON          ;;CALL FOR TYPEOUT
7812          :*
7813          :*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
7814          :*CALL:
7815          :*      MOV          NUM,-(SP)      ;;NUMBER TO BE TYPED
7816          :*      TYPOC          ;;CALL FOR TYPEOUT
7817          :*
7818 057372 017646 000000          $TYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
7819 057376 116637 000001 057615  MOVB     1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
7820 057404 112637 057617          MOVB     (SP)+, $OMODE+1      ;;NUMBER OF DIGITS TO TYPE
7821 057410 062716 000002          ADD      #2, (SP)          ;;ADJUST RETURN ADDRESS
7822 057414 000406
7823 057416 112737 000001 057615  $TYPOC: MOVB     #1, $OFILL      ;;SET THE ZERO FILL SWITCH
7824 057424 112737 000006 057617  MOVB     #6, $OMODE+1      ;;SET FOR SIX(6) DIGITS
7825 057432 112737 000005 057614  $TYPON: MOVB     #5, $OCNT      ;;SET THE ITERATION COUNT
7826 057440 010346          MOV      R3,-(SP)          ;;SAVE R3
7827 057442 010446          MOV      R4,-(SP)          ;;SAVE R4
7828 057444 010546          MOV      R5,-(SP)          ;;SAVE R5
7829 057446 113704 057617          MOVB     $OMODE+1, R4      ;;GET THE NUMBER OF DIGITS TO TYPE
7830 057452 005404          NEG      R4
7831 057454 062704 000006          ADD      #6, R4          ;;SUBTRACT IT FOR MAX. ALLOWED
7832 057460 110437 057616          MOVB     R4, $OMODE      ;;SAVE IT FOR USE
7833 057464 113704 057615          MOVB     $OFILL, R4      ;;GET THE ZERO FILL SWITCH
7834 057470 016605 000012          MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
7835 057474 005003          CLR      R3          ;;CLEAR THE OUTPUT WORD
7836 057476 006105          1$: ROL     R5          ;;ROTATE MSB INTO "C"
7837 057500 000404          BR      3$          ;;GO DO MSB
7838 057502 006105          2$: ROL     R5          ;;FORM THIS DIGIT
7839 057504 006105
7840 057506 006105
7841 057510 010503
7842 057512 006103          3$: ROL     R3          ;;GET LSB OF THIS DIGIT
7843 057514 105337 057616          DECB     $OMODE          ;;TYPE THIS DIGIT?
7844 057520 100016          BPL     7$          ;;BR IF NO
7845 057522 042703 177770          BIC     #177770, R3      ;;GET RID OF JUNK
7846 057526 001002          BNE     4$          ;;TEST FOR 0
7847 057530 005704          TST     R4          ;;SUPPRESS THIS 0?
7848 057532 001403          BEQ     5$          ;;BR IF YES
7849 057534 005204          4$: INC     R4          ;;DON'T SUPPRESS ANYMORE 0'S
7850 057536 052703 000060          BIS     #'0, R3          ;;MAKE THIS DIGIT ASCII
7851 057542 052703 000040          5$: BIS     #' , R3      ;;MAKE ASCII IF NOT ALREADY
7852 057546 110337 057612          MOVB     R3, 8$          ;;SAVE FOR TYPING
7853 057552 10 401 057612          TYPE     8$          ;;GO TYPE THIS DIGIT
7854 057556 105337 057614          7$: DECB     $OCNT          ;;COUNT BY 1
7855 057562 003347          BGT     2$          ;;BR IF MORE TO DO
7856 057564 002402          BLT     6$          ;;BR IF DONE
    
```



```

7857 057566 005204          INC      R4          ;; INSURE LAST DIGIT ISN'T A BLANK
7858 057570 000744          BR       2$          ;; GO DO THE LAST DIGIT
7859 057572 012605          6$: MOV    (SP)+,R5    ;; RESTORE R5
7860 057574 012604          MOV    (SP)+,R4    ;; RESTORE R4
7861 057576 012603          MOV    (SP)+,R3    ;; RESTORE R3
7862 057600 016666 000002 000004 MOV    2(SP),4(SP)  ;; SET THE STACK FOR RETURNING
7863 057606 012616          MOV    (SP)+,(SP)
7864 057610 000002          RTI          ;; RETURN
7865 057612 000          8$: .BYTE 0          ;; STORAGE FOR ASCII DIGIT
7866 057613 000          .BYTE 0          ;; TERMINATOR FOR TYPE ROUTINE
7867 057614 000          $OCNT: .BYTE 0     ;; OCTAL DIGIT COUNTER
7868 057615 000          $OFILL: .BYTE 0    ;; ZERO FILL SWITCH
7869 057616 000000          $OMODE: .WORD 0    ;; NUMBER OF DIGITS TO TYPE
7870          .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7871
7872          ;; *****
7873          ;; *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7874          ;; *SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7875          ;; *NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7876          ;; *BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
7877          ;; *REPLACED WITH SPACES.
7878          ;; *CALL:
7879          ;; *      MOV    NUM,-(SP)          ;; PUT THE BINARY NUMBER ON THE STACK
7880          ;; *      TYPDS          ;; GO TO THE ROUTINE
7881
7882          $TYPDS:
7883          MOV    R0,-(SP)          ;; PUSH R0 ON STACK
7884          MOV    R1,-(SP)          ;; PUSH R1 ON STACK
7885          MOV    R2,-(SP)          ;; PUSH R2 ON STACK
7886          MOV    R3,-(SP)          ;; PUSH R3 ON STACK
7887          MOV    R5,-(SP)          ;; PUSH R5 ON STACK
7888          MOV    #20200,-(SP)     ;; SET BLANK SWITCH AND SIGN
7889          MOV    20(SP),R5        ;; GET THE INPUT NUMBER
7890          BPL    1$              ;; BR IF INPUT IS POS.
7891          NEG    R5              ;; MAKE THE BINARY NUMBER POS.
7892          MOVVB #'-,(SP)         ;; MAKE THE ASCII NUMBER NEG.
7893          CLR    R0              ;; ZERO THE CONSTANTS INDEX
7894          MOV    # $DBLK,R3       ;; SETUP THE OUTPUT POINTER
7895          MOVVB #' ,(R3)+        ;; SET THE FIRST CHARACTER TO A BLANK
7896          CLR    R2              ;; CLEAR THE BCD NUMBER
7897          MOV    $DTBL(R0),R1     ;; GET THE CONSTANT
7898          SUB    R1,R5           ;; FORM THIS BCD DIGIT
7899          BLT    4$              ;; BR IF DONE
7900          INC    R2              ;; INCREASE THE BCD DIGIT BY 1
7901          BR     3$
7902          ADD    R1,R5           ;; ADD BACK THE CONSTANT
7903          TST    R2              ;; CHECK IF BCD DIGIT=0
7904          BNE    5$             ;; FALL THROUGH IF 0
7905          TSTB  (SP)            ;; STILL DOING LEADING 0'S?
7906          BMI    7$             ;; BR IF YES
7907          ASLB  (SP)            ;; MSD?
7908          BCC   6$             ;; BR IF NO
7909          MOVVB 1(SP),-1(R3)     ;; YES--SET THE SIGN
7910          BIS   #'0,R2          ;; MAKE THE BCD DIGIT ASCII
7911          BIS   #' ,R2          ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
7912          MOVVB R2,(R3)+        ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
    
```

L13

```

7913 057742 005720          TST      (R0)+      ;; JUST INCREMENTING
7914 057744 020027 000010    CMP      R0,#10    ;; CHECK THE TABLE INDEX
7915 057750 002746          BLT      2$        ;; GO DO THE NEXT DIGIT
7916 057752 003002          BGT      8$        ;; GO TO EXIT
7917 057754 010502          MOV      R5,R2     ;; GET THE LSD
7918 057756 000764          BR       6$        ;; GO CHANGE TO ASCII
7919 057760 105726          8$: TSTB     (SP)+    ;; WAS THE LSD THE FIRST NON-ZERO?
7920 057762 100003          BPL      9$        ;; BR IF NO
7921 057764 116663 177777 177776  MOVB     -1(SP),-2(R3) ;; YES--SET THE SIGN FOR TYPING
7922 057772 105013          9$: CLRB     (R3)    ;; SET THE TERMINATOR
7923 057774 012605          MOV      (SP)+,R5  ;; POP STACK INTO R5
7924 057776 012603          MOV      (SP)+,R3  ;; POP STACK INTO R3
7925 060000 012602          MOV      (SP)+,R2  ;; POP STACK INTO R2
7926 060002 012601          MOV      (SP)+,R1  ;; POP STACK INTO R1
7927 060004 012600          MOV      (SP)+,R0  ;; POP STACK INTO R0
7928 060006 104401 060034          TYPE     $DBLK     ;; NOW TYPE THE NUMBER
7929 060012 016666 000002 000004  MOV      2(SP),4(SP) ;; ADJUST THE STACK
7930 060020 012616          MOV      (SP)+,(SP)
7931 060022 000002          RTI                          ;; RETURN TO USER
7932 060024 023420          $DTBL: 10000.
7933 060026 001750          1000.
7934 060030 000144          100.
7935 060032 000012          10.
7936 060034 000004          $DBLK: .BLKW 4
7937                                     .SBTTL TTY INPUT ROUTINE
7938
7939                                     ;;*****
7940                                     .ENABL LSB
7941
7942                                     ;;*****
7943                                     ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
7944                                     ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
7945                                     ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
7946                                     ;;*WHEN OPERATING IN TTY FLAG MODE.
7947 060044 022737 000176 001140  $CKSWR: CMP      #SWREG,SWR  ;; IS THE SOFT-SWR SELECTED?
7948 060052 001074          BNE      15$      ;; BRANCH IF NO
7949 060054 105777 121064          TSTB     2$TKS    ;; CHAR THERE?
7950 060060 100071          BPL      15$      ;; IF NO, DON'T WAIT AROUND
7951 060062 117746 121060          MOVB     2$TKB,-(SP) ;; SAVE THE CHAR
7952 060066 042716 177600          BIC      #1C17,(SP) ;; STRIP-OFF THE ASCII
7953 060072 022726 000007          CMP      #7,(SP)+  ;; IS IT A CONTROL G?
7954 060076 001062          BNE      15$      ;; NO, RETURN TO USER
7955 060100 123727 001134 000001  CMPB     $AUTOB,#1  ;; ARE WE RUNNING IN AUTO-MODE?
7956 060106 001456          BEQ      15$      ;; BRANCH IF YES
7957
7958 060110 104401 060717          SGT$WR: TYPE     , $CNTLG  ;; ECHO THE CONTROL-G (↑G)
7959 060114 104401 060724          TYPE     $MSWR     ;; TYPE CURRENT CONTENTS
7960 060120 013746 000176          MOV      SWREG,-(SP) ;; SAVE SWREG FOR TYPEOUT
7961 060124 104402          TYPOC    ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
7962 060126 104401 060735          TYPE     , $MNEW   ;; PROMPT FOR NEW SWR
7963 060132 005046          19$: CLR      -(SP)  ;; CLEAR COUNTER
7964 060134 005046          CLR      -(SP)    ;; THE NEW SWR
7965 060136 105777 121002          7$: TSTB     2$TKS    ;; CHAR THERE?
7966 060142 100375          BPL      7$        ;; IF NOT TRY AGAIN
7967
7968 060144 117746 120776          MOVB     2$TKB,-(SP) ;; PICK UP CHAR

```

M13

```

7969 060150 042716 177600          BIC      #1C177,(SF)      ;;MAKE IT 7-BIT ASCII
7970
7971
7972
7973 060154 021627 000025          9$:      CMP      (SP),#25      ;; IS IT A CONTROL-U?
7974 060160 001005                    BNE      10$              ;; BRANCH IF NOT
7975 060162 104401 060712          TYPE    ,SCNTLU        ;; YES, ECHO CONTROL-U (↑U)
7976 060166 062706 000006          20$:    ADD      #6,SP        ;; IGNORE PREVIOUS INPUT
7977 060172 000757                    BR       19$              ;; LET'S TRY IT AGAIN
7978
7979
7980 060174 021627 000015          10$:    CMP      (SP),#15      ;; IS IT A <CR>?
7981 060200 001022                    BNE      16$              ;; BRANCH IF NO
7982 060202 005766 000004          TST     4(SP)           ;; YES, IS IT THE FIRST CHAR?
7983 060206 001403                    BEQ     11$              ;; BRANCH IF YES
7984 060210 016677 000002 120722  MOV     2(SP),@SWR      ;; SAVE NEW SWR
7985 060216 062706 000006          11$:    ADD      #6,SP        ;; CLEAR UP STACK
7986 060222 104401 001211          14$:    TYPE    $CRLF        ;; ECHO <CR> AND <LF>
7987 060226 123727 001135 000001  CMPB   $INTAG,#1       ;; RE-ENABLE TTY KBD INTERRUPTS?
7988 060234 001003                    BNE     15$              ;; BRANCH IF NOT
7989 060236 012777 000100 120700  MOV     #100,@STKS     ;; RE-ENABLE TTY KBD INTERRUPTS
7990 060244 000002          15$:    RTI                    ;; RETURN
7991 060246 004737 057322          16$:    JSR     PC,$TYPEC     ;; ECHO CHAR
7992 060252 021627 000060          CMP     (SP),#60       ;; CHAR < 0?
7993 060256 002420                    BLT     18$              ;; BRANCH IF YES
7994 060260 021627 000067          CMP     (SP),#67       ;; CHAR > 7?
7995 060264 003015                    BGT     18$              ;; BRANCH IF YES
7996 060266 042726 000060          BIC     #60,(SP)+      ;; STRIP-OFF ASCII
7997 060272 005766 000002          TST     2(SP)           ;; IS THIS THE FIRST CHAR
7998 060276 001403                    BEQ     17$              ;; BRANCH IF YES
7999 060300 006316                    ASL     (SP)            ;; NO, SHIFT PRESENT
8000 060302 006316                    ASL     (SP)            ;; CHAR OVER TO MAKE
8001 060304 006316                    ASL     (SP)            ;; ROOM FOR NEW ONE.
8002 060306 005266 000002          17$:    INC     2(SP)         ;; KEEP COUNT OF CHAR
8003 060312 056616 177776          BIS     -2(SP),(SP)    ;; SET IN NEW CHAR
8004 060316 000707                    BR      7$              ;; GET THE NEXT ONE
8005 060320 104401 001210          18$:    TYPE    $QUES        ;; TYPE ?<CR><LF>
8006 060324 000720                    BR      20$              ;; SIMULATE CONTROL-U
8007 .DSABL  LSB
8008
8009
8010
8011 *****
8012 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
8013 *CALL:
8014 * RDCHR                                ;; INPUT A SINGLE CHARACTER FROM THE TTY
8015 * RETURN HERE                          ;; CHARACTER IS ON THE STACK
8016 *                                       ;; WITH PARITY BIT STRIPPED OFF
8017
8018 SRDCHR: MOV      (SP),-(SP)           ;; PUSH DOWN THE PC
8019 060330 016666 000004 000002  MOV     4(SP),2(SP)    ;; SAVE THE PS
8020 060336 105777 120602          1$:    TSTB   @STKS        ;; WAIT FOR
8021 060342 100375                    BPL     1$              ;; A CHARACTER
8022 060344 117766 120576 000004  MOVB   @STKB,4(SP)    ;; READ THE TTY
8023 060352 042766 177600 000004  BIC     #1C<177>,4(SP) ;; GET RID OF JUNK IF ANY
8024 060360 026627 000004 000023  CMP     4(SP),#23     ;; IS IT A CONTROL-S?

```

```

8025 060366 001013          BNE      3$          ;; BRANCH IF NO
8026 060370 105777 120550 23:  TSTB     3$STKS     ;; WAIT FOR A CHARACTER
8027 060374 100375          BPL      2$          ;; LOOP UNTIL ITS THERE
8028 060376 117746 120544  MOVB     3$STKB, -(SP) ;; GET CHARACTER
8029 060402 042716 177600  BIC      #1C177, (SP) ;; MAKE IT 7-BIT ASCII
9030 060406 022627 000021  CMP      (SP)+, #21   ;; IS IT A CONTROL-G?
8031 060412 001366          SNE      2$          ;; IF NOT DISCARD IT
8032 060414 000750          BR       1$          ;; YES, RESUME
8033 060416 026627 000004 000140 3$:  CMP      4(SP), #140  ;; IS IT UPPER CASE?
8034 060424 0C2407          BLT      4$          ;; BRANCH IF YES
8035 060426 026627 000004 000175  CMP      4(SP), #175  ;; IS IT A SPECIAL CHAR?
8036 060434 003003          BCT      4$          ;; BRANCH IF YES
8037 060436 042766 000040 000004  BIC      #40, 4(SP)   ;; MAKE IT UPPER CASE
8038 060444 000002          4$:  RTI          ;; GO BACK TO USER
8039                                     ;; *****
8040                                     ;; *THIS ROUTINE WILL INPUT A STRING FROM THE TTY
8041                                     ;; *CALL:
8042                                     ;; *
8043                                     ;; *   RDLIN          ;; INPUT A STRING FROM THE TTY
8044                                     ;; *   RETURN HERE   ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
8045                                     ;; *                                     ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
8046 060446 010346          $RDLIN: MOV      R3, -(SP)   ;; SAVE R3
8047 060450 005046          CLR      -(SP)      ;; CLEAR THE RUBOUT KEY
8048 060452 012703 060702 1$:  MOV      #1$TTYIN, R3 ;; GET ADDRESS
8049 060456 022703 060712 2$:  CMP      #1$TTYIN+8., R3 ;; BUFFER FULL?
8050 060462 101456          BLOS     4$          ;; BR IF YES
8051 060464 104410          RDCHR   ;; GO READ ONE CHARACTER FROM THE TTY
8052 060466 112613          MOVB     (SP)+, (R3) ;; GET CHARACTER
8053 060470 122713 000177 10$:  CMPB     #177, (R3)   ;; IS IT A RUBOUT
8054 060474 001022          BNE      5$          ;; BR IF NO
8055 060476 005716          TST      (SP)       ;; IS THIS THE FIRST RUBOUT?
8056 060500 001007          BNE      6$          ;; BR IF NO
8057 060502 112737 000134 060700  MOVB     #' \, 9$    ;; TYPE A BACK SLASH
8058 060510 104401 060700          TYPE     9$
8059 060514 012716 177777          MOV      #-1, (SP)  ;; SET THE RUBOUT KEY
8060 060520 005303          DEC      R3         ;; BACKUP BY ONE
8061 060522 020327 060702 6$:  CMP      R3, #1$TTYIN ;; STACK EMPTY?
8062 060526 103434          BLO      4$          ;; BR IF YES
8063 060530 111337 060700          MOVB     (R3), 9$   ;; SETUP TO TYPEOUT THE DELETED CHAR.
8064 060534 104401 060700          TYPE     9$
8065 060540 000746          BR       2$          ;; GO READ ANOTHER CHAR.
8066 060542 005716          5$:  TST      (SP)       ;; RUBOUT KEY SET?
8067 060544 001406          BEQ      7$          ;; BR IF NO
8068 060546 112737 000134 060700  MOVB     #' \, 9$    ;; TYPE A BACK SLASH
8069 060554 104401 060700          TYPE     9$
8070 060560 005016          CLR      (SP)       ;; CLEAR THE RUBOUT KEY
8071 060562 122713 000025 7$:  CMPB     #25, (R3)   ;; IS CHARACTER A CTRL U?
8072 060566 001003          BNE      8$          ;; BR IF NO
8073 060570 104401 060712          TYPE     , $CNTLU   ;; TYPE A CONTROL "U"
8074 060574 000726          BR       1$          ;; GO START OVER
8075 060576 122713 000022 8$:  CMPB     #22, (R3)   ;; IS CHARACTER A "↑R"?
8076 060602 001011          BNE      3$          ;; BRANCH IF NO
8077 060604 105013          CLRB    (R3)        ;; CLEAR THE CHARACTER
8078 060606 104401 001211          TYPE     , $CRLF    ;; TYPE A "CR" & "LF"
8079 060612 104401 060702          TYPE     , $TTYIN   ;; TYPE THE INPUT STRING
8080 060616 000717          BR       2$          ;; GO PICKUP ANOTHER CHACTER
    
```

```

8081 060620 104401 001210 4$: TYPE $QUES ;;TYPE A '?'
8082 060624 000712 BR 15 ;;CLEAR THE BUFFER AND LOOP
8083 060626 111337 060700 3$: MOV8 (R3),95 ;;ECHO THE CHARACTER
8084 060632 104401 060700 TYPE 95
8085 060636 122723 000015 CMPS #15,(R3)+ ;;CHECK FOR RETURN
8086 060642 001305 BNE 25 ;;LOOP IF NOT RETURN
8087 060644 105063 177777 CLR8 -1(R3) ;;CLEAR RETURN (THE 15)
8088 060650 104401 001212 TYPE $LF ;;TYPE A LINE FEED
8089 060654 005726 TST (SP)+ ;;CLEAN RUBOUT KEY FROM THE STACK
8090 060656 012603 MOV (SP)+,R3 ;;RESTORE R3
8091 060660 011646 MOV (SP)-,(SP) ;;ADJUST THE STACK AND PUT ADDRESS OF THE
8092 060662 016666 000004 000002 MOV 4(SP),2(SP) ;; FIRST ASCII CHARACTER ON IT
8093 060670 012766 060702 000004 MOV #TTYIN,4(SP)
8094 060676 000002 RTI ;;RETURN
8095 060700 000 9$: .BYTE 0 ;;STORAGE FOR ASCII CHAR. TO TYPE
8096 060701 000 .BYTE 0 ;;TERMINATOR
8097 060702 000010 $TTYIN: .BLKB 8. ;;RESERVE 8 BYTES FOR TTY INPUT
8098 060712 052536 005015 000 $CNTLU: .ASCIZ /?U/<15><12> ;;CONTROL "U"
8099 060717 136 006507 000012 $CNTLG: .ASCIZ /?G/<15><12> ;;CONTROL "G"
9100 060724 005015 053523 020122 $MSWR: .ASCIZ <15><12>/SWR = /
8101 060732 020075 000
8102 060735 040 047040 053505 $MNEW: .ASCIZ / NEW = /
8103 060742 036440 000040
8104
8105 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
8106
8107 ;;*****
8108 ;;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
8109 ;;*CHANGE IT TO BINARY.
8110 ;;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
8111 ;;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
8112 ;;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
8113 ;;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
8114 ;;*CALL:
8115 ;;* RDOCT ;;READ AN OCTAL NUMBER
8116 ;;* RETURN HERE ;;LOW ORDER BITS ARE ON TOP OF THE STACK
8117 ;;* ;;HIGH ORDER BITS ARE IN $HIOCT
8118 060746 011646 000004 000002 $RDOCT: MOV (SP)-,(SP) ;;PROVIDE SPACE FOR THE
8119 060750 016666 MOV 4(SP),2(SP) ;;INPUT NUMBER
8120 060756 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
8121 060760 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
8122 060762 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
8123 060764 104411 1$: RDLIN ;;READ AN ASCII LINE
8124 060766 012600 MOV (SP)+,R0 ;;GET ADDRESS OF 1ST CHARACTER
8125 060770 010037 061074 MOV R0,$5 ;;AND SAVE IT
8126 060774 005001 CLR R1 ;;CLEAR DATA WORD
8127 060776 005002 CLR R2
8128 061000 112046 2$: MOV8 (R0)+,-(SP) ;;PICKUP THIS CHARACTER
8129 061002 001420 BEQ 3$ ;;IF ZERO GET OUT
8130 061004 122716 000060 CMPB #'0,(SP) ;;MAKE SURE THIS CHARACTER
8131 061010 003026 BGT 4$ ;;IS AN OCTAL DIGIT
8132 061012 122716 000067 CMPB #'7,(SP)
8133 061016 002423 BLT 4$
8134 061020 006301 ASL R1 ;;*2
8135 061022 006102 ROL R2
8136 061024 006301 ASL R1 ;;*4

```

```

8137 061026 006102          RUL      R2
8138 061030 006301          ASL      R1          ;;*8
8139 061032 006102          ROL      R2
8140 061034 042716 177770   BIC      #1C7,(SP)    ;;STRIP THE ASCII JUNK
8141 061040 062601          ADD      (SP)+,R1    ;;ADD IN THIS DIGIT
8142 061042 000756          BR       2$          ;;LOOP
8143 061044 005726          3$:     TST      (SP)+    ;;CLEAN TERMINATOR FROM STACK
8144 061046 010166 000012   MOV      R1,12,SP    ;;SAVE THE RESULT
8145 061052 010237 061104   MOV      R2,$HIOCT
8146 061056 012602          MOV      (SP)+,R2    ;;POP STACK INTO R2
8147 061060 012601          MOV      (SP)+,R1    ;;POP STACK INTO R1
8148 061062 012600          MOV      (SP)+,R0    ;;POP STACK INTO R0
8149 061064 000002          RTI
8150 061066 005726          4$:     TST      (SP)+    ;;CLEAN PARTIAL FROM STACK
8151 061070 105010          CLRB     (R0)        ;;SET A TERMINATOR
8152 061072 104401          TYPE
8153 061074 000000          5$:     .WORD    0      ;;TYPE UP THRU THE BAD CHAR.
8154 061076 104401 001210   TYPE     $QUES      ;;?" "CR" & "LF"
8155 061102 000730          BR       1$          ;;TRY AGAIN
8156 061104 000000          $HIOCT: .WORD    0    ;;HIGH ORDER BITS GO HERE
8157
8158 .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

```

```

*****
; *SAVE R0-R5
; *CALL:
; * SAVREG
; *UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
; *
; *TOP----(+16)
; * +2----(+18)
; * +4----R5
; * +6----R4
; * +8----R3
; *+10----R2
; *+12----R1
; *+14----R0

```

```

8174 061106          $SAVREG:
8175 061106 010046          MOV      R0,-(SP)    ;;PUSH R0 ON STACK
8176 061110 010146          MOV      R1,-(SP)    ;;PUSH R1 ON STACK
8177 061112 010246          MOV      R2,-(SP)    ;;PUSH R2 ON STACK
8178 061114 010346          MOV      R3,-(SP)    ;;PUSH R3 ON STACK
8179 061116 010446          MOV      R4,-(SP)    ;;PUSH R4 ON STACK
8180 061120 010546          MOV      R5,-(SP)    ;;PUSH R5 ON STACK
8181 061122 016646 000022   MOV      22(SP),-(SP) ;;SAVE PS OF MAIN FLOW
8182 061126 016646 000022   MOV      22(SP),-(SP) ;;SAVE PC OF MAIN FLOW
8183 061132 016646 000022   MOV      22(SP),-(SP) ;;SAVE PS OF CALL
8184 061136 016646 000022   MOV      22(SP),-(SP) ;;SAVE PC OF CALL
8185 061142 000002          RTI

```

```

8186
8187 ;*RESTORE R0-R5
8188 ;*CALL:
8189 ;* RESREG
8190 $RESREG:
8191 061144 012666 000022   MOV      (SP)+,22(SP) ;;RESTORE PC OF CALL
8192 061150 012666 000022   MOV      (SP)+,22(SP) ;;RESTORE PS OF CALL

```

```

8193 061154 012666 000022      MOV      (SP)+,22(SP)      ;;RESTORE PC OF MAIN FLOW
8194 061160 012666 000022      MOV      (SP)+,22(SP)      ;;RESTORE PS OF MAIN FLOW
8195 061164 012605              MOV      (SP)+,R5          ;;POP STACK INTO R5
8196 061166 012604              MOV      (SP)+,R4          ;;POP STACK INTO R4
8197 061170 012603              MOV      (SP)+,R3          ;;POP STACK INTO R3
8198 061172 012602              MOV      (SP)+,R2          ;;POP STACK INTO R2
8199 061174 012601              MOV      (SP)+,R1          ;;POP STACK INTO R1
8200 061176 012600              MOV      (SP)+,R0          ;;POP STACK INTO R0
8201 061200 000002              RTI

```

.SBTTL POWER DOWN AND UP ROUTINES

POWER DOWN ROUTINE

```

8206 061202 017737 117732 002026 $PWRDN: MOV      @SWR,SAVSWR      ;;SAVE SWITCH REGISTER
8207 061210 012737 061230 000024      MOV      @SPWRUP,PWRVEC    ;;SET UP VECTOR
8208 061216 012737 000340 000026      MOV      @PR7,PWRVEC+2
8209 061224 000000              HALT
8210 061226 000776              BR       -2      ;HANG UP

```

POWER UP ROUTINE

```

8213 061230 005037 061320      $PWRUP: CLR      $PWRCT      ;;LOAD WAIT COUNT
8214 061234 012737 000144 061322      MOV      #100,$PWRCT+2
8215 061242 005237 061320      1$: INC      $PWRCT      ;;WAIT FOR TELETYPE
8216 061246 001375              BNE     1$
8217 061250 005337 061322      DEC      $PWRCT+2
8218 061254 001372              BNE     1$
8219 061256 012737 061202 000024      MOV      @SPWRDN,PWRVEC    ;;SET UP FOR POWER DOWN VECTOR
8220 061264 012737 000340 000026      MOV      @PR7,PWRVEC+2
8221 061272 012706 001100              MOV      @STACK,SP        ;;FORCE STACK
8222 061276 104401 061324              TYPE     $POWER           ;;TYPE POWER
8223 061302 004737 055274              JSR     PC,CHKPAR         ;;REINITIALIZE MEMORY CHECK ENABLE
8224 061306 013777 002026 117624      MOV      SAVSWR,@SWR      ;;RESTORE SWITCH REGISTER
8225 061314 000177 117566              JMP     @SLPADR          ;;GO BACK TO LAST TEST
8226 061320 000000 000000      $PWRCT: .WORD 0,0        ;;TELETYPE TIME OUT
8227 061324 047520 042527 000122      $POWER: .ASCIZ /POWER/
8228 061324 000000              .EVEN

```

.SBTTL TRAP DECODER

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

```

8239 061332 010046              $TRAP: MOV      RO,-(SP)      ;;SAVE RO
8240 061334 016600 000002      MOV      2(SP),RO        ;;GET TRAP ADDRESS
8241 061340 005740              TST     -(RO)           ;;BACKUP BY 2
8242 061342 111000              MOV     (RO),RO        ;;GET RIGHT BYTE OF TRAP
8243 061344 006300              ASL     RO              ;;POSITION FOR INDEXING
8244 061346 016000 061366      MOV     $TRPAD(RO),RO    ;;INDEX TO TABLE
8245 061352 000200              RTS     RO              ;;GO TO ROUTINE

```


E14

```

8249          ::THIS IS USE TO HANDLE THE "GETPRI" MACRO
8250
8251 061354 011646          $TRAP2: MOV      (SP),-(SP)      ;;MOVE THE PC DOWN
8252 061356 016666 000004 000002  MOV      4(SP),2(SP)  ;;MOVE THE PSW DOWN
8253 061364 000002          RTI                          ;;RESTORE THE PSW
8254
8255          .SBTTL TRAP TABLE
8256
8257          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
8258          ;*BY THE "TRAP" INSTRUCTION.
8259
8260          :          ROUTINE
8261          :          -----
8262 061366 061354          $TRPAD: .WORD  $TRAP2
8263 061370 057110          $TYPE   ;;CALL=TYPE   TRAP+1(104401) TTY TYPEOUT ROUTINE
8264 061372 057416          $TYPOC  ;;CALL=TYPOC  TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8265 061374 057372          $TYPOS  ;;CALL=TYPOS  TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
8266 061376 057432          $TYPON  ;;CALL=TYPON  TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
8267 061400 057620          $TYPDS  ;;CALL=TYPDS  TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
8268
8269 061402 060114          $GTSWR  ;;CALL=GTSWR  TRAP+6(104406) GET SOFT-SWR SETTING
8270
8271 061404 060044          $CKSWR  ;;CALL=CKSWR  TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
8272 061406 060326          $RDCHR  ;;CALL=RDCHR  TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
8273 061410 060446          $RDLIN  ;;CALL=RDLIN  TRAP+11(104411) TTY TYPEIN STRING ROUTINE
8274 061412 060746          $RDOCT  ;;CALL=RDOCT  TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
8275 061414 061106          $$SAVREG ;;CALL=SAVREG  TRAP+13(104413) SAVE R0-R5 ROUTINE
8276 061416 061144          $RESREG ;;CALL=RESREG  TRAP+14(104414) RESTORE R0-R5 ROUTINE
8277 061420 056134          $COPI$  ;;CALL=COPI$  TRAP+15(104415) INTERNAL LOOP ON ERROR
  
```

					.SBTTL DATA TABLE FOR ERROR PRINT OUT	
8278						
8279						
8280	061422	001220	002024		DT000:	.WORD \$TESTN, TRAPP
8281	061426	001220	001116	001122	DT001:	.WORD \$TESTN, \$ERRPC, \$BDADR
8282	061434	001220	001116	001124	DT002:	.WORD \$TESTN, \$ERRPC, \$GDDAT, \$BDDAT
8283	061442	001126				
8284	061444	001220	001116	002010	DT003:	.WORD \$TESTN, \$ERRPC, CONFIG, \$GDDAT, \$BDDAT
8285	061452	001124	001126			
8286	061456	001220	001116	002014	DT004:	.WORD \$TESTN, \$ERRPC, PREREG, \$GDDAT, \$BDDAT
8287	061464	001124	001126			
8288	061470	001220	001116		DT005:	.WORD \$TESTN, \$ERRPC
8289	061474	001220	001116	002000	DT006:	.WORD \$TESTN, \$ERRPC, RKVEC, \$BDADR
8290	061502	001122				
8291	061504	001220	001116	002020	DT011:	.WORD \$TESTN, \$ERRPC, PRIOR
8292	061512	001220	001116	001740	DT016:	.WORD \$TESTN, \$ERRPC, E.CS1, T.CS1, E.CS2, T.CS2
8293	061520	001700	001750	001710		
8294	061526	001220	001116	001740	DT024:	.WORD \$TESTN, \$ERRPC, E.CS1, T.CS1, E.CS2, T.CS2, SILCNT
8295	061534	001700	001750	001710		
8296	061542	002022				
8297	061544	001220	001116	001740	DT026:	.WORD \$TESTN, \$ERRPC, E.CS1, T.CS1, E.CS2, T.CS2, SILCNT, \$GDDAT, \$BDDAT
8298	061552	001700	001750	001710		
8299	061560	002022	001124	001126		

				.SBTTL DATA FORMAT FOR ERROR PRINT OUT	
8300					
8301					
8302	061566	000001		DF000:	.WORD 1
8303	061570	002	000		.BYTE 2,0
8304	061572	000004		DF001:	.WORD 4,0 ;ERROR 1
8305	061574	000	000		.BYTE 0,0
8306	061576	062261			.WORD DH000A
8307	061600	000	000		.BYTE 0,0
8308	061602	062277			.WORD DH000B
8309	061604	002	000		.BYTE 2,0
8310	061606	062342			.WORD DH001
8311	061610	001	000		.BYTE 1,0
8312	061612	000005		DF002:	.WORD 5,0 ;ERROR 2
8313	061614	000	000		.BYTE 0,0
8314	061616	062261			.WORD DH000A
8315	061620	000	000		.BYTE 0,0
8316	061622	062277			.WORD DH000B
8317	061624	002	000		.BYTE 2,0
8318	061626	062361			.WORD DH002A
8319	061630	000	000		.BYTE 0,0
8320	061632	062400			.WORD DH002B
8321	061634	002	000		.BYTE 2,0
8322	061636	000005		DF003:	.WORD 5,0 ;ERROR 3
8323	061640	000	000		.BYTE 0,0
8324	061642	062261			.WORD DH000A
8325	061644	000	000		.BYTE 0,0
8326	061646	062277			.WORD DH000B
8327	061650	002	000		.BYTE 2,0
8328	061652	062416			.WORD DH003A
8329	061654	000	000		.BYTE 0,0
8330	061656	062445			.WORD DH003B
8331	061660	003	000		.BYTE 3,0
8332	061662	000005		DF004:	.WORD 5,0 ;ERROR 4
8333	061664	000	000		.BYTE 0,0
8334	061666	062261			.WORD DH000A
8335	061670	000	000		.BYTE 0,0
8336	061672	062277			.WORD DH000B
8337	061674	002	000		.BYTE 2,0
8338	061676	062473			.WORD DH004A
8339	061700	000	000		.BYTE 0,0
8340	061702	062522			.WORD DH004B
8341	061704	003	000		.BYTE 3,0
8342	061706	000003		DF005:	.WORD 3,0 ;ERROR 5
8343	061710	000	000		.BYTE 0,0
8344	061712	062261			.WORD DH000A
8345	061714	000	000		.BYTE 0,0
8346	061716	062277			.WORD DH000B
8347	061720	002	000		.BYTE 2,0
8348	061722	000005		DF006:	.WORD 5,0 ;ERROR 6
8349	061724	000	000		.BYTE 0,0
8350	061726	062261			.WORD DH000A
8351	061730	000	000		.BYTE 0,0
8352	061732	062277			.WORD DH000B
8353	061734	002	000		.BYTE 2,0
8354	061736	062550			.WORD DH006A
8355	061740	000	000		.BYTE 0,0

8356	061742	062567			.WORD	DH006B	
8357	061744	002	000		.BYTE	2,0	
8358	061746	000004		DF011:	.WORD	4,0	;ERROR 11
8359	061750	000	000		.BYTE	0,0	
8360	061752	062261			.WORD	DH000A	
8361	061754	000	000		.BYTE	0,0	
8362	061756	062277			.WORD	DH000B	
8363	061760	002	000		.BYTE	2,0	
8364	061762	062606			.WORD	DH011	
8365	061764	001	000		.BYTE	1,0	
8366	061766	000005		DF016:	.WORD	5	;ERROR 16
8367	061770	000	000		.BYTE	0,0	
8368	061772	062261			.WORD	DH000A	
8369	061774	000	000		.BYTE	0,0	
8370	061776	062277			.WORD	DH000B	
8371	062000	002	000		.BYTE	2,0	
8372	062002	062550			.WORD	DH006A	
8373	062004	000	000		.BYTE	0,0	
8374	062006	062670			.WORD	DH016B	
8375	062010	004	000		.BYTE	4,0	
8376	062012	000005		DF024:	.WORD	5	;ERROR 24
8377	062014	000	000		.BYTE	0,0	
8378	062016	062261			.WORD	DH000A	
8379	062020	000	000		.BYTE	0,0	
8380	062022	062277			.WORD	DH000B	
8381	062024	002	000		.BYTE	2,0	
8382	062026	062726			.WORD	DH024A	
8383	062030	000	000		.BYTE	0,0	
8384	062032	062773			.WORD	DH024B	
8385	062034	005	000		.BYTE	5,0	
8386	062036	000005		DF026:	.WORD	5	;ERROR 26
8387	062040	000	000		.BYTE	0,0	
8388	062042	062261			.WORD	DH000A	
8389	062044	000	000		.BYTE	0,0	
8390	062046	062277			.WORD	DH000B	
8391	062050	002	000		.BYTE	2,0	
8392	062052	063040			.WORD	DH026A	
8393	062054	000	000		.BYTE	0,0	
8394	062056	063127			.WORD	DH026B	
8395	062060	007	000		.BYTE	7,0	

```

8396 .SBTTL ASCII MESSAGES
8397
8398 062062 005015 045522 030466 OPR001: .ASCIZ <15><12>/RK611 BUS ADDRESS ( /
8399 062070 020061 052502 020123
8400 062076 042101 051104 051505
8401 062104 020123 020050 000
8402 062111 040 020051 020075 OPR002: .ASCIZ / ) = /
8403 062116 000
8404 062117 122 033113 030461 OPR003: .ASCIZ /RK611 VECTOR ADDRESS ( /
8405 062124 053040 041505 047524
8406 062132 020122 042101 051104
8407 062140 051505 020123 020050
8408 062146 000
8409 062147 122 033113 030461 OPR004: .ASCIZ /RK611 PRIORITY ( /
8410 062154 050040 044522 051117
8411 062162 052111 020131 020050
8412 062170 000
8413 062171 040 000040 SPACE2: .ASCIZ / /
8414 062174 005015 051120 043517 ABORT: .ASCIZ <15><12>/PROGRAM ABORTED BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
8415 062202 040522 020115 041101
8416 062210 051117 042524 020104
8417 062216 042502 040503 051525
8418 062224 020105 051105 047522
8419 062232 020122 044124 042522
8420 062240 044123 046117 020104
8421 062246 054105 042503 042105
8422 062254 042105 005015 000

```

.SBTTL DATA HEADERS										
8423										
8424										
8425	062261	124	051505	020124	DH000A:	.ASCIZ	/TEST	ERROR/		
8426	062266	020040	042440	051122						
8427	062274	051117	000							
8428	062277	116	046525	020040	DH000B:	.ASCIZ	/NUM	PC/		
8429	062304	020040	050040	000103						
8430	062312	042524	052123	020040	DH000C:	.ASCII	/TEST	TRAP/<15><12>		
8431	062320	020040	051124	050101						
8432	062326	005015								
8433	062330	052516	020115	020040		.ASCIZ	/NUM	PC/		
8434	062336	050040	000103							
8435	062342	047125	041111	051525	DH001:	.ASCIZ	/UNIBUS	ADDRESS/		
8436	062350	040440	042104	042522						
8437	062356	051523	000							
8438	062361	105	050130	041505	DH002A:	.ASCIZ	/EXPECT	ACTUAL/		
8439	062366	020124	040440	052103						
8440	062374	040525	000114							
8441	062400	040526	053114	020105	DH002B:	.ASCIZ	/VALVE	VALUE/		
8442	062406	020040	040526	052514						
8443	062414	000105								
8444	062416	040526	052514	020105	DH003A:	.ASCIZ	/VALUE	EXPECT	ACTUAL/	
8445	062424	020040	054105	042520						
8446	062432	052103	020040	041501						
8447	062440	052524	046101	000						
8448	062445	127	044522	052124	DH003B:	.ASCIZ	/WRITTEN	VALVE	VALUE/	
8449	062452	047105	053040	046101						
8450	062460	042526	020040	053040						
8451	062466	046101	042525	000						
8452	062473	120	042522	020126	DH004A:	.ASCIZ	/PREV	EXPECT	ACTUAL/	
8453	062500	020040	042440	050130						
8454	062506	041505	020124	040440						
8455	062514	052103	040525	000114						
8456	062522	040526	052514	020105	DH004B:	.ASCIZ	/VALUE	VALVE	VALVE/	
8457	062530	020040	040526	053114						
8458	062536	020105	020040	040526						
8459	062544	053114	000105							
8460	062550	054105	042520	052103	DH006A:	.ASCIZ	/EXPECT	ACTUAL/		
8461	062556	020040	041501	052524						
8462	062564	046101	000							
8463	062567	126	041505	047524	DH006B:	.ASCIZ	/VECTOR	VECTOR/		
8464	062574	020122	053040	041505						
8465	062602	047524	000122							
8466	062606	051120	041517	051505	DH011:	.ASCIZ	/PROCESSOR	PRIORITY/		
8467	062614	047523	020122	051120						
8468	062622	047511	044522	054524						
8469	062630	000								
8470	062631	105	050130	041505	DH016A:	.ASCIZ	/EXPECT	ACTUAL	EXPECT	ACTUAL/
8471	062636	020124	040440	052103						
8472	062644	040525	020114	042440						
8473	062652	050130	041505	020124						
8474	062660	040440	052103	040525						
8475	062666	000114								
8476	062670	045522	051503	020061	DH016B:	.ASCIZ	/RKCS1	RKCS1	RKCS2	RKCS2/
8477	062676	020040	045522	051503						
8478	062704	020061	020040	045522						

8514					.SBTTL	ERROR MESSAGES
8515						
8516	063214	047125	054105	042520	EM000:	.ASCIZ /UNEXPECTED MEMORY PARITY ENABLE TRAP/
8517	063222	052103	042105	046440		
8518	063230	046505	051117	020131		
8519	063236	040520	044522	054524		
8520	063244	042440	040516	046102		
8521	063252	020105	051124	050101		
8522	063260	000				
8523	063261	116	047117	042455	EM1:	.ASCIZ /NON-EXISTENT MEMORY WHEN ACCESSING RK611 REG/
8524	063266	044530	052123	047105		
8525	063274	020124	042515	047515		
8526	063302	054522	053440	042510		
8527	063310	020116	041501	042503		
8528	063316	051523	047111	020107		
8529	063324	045522	030466	020061		
8530	063332	042522	000107			
8531	063336	052101	042524	050115	EM2:	.ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A UNIBUS INIT/
8532	063344	044524	043516	052040		
8533	063352	020117	046103	040505		
8534	063360	020122	045522	030466		
8535	063366	020061	044527	044124		
8536	063374	040440	052440	044516		
8537	063402	052502	020123	047111		
8538	063410	052111	000			
8539	063413	101	052124	046505	EM3:	.ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR/
8540	063420	052120	047111	020107		
8541	063426	047524	041440	042514		
8542	063434	051101	051040	033113		
8543	063442	030461	053440	052111		
8544	063450	020110	020101	047503		
8545	063456	052116	047522	046114		
8546	063464	051105	041440	042514		
8547	063472	051101	000			
8548	063475	101	052124	046505	EM4:	.ASCIZ /ATTEMPTING TO WRITE BUS ADD/
8549	063502	052120	047111	020107		
8550	063510	047524	053440	044522		
8551	063516	042524	041040	051525		
8552	063524	040440	042104	000		
8553	063531	101	052124	046505	EM5:	.ASCIZ /ATTEMPTING TO CLEAR BUS ADD WITH A UNIBUS INIT/
8554	063536	052120	047111	020107		
8555	063544	047524	041440	042514		
8556	063552	051101	041040	051525		
8557	063560	040440	042104	053440		
8558	063566	052111	020110	020101		
8559	063574	047125	041111	051525		
8560	063602	044440	044516	000124		
8561	063610	052101	042524	050115	EM6:	.ASCII /ATTEMPTING TO CLEAR BUS ADD/
8562	063616	044524	043516	052040		
8563	063624	020117	046103	040505		
8564	063632	020122	052502	020123		
8565	063640	042101	104			
8566	063643	127	052111	020110		.ASCIZ /WITH A CONTROLLER CLEAR/
8567	063650	020101	047503	052116		
8568	063656	047522	046114	051105		
8569	063664	041440	042514	051101		

8570	063672	000			
8571	063673	101	052124	046505	EM7: .ASCIZ /ATTEMPTING TO WRITE WORD COUNT/
8572	063700	052120	047111	020107	
8573	063706	047524	053440	044522	
8574	063714	042524	053440	051117	
8575	063722	020104	047503	047125	
8576	063730	000124			
8577	063732	052101	042524	050115	EM8: .ASCIZ /ATTEMPTING TO WRITE DISK ADD/
8578	063740	044524	043516	052040	
8579	063746	020117	051127	052111	
8580	063754	020105	044504	045523	
8581	063762	040440	042104	000	
8582	063767	101	052124	046505	EM9: .ASCIZ /ATTEMPTING TO WRITE SPARE REG/
8583	063774	052120	047111	020107	
8584	064002	047524	053440	044522	
8585	064010	042524	051440	040520	
8586	064016	042522	051040	043505	
8587	064024	000			
8588	064025	101	052124	046505	EM10: .ASCIZ /ATTEMPTING TO WRITE DRIVE STATUS REG/
8589	064032	052120	047111	020107	
8590	064040	047524	053440	044522	
8591	064046	042524	042040	044522	
8592	064054	042526	051440	040524	
8593	064062	052524	020123	042522	
8594	064070	000107			
8595	064072	052101	042524	050115	EM11: .ASCIZ /ATTEMPTING TO WRITE ERROR REG/
8596	064100	044524	043516	052040	
8597	064106	020117	051127	052111	
8598	064114	020105	051105	047522	
8599	064122	020122	042522	000107	
8600	064130	052101	042524	050115	EM12: .ASCIZ /ATTEMPTING TO WRITE MAINT.REG 2/
8601	064136	044524	043516	052040	
8602	064144	020117	051127	052111	
8603	064152	020105	040515	047111	
8604	064160	027124	042522	020107	
8605	064166	000062			
8606	064170	052101	042524	050115	EM13: .ASCIZ /ATTEMPTING TO WRITE MAINT REG 3/
8607	064176	044524	043516	052040	
8608	064204	020117	051127	052111	
8609	064212	020105	040515	047111	
8610	064220	020124	042522	020107	
8611	064226	000063			
8612	064230	052101	042524	050115	EM14: .ASCIZ /ATTEMPTING TO WRITE ECC PATTERN/
8613	064236	044524	043516	052040	
8614	064244	020117	051127	052111	
8615	064252	020105	041505	020103	
8616	064260	040520	052124	051105	
8617	064266	000116			
8618	064270	052101	042524	050115	EM15: .ASCIZ /ATTEMPTING TO WRITE ECC POSITION/
8619	064276	044524	043516	052040	
8620	064304	020117	051127	052111	
8621	064312	020105	041505	020103	
8622	064320	047520	044523	044524	
8623	064326	047117	000		
8624	064331	101	052124	046505	EM16: .ASCIZ /ATTEMPTING TO WRITE CS1/
8625	064336	052120	047111	020107	

8626	064344	047524	053440	044522	
8627	064352	042524	041440	030523	
8628	064360	000			
8629	064361	101	052124	046505	EM17: .ASCIZ "ATTEMPTING TO WRITE ATTENTION SUMMARY/OFFSET"
8630	064366	052120	047111	020107	
8631	064374	047524	053440	044522	
8632	064402	042524	040440	052124	
8633	064410	047105	044524	047117	
8634	064416	051440	046525	040515	
8635	064424	054522	047457	043106	
8636	064432	042523	000124		
8637	064436	052101	042524	050115	EM18: .ASCIZ /ATTEMPTING TO WRITE CS2/
8638	064444	044524	043516	052040	
8639	064452	020117	051127	052111	
8640	064460	020105	051503	000062	
8641	064466	052101	042524	050115	EM19: .ASCIZ /ATTEMPTING TO WRITE DISK CYL ADD/
8642	064474	044524	043516	052040	
8643	064502	020117	051127	052111	
8644	064510	020105	044504	045523	
8645	064516	041440	046131	040440	
8646	064524	042104	000		
8647	064527	101	052124	046505	EM20: .ASCIZ /ATTEMPTING TO WRITE MAINT REG 1/
8648	064534	052120	047111	020107	
8649	064542	047524	053440	044522	
8650	064550	042524	046440	044501	
8651	064556	052116	051040	043505	
8652	064564	030440	000		
8653	064567	101	052124	046505	EM21: .ASCII /ATTEMPTING TO CLEAR BUS ADD /
8654	064574	052120	047111	020107	
8655	064602	047524	041440	042514	
8656	064610	051101	041040	051525	
8657	064615	040440	042104	040	
8658	064623	127	052111	020110	.ASCIZ /WITH A SUB CLEAR/
8659	064630	020101	052523	020102	
8660	064636	046103	040505	000122	
8661	064644	052101	042524	050115	EM22: .ASCII /ATTEMPTING TO CLEAR CS1 /
8662	064652	044524	043516	052040	
8663	064660	020117	046103	040505	
8664	064666	020122	051503	020061	
8665	064674	044527	044124	040440	.ASCIZ /WITH A SUB CLEAR/
8666	064702	051440	041125	041440	
8667	064710	042514	051101	000	
8668	064715	101	052124	046505	EM23: .ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A SUBSYSTEM CLEAR/
8669	064722	052120	047111	020107	
8670	064730	047524	041440	042514	
8671	064736	051101	051040	033113	
8672	064744	030461	053440	052111	
8673	064752	020110	020101	052523	
8674	064760	051502	051531	042524	
8675	064766	020115	046103	040505	
8676	064774	000122			
8677	064776	047111	051124	042040	EM24: .ASCII /INTR DID NOT OCCUR WHEN READY AND/<15><12>
8678	065004	042111	047040	052117	
8679	065012	047440	041503	051125	
8680	065020	053440	042510	020116	
8681	065026	042522	042101	020131	

8682	065034	047101	006504	012	
8683	065041	111	052116	020122	.ASCIZ /INTR ENABLE SET IN CSI/
8684	065046	047105	041101	042514	
8685	065054	051440	052105	044440	
8686	065062	020116	051503	000061	
8687	065070	047125	054105	042520	EM25: .ASCIZ /UNEXPECTED INTR WHEN PROCESSOR PRIORITY LOWERED/
8688	065076	052103	042105	044440	
8689	065104	052116	020122	044127	
8690	065112	047105	050040	047522	
8691	065120	042503	051523	051117	
8692	065126	050040	044522	051117	
8693	065134	052111	020131	047514	
8694	065142	042527	042522	000104	
8695	065150	047111	051124	053040	EM26: .ASCIZ /INTR VECTOR ADDRESS INCORRECT/
8696	065156	041505	047524	020122	
8697	065164	042101	051104	051505	
8698	065172	020123	047111	047503	
8699	065200	051122	041505	000124	
8700	065206	047111	051124	042040	EM27: .ASCIZ /INTR DID NOT CLEAR IN RK611/
8701	065214	042111	047040	052117	
8702	065222	041440	042514	051101	
8703	065230	044440	020116	045522	
8704	065236	030466	000061		
8705	065242	054105	042520	052103	EM28: .ASCIZ /EXPECTED INTR DID NOT OCCUR AT PROCESSOR PRIORITY/
8706	065250	042105	044440	052116	
8707	065256	020122	044504	020104	
8708	065264	047516	020124	041517	
8709	065272	052503	020122	052101	
8710	065300	050040	047522	042503	
8711	065306	051523	051117	050040	
8712	065314	044522	051117	052111	
8713	065322	000131			
8714	065324	047125	054105	042520	EM29: .ASCIZ /UNEXPECTED INTR OCCURRED AT PROCESSOR PRIORITY/
8715	065332	052103	042105	044440	
8716	065340	052116	020122	041517	
8717	065346	052503	051122	042105	
8718	065354	040440	020124	051120	
8719	065362	041517	051505	047523	
8720	065370	020122	051120	047511	
8721	065376	044522	054524	000	
8722	065403	111	052116	020122	EM30: .ASCIZ /INTR DID NOT OCCUR WHEN PRIORITY LOWERED/
8723	065410	044504	020104	047516	
8724	065416	020124	041517	052503	
8725	065424	020122	044127	047105	
8726	065432	050040	044522	051117	
8727	065440	052111	020131	047514	
8728	065446	042527	042522	000104	
8729	065454	042523	052124	047111	EM31: .ASCIZ /SETTING INTR ENABLE CAUSED EXPECTED INTR/
8730	065462	020107	047111	051124	
8731	065470	042440	040516	046102	
8732	065476	020105	040503	051525	
8733	065504	042105	042440	050130	
8734	065512	041505	042524	020104	
8735	065520	047111	051124	000	
8736	065525	103	047117	051124	EM32: .ASCIZ /CONTROLLER CLEAR DID CLEAR PENDING INTR/
8737	065532	046117	042514	020122	

8738	065540	046103	040505	020122	
8739	065546	044504	020104	046103	
8740	065554	040505	020122	042520	
8741	065562	042116	047111	020107	
8742	065570	047111	051124	000	
8743	065575	101	052124	046505	EM33: .ASCIZ /ATTEMPTING TO WRITE ONE WORD OR DATA/
8744	065602	052120	047111	020107	
8745	065610	047524	053440	044522	
8746	065616	042524	047440	042516	
8747	065624	053440	051117	020104	
8748	065632	051117	042040	052101	
8749	065640	000101			
8750	065642	052101	042524	050115	EM34: .ASCIZ /ATTEMPTING TO READ SILO WHEN EMPTY/
8751	065650	044524	043516	052040	
8752	065656	020117	042522	042101	
8753	065664	051440	046111	020117	
8754	065672	044127	047105	042440	
8755	065700	050115	054524	000	
8756	065705	101	052124	046505	EM35: .ASCIZ /ATTEMPTING TO CLEAR DATA LATA/
8757	065712	052120	047111	020107	
8758	065720	047524	041440	042514	
8759	065726	051101	042040	052101	
8760	065734	020101	040514	040524	
8761	065742	000			
8762	065743	101	052124	046505	EM36: .ASCIZ /ATTEMPTING TO READ SILO CONTAINING ONE WORD/
8763	065750	052120	047111	020107	
8764	065756	047524	051040	040505	
8765	065764	020104	044523	047514	
8766	065772	041440	047117	040524	
8767	066000	047111	047111	020107	
8768	066006	047117	020105	047527	
8769	066014	042122	000		
8770	066017	101	052124	046505	EM37: .ASCIZ /ATTEMPTING TO LOAD SILO/
8771	066024	052120	047111	020107	
8772	066032	047524	046040	040517	
8773	066040	020104	044523	047514	
8774	066046	000			
8775	066047	101	052124	046505	EM38: .ASCIZ /ATTEMPTING TO READ SILO/
8776	066054	052120	047111	020107	
8777	066062	047524	051040	040505	
8778	066070	020104	044523	047514	
8779	066076	000			
8780	066077	101	052124	046505	EM39: .ASCIZ /ATTEMPTING TO LOAD FULL SILO (66 WORDS)/
8781	066104	052120	047111	020107	
8782	066112	047524	046040	040517	
8783	066120	020104	052506	046114	
8784	066126	051440	046111	020117	
8785	066134	033050	020066	047527	
8786	066142	042122	024523	000	
8787	066147	104	052101	020101	EM40: .ASCIZ /DATA LATE DID NOT CAUSE EXPECTED INTR/
8788	066154	040514	042524	042040	
8789	066162	042111	047040	052117	
8790	066170	041440	052501	042523	
8791	066176	042440	050130	041505	
8792	066204	042524	020104	047111	
8793	066212	051124	000		

8794	066215	125	042516	050130	EM41:	.ASCIZ	/UNEXPECTED INTR DUE TO UNCLEARED CONTROLLER ERROR/
8795	066222	041505	042524	020104			
8796	066230	047111	051124	042040			
8797	066236	042525	052040	020117			
8798	066244	047125	046103	040505			
8799	066252	042522	020104	047503			
8800	066260	052116	047522	046114			
8801	066266	051105	042440	051122			
8802	066274	051117	000				
8803	066277	103	047117	051124	EM42:	.ASCII	/CONTROLLER CLEAR DID NOT CLEAR PENDING INTR/<15><12>
8804	066304	046117	042514	020122			
8805	066312	046103	040505	020122			
8806	066320	044504	020104	047516			
8807	066326	020124	046103	040505			
8808	066334	020122	042520	042116			
8809	066342	047111	020107	047111			
8810	066350	051124	005015				
8811	066354	052504	020105	047524		.ASCIZ	/DUE TO CONTROLLER ERROR/
8812	066362	041440	047117	051124			
8813	066370	046117	042514	020122			
8814	066376	051105	047522	000122			
8815	066404	047503	052116	047522	EM43:	.ASCIZ	/CONTROLLER ERROR CAUSED INTR WITH INTR ENABLE RESET/
8816	066412	046114	051105	042440			
8817	066420	051122	051117	041440			
8818	066426	052501	042523	020104			
8819	066434	047111	051124	053440			
8820	066442	052111	020110	047111			
8821	066450	051124	042440	040516			
8822	066456	046102	020105	042522			
8823	066464	042523	000124				
8824	066470	047111	051124	042040	EM44:	.ASCII	/INTR DID NOT OCCUR WHEN INTR ENABLE SET/
8825	066476	042111	047040	052117			
8826	066504	047440	041503	051125			
8827	066512	053440	042510	020116			
8828	066520	047111	051124	042440			
8829	066526	040516	046102	020105			
8830	066534	042523	124				
8831	066537	015	053412	052111		.ASCIZ	<15><12>/WITH INTR PENDING DUE TO DATA LATE/
8832	066544	020110	047111	051124			
8833	066552	050040	047105	044504			
8834	066560	043516	042040	042525			
8835	066566	052040	020117	040504			
8836	066574	040524	046040	052101			
8837	066602	000105					
8838	066604	051503	020061	047111	EM1000:	.ASCIZ	/CSI INCORRECT/
8839	066612	047503	051122	041505			
8840	066620	000124					
8841	066622	047527	042122	041440	EM1001:	.ASCIZ	/WORD COUNT INCORRECT/
8842	066630	052517	052116	044440			
8843	066636	041516	051117	042522			
8844	066644	052103	000				
8845	066647	102	051525	040440	EM1002:	.ASCIZ	/BUS ADD INCORRECT/
8846	066654	042104	044440	041516			
8847	066662	051117	042522	052103			
8848	066670	000					
8849	066671	104	051511	020113	EM1003:	.ASCIZ	/DISK ADD INCORRECT/

E15

RA611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 164
DZR6A8.P11 ERROR MESSAGES

SEQ 0165

8850	066676	042101	020104	047111	
8851	066704	047503	051122	041505	
8852	066712	000124			
8853	066714	051503	020062	047111	EM1004: .ASCIZ /CS2 INCORRECT/
8854	066722	047503	051122	041505	
8855	066730	000124			
8856	066732	051104	053111	020105	EM1005: .ASCIZ /DRIVE STATUS REG INCORRECT/
8857	066740	052123	052101	051525	
8858	066746	051040	043505	044440	
8859	066754	041516	051117	042522	
8860	066762	052103	000		
8861	066765	105	051122	051117	EM1006: .ASCIZ /ERROR REG INCORRECT/
8862	066772	051040	043505	044440	
8863	067000	041516	051117	042522	
8864	067006	052103	000		
8865	067011	101	052124	047105	EM1007: .ASCIZ "ATTENTION SUMMARY/OFFSET INCORRECT"
8866	067016	044524	047117	051440	
8867	067024	046525	040515	054522	
8868	067032	047457	043106	042523	
8869	067040	020124	047111	047503	
8870	067046	051122	041505	000124	
8871	067054	040504	040524	041040	EM1008: .ASCIZ /DATA BUFFER INCORRECT/
8872	067062	043125	042506	020122	
8873	067070	047111	047503	051122	
8874	067076	041505	000124		
8875	067102	040515	047111	020124	EM1009: .ASCIZ /MAINT REG 1 INCORRECT/
8876	067110	042522	020107	020061	
8877	067116	047111	047503	051122	
8878	067124	041505	000124		
8879	067130	041505	020103	047520	EM1012: .ASCIZ /ECC POS INCORRECT/
8880	067136	020123	047111	047503	
8881	067144	051122	041505	000124	
8882	067152	041505	020103	040520	EM1013: .ASCIZ /ECC PAT INCORRECT/
8883	067160	020124	047111	047503	
8884	067166	051122	041505	000124	
8885	067174	051503	020061	044103	EM1014: .ASCIZ /CS1 CHANGED AFTER READING ALL REGS/
8886	067202	047101	042507	020104	
8887	067210	043101	042524	020122	
8888	067216	042522	042101	047111	
8889	067224	020107	046101	020114	
8890	067232	042522	051507	000	
8891	067237	103	031123	041440	EM1015: .ASCIZ /CS2 CHANGED AFTER READING ALL REGS/
8892	067244	040510	043516	042105	
8893	067252	040440	052106	051105	
8894	067260	051040	040505	044504	
8895	067266	043516	040440	046114	
8896	067274	051040	043505	000123	
8897	067302	054503	020114	042101	EM1016: .ASCIZ /CYL ADD INCORRECT/
8898	067310	020104	047111	047503	
8899	067316	051122	041505	000124	
8900	067324	051503	020061	040527	EM1017: .ASCIZ /CS1 WAS MODIFIED/
8901	067332	020123	047515	044504	
8902	067340	044506	042105	000	
8903	067345	127	051117	020104	EM1018: .ASCIZ /WORD COUNT WAS MODIFIED/
8904	067352	047503	047125	020124	
8905	067360	040527	020123	047515	

8906	067366	044504	044506	042105	
8907	067374	000			
8908	067375	102	051525	040440	EM1019: .ASCIZ /BUS ADD WAS MODIFIED/
8909	067402	042104	053440	051501	
8910	067410	046440	042117	043111	
8911	067416	042511	000104		
8912	067422	044504	045523	040440	EM1020: .ASCIZ /DISK ADDRESS REG WAS MODIFIED/
8913	067430	042104	042522	051523	
8914	067436	051040	043505	053440	
8915	067444	051501	046440	042117	
8916	067452	043111	042511	000104	
8917	067460	051503	020062	040527	EM1021: .ASCIZ /CS2 WAS MODIFIED/
8918	067466	020123	047515	044504	
8919	067474	044506	042105	000	
8920	067501	104	044522	042526	EM1022: .ASCIZ /DRIVE STATUS REG WAS MODIFIED/
8921	067506	051440	040524	052524	
8922	067514	020123	042522	020107	
8923	067522	040527	020123	047515	
8924	067530	044504	044506	042105	
8925	067536	000			
8926	067537	105	051122	051117	EM1023: .ASCIZ /ERROR REG WAS MODIFIED/
8927	067544	051040	043505	053440	
8928	067552	051501	046440	042117	
8929	067560	043111	042511	000104	
8930	067566	052101	042524	052116	EM1024: .ASCIZ "ATTENTION SUMMARY/OFFSET WAS MODIFIED"
8931	067574	047511	020116	052523	
8932	067602	046515	051101	027531	
8933	067610	043117	051506	052105	
8934	067616	053440	051501	046440	
8935	067624	042117	043111	042511	
8936	067632	000104			
8937	067634	054503	020114	042101	EM1025: .ASCIZ /CYL ADD WAS MODIFIED/
8938	067642	020104	040527	020123	
8939	067650	047515	044504	044506	
8940	067656	042105	000		
8941	067661	115	044501	052116	EM1026: .ASCIZ /MAINT REG 1 WAS MODIFIED/
8942	067666	051040	043505	030440	
8943	067674	053440	051501	046440	
8944	067702	042117	043111	042511	
8945	067710	000104			
8946	067712	041505	020103	047520	EM1029: .ASCIZ /ECC POS WAS MODIFIED/
8947	067720	020123	040527	020123	
8948	067726	047515	044504	044506	
8949	067734	042105	000		
8950	067737	105	041503	050040	EM1030: .ASCIZ /ECC PAT WAS MODIFIED/
8951	067744	052101	053440	051501	
8952	067752	046440	042117	043111	
8953	067760	042511	000104		
8954					
8955	067764	000400			
8956	070764	100	100		
8957	070766	101	101		
8958	070770	102	102		
8959	070772	103	103		
8960	070774	104	104		
8961	070776	105	105		

```

.EVEN
SAVVEC: .BLKW 400 ;STORAGE FOR TRAP CATCHER
SILO: .BYTE 100,100 ;CONFIGURATION FOR SILO TEST
       .BYTE 101,101
       .BYTE 102,102
       .BYTE 103,103
       .BYTE 104,104
       .BYTE 105,105

```

8962	071000	106	106	.BYTE	106,106
8963	071002	107	107	.BYTE	107,107
8964	071004	110	110	.BYTE	110,110
8965	071006	111	111	.BYTE	111,111
8966	071010	112	112	.BYTE	112,112
8967	071012	113	113	.BYTE	113,113
8968	071014	114	114	.BYTE	114,114
8969	071016	115	115	.BYTE	115,115
8970	071020	116	116	.BYTE	116,116
8971	071022	117	117	.BYTE	117,117
8972	071024	120	120	.BYTE	120,120
8973	071026	121	121	.BYTE	121,121
8974	071030	122	122	.BYTE	122,122
8975	071032	123	123	.BYTE	123,123
8976	071034	124	124	.BYTE	124,124
8977	071036	125	125	.BYTE	125,125
8978	071040	126	126	.BYTE	126,126
8979	071042	127	127	.BYTE	127,127
8980	071044	130	130	.BYTE	130,130
8981	071046	131	131	.BYTE	131,131
8982	071050	132	132	.BYTE	132,132
8983	071052	133	133	.BYTE	133,133
8984	071054	134	134	.BYTE	134,134
8985	071056	135	135	.BYTE	135,135
8986	071060	136	136	.BYTE	136,136
8987	071062	137	137	.BYTE	137,137
8988	071064	140	140	.BYTE	140,140
8989	071066	141	141	.BYTE	141,141
8990	071070	142	142	.BYTE	142,142
8991	071072	143	143	.BYTE	143,143
8992	071074	144	144	.BYTE	144,144
8993	071076	145	145	.BYTE	145,145
8994	071100	146	146	.BYTE	146,146
8995	071102	147	147	.BYTE	147,147
8996	071104	150	150	.BYTE	150,150
8997	071106	151	151	.BYTE	151,151
8998	071110	152	152	.BYTE	152,152
8999	071112	153	153	.BYTE	153,153
9000	071114	154	154	.BYTE	154,154
9001	071116	155	155	.BYTE	155,155
9002	071120	156	156	.BYTE	156,156
9003	071122	157	157	.BYTE	157,157
9004	071124	160	160	.BYTE	160,160
9005	071126	161	161	.BYTE	161,161
9006	071130	162	162	.BYTE	162,162
9007	071132	163	163	.BYTE	163,163
9008	071134	164	164	.BYTE	164,164
9009	071136	165	165	.BYTE	165,165
9010	071140	166	166	.BYTE	166,166
9011	071142	167	167	.BYTE	167,167
9012	071144	170	170	.BYTE	170,170
9013	071146	171	171	.BYTE	171,171
9014	071150	172	172	.BYTE	172,172
9015	071152	173	173	.BYTE	173,173
9016	071154	174	174	.BYTE	174,174
9017	071156	175	175	.BYTE	175,175

H15

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 167
DZR6A8.P11 ERROR MESSAGES

SEQ 0168

9018	071160	176	176	.BYTE	176,176
9019	071162	177	177	.BYTE	177,177
9020	071164	200	200	.BYTE	200,200
9021	071166	201	201	.BYTE	201,201
9022	000001				

.END

ABASE = 177440	138#	373	414
ABORT = 062174	7705	8414#	
ACDW1 = 000000	373	416	
ACDW2 = 000000	373	417	
ACLO = 000010	233#		
ACPUOP = 000000	373	388	
ADCW0 = 000000	373		
ADDW1 = 000000	373		
ADDW10 = 000000	373		
ADDW11 = 000000	373		
ADDW12 = 000000	373		
ADDW13 = 000000	373		
ADDW14 = 000000	373		
ADDW15 = 000000	373		
ADDW2 = 000000	373		
ADDW3 = 000000	373		
ADDW4 = 000000	373		
ADDW5 = 000000	373		
ADDW6 = 000000	373		
ADDW7 = 000000	373		
ADDW8 = 000000	373		
ADDW9 = 000000	373		
ADEVCT = 000000	373	379	
ADEVN = 000000	373	415	
ARENV = 000000	373	384	
ARENVM = 000000	373	385	
AFATAL = 000000	373	376	
AMADR1 = 000000	373	401	
AMADR2 = 000000	373	405	
AMADR3 = 000000	373	408	
AMADR4 = 000000	373	411	
AMAMS1 = 000000	373	395	
AMAMS2 = 000000	373	403	
AMAMS3 = 000000	373	406	
AMAMS4 = 000000	373	409	
AMSGAD = 000000	373	381	
AMSGLG = 000000	373	382	
AMSGTY = 000000	373	375	
AMTYP1 = 000000	373	396	
AMTYP2 = 000000	373	404	
AMTYP3 = 000000	373	407	
AMTYP4 = 000000	373	410	
APASS = 000000	373	378	
APRIOR = 000005	137#	373	
APTCU = 000040	7582#	7744	
APTENV = 000001	7538	7580#	7614 7737
APTSIZ = 000200	711	7579#	
APTSPO = 000100	7540	7581#	7739
ASWREG = 000000	373	386	
ATESTN = 000000	373	377	
AUNIT = 000000	373	380	
AUSWR = 000000	373	387	
AVECT1 = 120210	136#	373	412
AVECT2 = 000000	373	413	
BAI = 000020	196#		
BA16 = 000400	181#		

BA17 =	001000	182#												
BIT0 =	000001	118#	178	211	230									
BIT00 =	000001	108#	118											
BIT01 =	000002	107#	117											
BIT02 =	000004	106#	116											
BIT03 =	000010	105#	115											
BIT04 =	000020	104#	114											
BIT05 =	000040	103#	113											
BIT06 =	000100	102#	112											
BIT07 =	000200	101#	111											
BIT08 =	000400	100#	110	7409										
BIT09 =	001000	99#	109	7427	7625									
BIT1 =	000002	117#	212											
BIT10 =	002000	98#	183	202	221	254	269	7602						
BIT11 =	004000	97#	184	203	222	239	255	270	7434					
BIT12 =	010000	96#	186	204	223	256								
BIT13 =	020000	95#	187	205	224	240	257	7609						
BIT14 =	040000	94#	188	206	225	241	258	7395						
BIT15 =	100000	93#	189	190	207	226	242	259						
BIT2 =	000004	116#	213	232										
BIT3 =	000010	115#	195	214	233									
BIT4 =	000020	114#	196	215	234	248	263							
BIT5 =	000040	113#	197	216	235	249	264							
BIT6 =	000100	112#	179	198	217	236	250	265						
BIT7 =	000200	111#	180	199	218	237	251	266						
BIT8 =	000400	110#	181	200	219	238	252	267						
BIT9 =	001000	109#	182	201	220	253	268							
BPTVEC =	000014	125#												
BSE =	000200	218#												
CCLR =	100000	190#	945	1037	1122	1124	1153	1223	1252	1324	1368	1454	1492	1578
		1616	1702	1740	1826	1865	1956	1968	1997	2088	2100	2129	2220	2232
		2261	2352	2364	2384	2474	2558	2593	2677	2712	2796	2831	2915	2952
		3038	3074	3160	3196	3282	3318	3404	3440	3526	3562	3648	3684	3770
		3806	3892	3929	4015	4054	4140	4179	4265	4304	4390	4419	4511	4596
		4680	4764	4854	4940	4974	5060	5094	5180	5214	5300	5334	5421	5459
		5546	5584	5671	5709	5796	5829	5914	6016	6328	6361	6363	6398	6412
		6446	6554	6586	6702	6818	6934	7034	7116	7129	7145	7172	7206	7225
		7227	7262	7292										
CDT =	002000	183#												
CERR =	100000	189#	6403	6543	7107	7138								
CFMT =	010000	186#	1945	2077	2209	2341								
CHKPAR =	055274	798	852	1105	7349#	8225								
CKSWR =	104407	7394	7598	7624	8271#									
CLEAR =	000005	163#												
COE =	001000	220#												
CONFIG1 =	002012	649#	1863*	1966*	1967	1995*	2098*	2099	2127*	2230*	2231	2259*	2362*	2363
		3927*	4025*	4026	4052*	4150*	4151	4177*	4275*	4276	4302*	4400*	4401	
CONFIG =	002010	648#	1035*	1147*	1154	1156	1158	1225	1228	1233	1235*	1250*	1366*	1374
		1376	1464*	1490*	1498	1500	1588*	1614*	1622	1624	1712*	1738*	1746	1748
		1836*	1862*	1871	1873	1945	1967*	1968*	1994*	2003	2005	2077	2099*	2100*
		2126*	2135	2137	2209	2231*	2232*	2258*	2267	2269	2341	2363*	2364*	2382*
		2472*	2479	2481	2560	2563	2569*	2591*	2598	2600	2679	2682	2688*	2710*
		2717	2719	2798	2801	2807*	2829*	2836	2838	2917	2920	2926*	2950*	2958
		2960	3048*	3072*	3080	3082	3170*	3194*	3202	3204	3292*	3316*	3324	3326
		3414*	3438*	3446	3448	3536*	3560*	3568	3570	3658*	3682*	3690	3692	3780*
		3804*	3812	3814	3902*	3926*	3935	3937	4026*	4027*	4051*	4060	4062	4151*

INTR = 000300
IOTVEC = 000020
IR = 000100

174#													
126#	680*	681*											
198#	875	877	923	925	968	970	1016	1018	1062	1064	1183	1185	
1284	1286	1386	1409	1411	1510	1533	1535	1634	1657	1659	1758	1781	
1783	1906	1908	2038	2040	2170	2172	2302	2304	2414	2416	2490	2513	
2515	2609	2632	2634	2728	2751	2753	2847	2870	2672	2970	2993	2995	
3092	3115	3117	3214	3237	3239	3336	3359	3361	3458	3481	3483	3580	
3603	3605	3702	3725	3727	3824	3847	3849	3939	3948	4017	4064	4073	
4142	4189	4198	4267	4314	4323	4392	4457	4459	4541	4543	4626	4629	
4710	4712	4794	4796	4872	4901	4903	4992	5021	5023	5112	5141	5143	
5232	5261	5263	5357	5386	5388	5482	5511	5513	5607	5636	5638	5732	
5761	5763	5859	5861	5944	5946	6402	6416	6480	6482	6531	6544	6619	
6621	6735	6737	6851	6853	6967	6969	7044	7076	7100	7106	7149		

LF = 000012
MCLK = 000400
MDS = 001000
MEMBAS = 172100
MEMERR = 055364
MEMVEC = 000114
MEMO = 001000
MEMSK = 000017
MEMO = 002000

32#	7787	7793											
252#													
201#	3938	4063	4188	4313									
134#	7351												
7355	7368#												
133#	7355*	7355*											
253#													
246#													
254#	896	902	989	995	1083	1204	1305	1433	1557	1681	1805	1930	
2062	2194	2326	2435	2537	2656	2775	2894	3017	3139	3261	3383	3505	
3627	3749	3871	3994	4119	4244	4369	4478	4562	4647	4731	4815	4919	
5039	5159	5279	5344	5423	5469	5548	5594	5673	5719	5798	5880	5965	
6501	6643	6756	6872	6988									

MIND = 000200
MSP = 000100
MED = 010000
MEM = 004000
MEMPAS = 003054
NXF = 000004
OFFSET = 000015
OFST = 000004
OPI = 020000
OPR001 = 062062
OPR002 = 062111
OPR003 = 062117
OPR004 = 062147
OR = 000200

251#													
250#													
204#	3938	4063	4188	4313									
203#	3938	4063	4188	4313									
789	798#	7343											
213#													
167#													
232#													
224#													
739	8398#												
742	753	770	8402#										
749	8404#												
761	8409#												
199#	3938	4063	4188	4313	6480	6482	6619	6621	6735	6737	6851	6853	
6967	6969	7044	7067	7076	7137								

PACK = 000003
PARM = 002030
PAF.EN = 000001
PAT = 000020
PC = %000007

162#													
284	658#												
135#	7354												
248#													
52#	798*	852*	1105*	6042*	6043*	6044*	6045*	6046*	6047*	6048*	6049*	6050*	
6051*	6052*	6053*	6054*	6055*	6056*	6057*	6058*	6059*	6060*	6061*	6062*	6063*	
6064*	6065*	6066*	6067*	6068*	6069*	6070*	6071*	6072*	6073*	6074*	6075*	6076*	
6077*	6078*	6079*	6080*	6081*	6082*	6083*	6084*	6085*	6086*	6087*	6088*	6089*	
6090*	6091*	6092*	6093*	6094*	6095*	6096*	6097*	6098*	6099*	6100*	6101*	6102*	
6103*	6104*	6105*	6106*	6107*	6108*	6109*	6110*	6111*	6112*	6113*	6114*	6115*	
6116*	6117*	6118*	6119*	6120*	6121*	6122*	6123*	6124*	6125*	6126*	6127*	6128*	
6129*	6130*	6131*	6132*	6133*	6134*	6135*	6136*	6137*	6138*	6139*	6140*	6141*	
6142*	6143*	6144*	6145*	6146*	6147*	6148*	6149*	6150*	6151*	6152*	6153*	6154*	
6155*	6156*	6157*	6158*	6159*	6160*	6161*	6162*	6163*	6164*	6165*	6166*	6167*	
6168*	6169*	7312*	7315*	7337*	7342	7364*	7557*	7574*	7611*	7617*	7713*	7742*	

PCA = 004000	7761*	7768*	7775*	7789*	7791*	7991*	8225*													
PCD = 010000	255*	5343	5468	5593	5718															
PGE = 002000	256*	5343	5468	5593	5718															
PIP = 020000	202*	3938	4063	4188	4313															
PIRQ = 177772	240*																			
PIRQVE = 000240	38*																			
PREREG 002014	132*																			
	650*	1102*	1127*	1330*	1453*	1577*	1701*	1825*	1955*	2087*	2219*	2351*	2557*							
	2676*	2795*	2914*	3037*	3159*	3281*	3403*	3525*	3647*	3769*	3891*	4014*	4139*							
	4264*	4389*	4426*	4435*	4939*	5059*	5179*	5299*	5420*	5545*	5670*	5795*	8286							
PRIOR 002020	652*	6261*	6278*	6279*	6280*	6281*	6282*	6283*	6290*	8291										
PRC = 000000	55*																			
PR1 = 000040	56*																			
PR2 = 000100	57*																			
PR3 = 000140	58*																			
PR4 = 000200	59*																			
PR5 = 000240	60*	642																		
PR6 = 000300	61*																			
PR7 = 000340	62*	667	799	816	6012	6027	6207	6267	6298	6331	6339	6367	6374							
	7175	7184	7199	7230	7265	7276	7350	7356	8209	8222										
PS = 177776	35*	36																		
PSW = 177776	36*																			
PWRVEC = 000024	127*	686*	687*	8208*	8209*	8221*	8222*													
RDCHR = 104410	8051	8272*																		
RDDATA = 000021	59*																			
RDGATE = 100000	259*	5343	5468	5593	5718															
RDHEAD = 000025	171*																			
RDLIN = 104411	8123	8273*																		
RDOCT = 104412	743	754	771	8274*																
RDY = 000200	180*	854	856	917	919	947	949	1010	1012	1046	1048	1162	1164							
	1263	1265	1384	1508	1632	1756	1875	1958	2007	2090	2139	2222	2271							
	2354	2388	2390	2488	2607	2726	2845	2968	3090	3212	3334	3456	3578							
	3700	3822	3946	4071	4196	4321	4432	4436	4515	4517	4600	4602	4684							
	4686	4768	4770	4870	4990	5110	5230	5355	5480	5605	5730	5833	5835							
	5918	5920	6025	6265	6288	6362	6403	6415	6451	6453	6530	6543	6590							
	6592	6706	6708	6822	6824	6938	6940	7043	7107	7138	7148									
RECAL = 000013	166*																			
RESREG = 104414	7699	8276*																		
RESTAT 002040	282	661*																		
RESVEC = 000010	122*																			
RKASOF = 000016	149*	869	962	1056	1177	1278	1402	1526	1650	1774	1899	2031	2163							
	2295	2408	2506	2625	2744	2863	2986	3108	3230	3352	3446*	3447	3525							
	3527	3568*	3569	3647	3649	3690*	3691	3769	3771	3812*	3813	3891	3893							
	3970	4095	4220	4345	4451	4535	4620	4704	4788	4894	5014	5134	5254							
	5379	5504	5629	5754	5853	5938	6471	6610	6726	6842	6958									
RKBA = 000004	144*	859	952	1038*	1039	1106	1123*	1125	1167	1268	1374*	1375	1453							
	1455	1498*	1499	1577	1579	1622*	1623	1701	1703	1746*	1747	1825	1827							
	1881	2013	2145	2277	2393	2494	2613	2732	2851	2974	3096	3218	3340							
	3462	3584	3706	3828	3952	4077	4202	4327	4421*	4424	4520	4605	4689							
	4773	4876	4996	5116	5236	5361	5486	5611	5736	5838	5923	6456	6595							
	6711	6827	6943																	
RKCS1 = 000000	142*	853	916	945*	946	1009	1037*	1045	1122*	1124*	1153*	1161	1223*							
	1252*	1262	1324*	1368*	1383	1454*	1492*	1507	1578*	1616*	1631	1702*	1740*							
	1755	1826*	1865*	1871*	1872	1955	1956*	1957	1997*	2003*	2004	2087	2088*							
	2089	2129*	2135*	2136	2219	2220*	2221	2261*	2267*	2268	2351	2352*	2353							
	2384*	2387	2474*	2487	2558*	2593*	2606	2677*	2712*	2725	2796*	2831*	2844							

E16

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B
DZR6AB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 01-OCT-76 10:19 PAGE 178

SEQ 0178

	2915*	2952*	2967	3038*	3074*	3089	3160*	3196*	3211	3282*	3318*	3333	3404*
	3440*	3455	3526*	3562*	3577	3648*	3684*	3699	3770*	3806*	3821	3892*	3929*
	3945	4015*	4054*	4070	4140*	4179*	4195	4265*	4304*	4320	4390*	4419*	4422*
	4431	4511*	4514	4596*	4599	4680*	4683	4764*	4767	4854*	4869	4940*	4974*
	4989	5060*	5094*	5109	5180*	5214*	5229	5300*	5334*	5354	5421*	5459*	5479
	5546*	5584*	5604	5671*	5709*	5729	5796*	5829*	5832	5914*	5917	6016*	6025*
	6265*	6288*	6328*	6337*	6361*	6362*	6363*	6364*	6398*	6401	6412*	6413	6446*
	6450	6528	6541	6554*	6586*	6589	6702*	6705	6818*	6821	6934*	6937	7034*
	7041	7053	7065	7084	7105	7116*	7129*	7136	7145*	7146	7172*	7181*	7206*
	7225*	7227*	7231*	7262*	7274*	7292*							
RKCS2 = 000010	146#	874	922	967	1015	1061	1182	1283	1408	1532	1656	1780	1905
	2037	2169	2301	2413	2512	2631	2750	2869	2992	3114	3236	3358	3480
	3602	3724	3846	3935*	3936	4014	4016	4060*	4061	4139	4141	4185*	4186
	4264	4266	4310*	4311	4389	4391	4423*	4456	4540	4625	4709	4793	4900
	5020	5140	5260	5385	5510	5635	5760	5858	5943	6400	6414	6479	6529
RKDA = 000006	6542	6618	6734	6850	6966	7042	7054	7066	7085	7104	7134	7147	
	145#	864	957	1051	1172	1254*	1256	1325	1396	1520	1644	1768	1893
	2025	2157	2289	2403	2500	2619	2738	2857	2958*	2959	3037	3039	3080*
	3081	3159	3161	3202*	3203	3281	3283	3324*	3325	3403	3405	3474	3596
	3718	3840	3964	4089	4214	4339	4446	4530	4615	4699	4783	4888	5008
RKDB = 000024	5128	5248	5373	5498	5623	5748	5848	5933	6466	6605	6721	6837	6953
	151#	6399	6449*	6520	6540	6588*	6659	6704*	6775	6820*	6891	6936*	7007
RKDCYL= 000020	7037*	7052*	7064*	7077	7083	7103	7131*	7182	7226	7271			
	150#	890	983	1077	1198	1299	1426	1550	1674	1798	1923	2055	2187
	2319	2429	2530	2649	2768	2887	3010	3132	3254	3376	3498	3620	3742
	3864	3987	4112	4237	4362	4472	4556	4641	4725	4809	4860*	4861	4939
	4941	4980*	4981	5059	5061	5100*	5101	5179	5181	5220*	5221	5299	5301
RKDS = 000012	5403	5528	5653	5778	5874	5959	6495	6634	6750	6866	6982		
	147#	880	973	1067	1188	1289	1415	1539	1663	1787	1912	2044	2176
	2308	2419	2519	2638	2757	2876	2999	3121	3243	3365	3487	3609	3731
	3853	3976	4101	4226	4351	4462	4513*	4546	4631	4715	4799	4907	5027
	5147	5267	5392	5517	5642	5767	5864	5949	6485	6624	6740	6856	6972
RKECPS= 000030	155#	310	1003	1096	1217	1318	1447	1571	1695	1819	1944	2076	2208
	2340	2448	2551	2670	2789	2908	3031	3153	3275	3397	3519	3641	3763
	3885	4008	4133	4258	4383	4491	4575	4560	4744	4828	4933	5053	5173
RKECPT= 000032	5293	5414	5539	5664	5789	5993	5916*	5978	6514	6653	6769	6885	7001
	156#	905	998	1091	1212	1313	1442	1566	1690	1814	1939	2071	2203
	2335	2443	2546	2665	2784	2903	3026	3148	3270	3392	3514	3636	3758
	3880	4003	4128	4253	4378	4486	4570	4655	4739	4823	4928	5048	5168
RKER = 000014	5288	5409	5534	5659	5784	5831*	5888	5973	6509	6648	6764	6880	6996
	148#	885	978	1072	1193	1294	1420	1544	1668	1792	1917	2049	2181
	2313	2424	2524	2643	2762	2881	3004	3126	3248	3370	3492	3614	3736
	3858	3981	4106	4231	4356	4467	4551	4598*	4636	4720	4804	4912	5032
RKMR1 = 000026	5152	5272	5397	5522	5647	5772	5869	5954	6490	6629	6745	6861	6977
	152#	895	988	1082	1203	1304	1432	1556	1680	1804	1929	2061	2193
	2325	2434	2536	2655	2774	2893	3016	3138	3260	3382	3504	3626	3748
	3870	3993	4118	4243	4368	4477	4561	4646	4730	4814	4918	5038	5158
	5278	5340*	5341	5420	5422	5465*	5466	5545	5547	5590*	5591	5670	5672
	5715*	5716	5795	5797	5879	5964	6500	6639	6755	6871	6987		
RKMR2 = 000034	153#	4682*											
RKMR3 = 000036	154#	4766*											
RKPRI = 002002	642#	787*	6253	6278	6284								
RKSPAR= 000022	157#	2386*											
RKVEC = 002000	641#	785*	786*	6189	6251	6276*	6291*	6313	6329	6345	6365	6380	7173
	7192*	7207	7228	7244	7263	7273*	7293	8289					
RKWC = 000002	143#	1154*	1155	1224	1253*	1273	1373*	1390	1497*	1514	1621*	1638	1745*

F16

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B
DZR6A8.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 01-OCT-76 10:19 PAGE 179

SEQ 0179

		1762	1870*	1887	2002*	2019	2134*	2151	2266*	2283	2385*	2398	2479*	2480
		2557	2559	2598*	2599	2676	2678	2717*	2718	2795	2797	2836*	2837	2914
		2916	2957*	2980	3079*	3102	3201*	3224	3323*	3346	3445*	3468	3567*	3590
		3689*	3712	3811*	3834	3934*	3958	4059*	4083	4184*	4208	4309*	4333	4420*
		4441	4512*	4525	4597*	4610	4681*	4694	4765*	4778	4859*	4882	4979*	5002
		5099*	5122	5219*	5242	5339*	5367	5464*	5492	5589*	5617	5714*	5742	5830*
		5843	5915*	5928	6447*	6461	6587*	6600	6703*	6716	6819*	6832	6935*	6948
RLS	= 000010	195#												
RO	=%000000	43#	791*	793*	6002*	6004	6008*	6011*	6012*	6032*	6034*	6176*	6178*	6192*
		6194*	6214*	6216*	6227*	6229*	6476*	6477*	6615*	6616*	6731*	6732*	6847*	6848*
		6963*	6964*	7038*	7039*	7078*	7079*	7130*	7132*	7334*	7337	7534	7542*	7546
		7547	7549*	7550*	7551	7573*	7649*	7650*	7651*	7652*	7653*	7654*	7655*	7656
		7661	7666	7669*	7670	7672	7673	7685	7690	7735	7736*	7741	7746	7749*
		7883	7893*	7897	7913	7914	7927*	8120	8124*	8125	8128	8148*	8151*	8175
		8200*	8240	8241*	8242	8243*	8244*	8245*	8246*					
R1	=%000001	44#	790*	793	1365*	1465*	1489*	1589*	1613*	1713*	1737*	1837*	1861*	1969*
		1993*	2101*	2125*	2233*	2257*	2365*	2471*	2570*	2590*	2689*	2709*	2808*	2828*
		2927*	2949*	3049*	3071*	3171*	3193*	3293*	3315*	3415*	3437*	3537*	3559*	3659*
		3681*	3781*	3803*	3903*	3925*	4028*	4050*	4153*	4175*	4278*	4300*	4403*	4851*
		4951*	4971*	5071*	5091*	5191*	5211*	5311*	5331*	5436*	5456*	5561*	5581*	5686*
		5706*	5811*	6001*	6004*	6009*	6011	6013*	6031*	6034	6175*	6178	6191*	6194
		6213*	6216	6226*	6229	6251*	6252*	6313*	6314*	6315*	6316*	6329*	6330*	6331*
		6345*	6346*	6347*	6348*	6365*	6366*	6367*	6380*	6381*	6382*	6383*	6581*	6668*
		6697*	6784*	6813*	6900*	6929*	7016*	7173*	7174*	7175*	7207*	7208*	7209*	7210*
		7228*	7229*	7230*	7244*	7245*	7246*	7247*	7263*	7264*	7265*	7293*	7294*	7295*
		7296*	7535	7572*	7666*	7678	7884	7897*	7898	7902	7926*	8121	8126*	8134*
		8136*	8138*	8141*	8144	8147*	8176	8199*						
R2	=%000002	45#	814*	818	819	825	849*	853	859	864	869	874	880	890
		895	905	910	916	922	943*	945*	946	952	957	962	967	973
		983	988	998	1003	1009	1015	1034*	1037*	1038*	1039	1045	1051	1056
		1061	1067	1072	1077	1082	1091	1096	1106	1121*	1122*	1123*	1124*	1125
		1146*	1153*	1154*	1155	1161	1167	1172	1177	1182	1188	1193	1198	1203
		1212	1217	1223*	1224	1249*	1252*	1253*	1254*	1256	1262	1268	1273	1278
		1283	1289	1294	1299	1304	1313	1318	1324*	1325	1373*	1374*	1375	1383
		1390	1396	1402	1408	1415	1420	1426	1432	1442	1447	1453	1454*	1455
		1497*	1498*	1499	1507	1514	1520	1526	1532	1539	1544	1550	1556	1566
		1571	1577	1578*	1579	1621*	1622*	1623	1631	1638	1644	1650	1656	1663
		1668	1674	1680	1690	1695	1701	1702*	1703	1745*	1746*	1747	1755	1762
		1768	1774	1780	1787	1792	1798	1804	1814	1819	1825	1826*	1827	1870*
		1871*	1872	1881	1887	1893	1899	1905	1912	1917	1923	1929	1939	1944
		1955	1956*	1957	2002*	2003*	2004	2013	2019	2025	2031	2037	2044	2049
		2055	2061	2071	2076	2087	2088*	2089	2089	2134*	2135*	2136	2145	2151
		2163	2169	2176	2181	2187	2193	2203	2208	2208	2219	2220*	2221	2266*
		2268	2277	2283	2289	2295	2301	2308	2313	2319	2325	2335	2340	2351
		2352*	2353	2381*	2384*	2385*	2386*	2387	2393	2398	2403	2408	2413	2419
		2424	2429	2434	2443	2448	2479*	2480	2487	2494	2500	2506	2512	2519
		2524	2530	2536	2546	2551	2557	2558*	2559	2598*	2599	2606	2613	2619
		2625	2631	2638	2643	2649	2655	2665	2670	2676	2677*	2678	2717*	2718
		2725	2732	2738	2744	2750	2757	2762	2768	2774	2784	2789	2795	2796*
		2797	2836*	2837	2844	2851	2857	2863	2869	2876	2881	2887	2893	2903
		2908	2914	2915*	2916	2957*	2958*	2959	2967	2974	2980	2986	2992	2999
		3004	3010	3016	3026	3031	3037	3038*	3039	3079*	3080*	3081	3089	3096
		3102	3108	3114	3121	3126	3132	3138	3148	3153	3159	3160*	3161	3201*
		3202*	3203	3211	3218	3224	3230	3236	3243	3248	3254	3260	3270	3275
		3281	3282*	3283	3323*	3324*	3325	3333	3340	3346	3352	3358	3365	3370
		3376	3382	3392	3397	3403	3404*	3405	3445*	3446*	3447	3455	3462	3468

G16

3474	3480	3487	3492	3498	3504	3514	3519	3525	3526*	3527	3567*	3568*
3569	3577	3584	3590	3596	3602	3603	3614	3620	3626	3636	3641	3647
3648*	3649	3689*	3690*	3691	3699	3706	3712	3718	3724	3731	3736	3742
3748	3758	3763	3769	3770*	3771	3811*	3812*	3813	3821	3828	3834	3840
3846	3853	3858	3864	3870	3880	3885	3891	3892*	3893	3934*	3935*	3936
3945	3952	3958	3964	3970	3976	3981	3987	3993	4003	4008	4014	4015*
4016	4059*	4060*	4061	4070	4077	4083	4089	4095	4101	4106	4112	4118
4128	4133	4139	4140*	4141	4184*	4185*	4186	4195	4202	4208	4214	4220
4226	4231	4237	4243	4253	4258	4264	4265*	4266	4309*	4310*	4311	4320
4327	4333	4339	4345	4351	4356	4362	4368	4378	4383	4389	4390*	4391
4418*	4419*	4420*	4421*	4422*	4423*	4424	4431	4441	4446	4451	4456	4462
4467	4472	4477	4486	4491	4508*	4511*	4512*	4513*	4514	4520	4525	4530
4535	4540	4546	4551	4556	4561	4570	4575	4593*	4596*	4597*	4598*	4599
4605	4610	4615	4620	4625	4631	4636	4641	4646	4655	4660	4677*	4680*
4681*	4682*	4683	4689	4694	4699	4704	4709	4715	4720	4725	4730	4739
4744	4761*	4764*	4765*	4766*	4767	4773	4778	4783	4788	4793	4799	4804
4809	4814	4823	4828	4859*	4860*	4861	4869	4876	4882	4888	4894	4900
4907	4912	4918	4928	4933	4939	4940*	4941	4979*	4980*	4981	4989	4996
5002	5008	5014	5020	5027	5032	5038	5048	5053	5059	5060*	5061	5099*
5100*	5101	5109	5116	5122	5128	5134	5140	5147	5152	5158	5168	5173
5179	5180*	5181	5219*	5220*	5221	5229	5236	5242	5248	5254	5260	5267
5272	5278	5288	5293	5299	5300*	5301	5339*	5340*	5341	5354	5361	5367
5373	5379	5385	5392	5397	5403	5409	5414	5420	5421*	5422	5464*	5465*
5466	5479	5486	5492	5498	5504	5510	5517	5522	5528	5534	5539	5545
5546*	5547	5589*	5590*	5591	5604	5611	5617	5623	5629	5635	5642	5647
5653	5659	5664	5670	5671*	5672	5714*	5715*	5716	5729	5736	5742	5748
5754	5760	5767	5772	5778	5784	5789	5795	5796*	5797	5826*	5829*	5830*
5831*	5832	5838	5843	5848	5853	5858	5864	5869	5874	5879	5888	5893
5911*	5914*	5915*	5916*	5917	5923	5928	5933	5938	5943	5949	5954	5959
5964	5973	5978	6016*	6025*	6265*	6288*	6328*	6337*	6361*	6362*	6363*	6364*
6397*	6398*	6399	6400	6401	6412*	6413	6414	6446*	6447*	6449*	6450	6456
6461	6466	6471	6479	6485	6490	6495	6500	6509	6514	6520	6528	6529
6540	6541	6542	6554*	6586*	6587*	6588*	6589	6595	6600	6605	6610	6618
6624	6629	6634	6639	6648	6653	6659	6702*	6703*	6704*	6705	6711	6716
6721	6726	6734	6740	6745	6750	6755	6764	6769	6775	6818*	6819*	6820*
6821	6827	6832	6837	6842	6850	6856	6861	6866	6871	6880	6885	6891
6934*	6935*	6936*	6937	6943	6948	6953	6958	6966	6972	6977	6982	6987
6996	7001	7007	7033*	7034*	7037*	7041	7042	7052*	7053	7054	7064*	7065
7066	7077	7083	7084	7085	7103	7104	7105	7116*	7128*	7129*	7131*	7134
7136	7145*	7146	7147	7171*	7172*	7181*	7182	7206*	7224*	7225*	7226	7227*
7231*	7261*	7262*	7271	7274*	7292*	7670*	7688*	7885	7896*	7900*	7903	7910*
7911*	7912	7917*	7925*	8122	8127*	8135*	8137*	8139*	8145	8146*	8177	8198*
46#	792*	794*	817*	820*	6003*	6005*	6010*	6014*	6033*	6035*	6177*	6179*
6193*	6195*	6215*	6217*	6228*	6230*	7035*	7037	7052	7064	7075*	7086	7351*
7354*	7672*	7674	7680*	7826	7835*	7841*	7842*	7845*	7850*	7851*	7852	7861*
7886	7894*	7895*	7909*	7912*	7921*	7922*	7924*	8046	8048*	8049	8052*	8053
8060*	8061	8063	8071	8075	8077*	8083	8085	8087*	8090*	8178	8197*	
47#	7353*	7357*	7668*	7676	7687*	7692	7827	7829*	7830*	7831*	7832	7833*
7847	7849*	7857*	7860*	8179	8196*							
48#	6017*	6024*	6042	6043	6044	6045	6046	6047	6048	6049	6050	6051
6052	6053	6054	6055	6056	6057	6058	6059	6060	6061	6062	6063	6064
6065	6066	6067	6068	6069	6070	6071	6072	6073	6074	6075	6076	6077
6078	6079	6080	6081	6082	6083	6084	6085	6086	6087	6088	6089	6090
6091	6092	6093	6094	6095	6096	6097	6098	6099	6100	6101	6102	6103
6104	6105	6106	6107	6108	6109	6110	6111	6112	6113	6114	6115	6116
6117	6118	6119	6120	6121	6122	6123	6124	6125	6126	6127	6128	6129

R3 =%000003

R4 =%000004

R5 =%000005

SW00	=	000001	90#	90		
SW01	=	000002	79#	89		
SW02	=	000004	78#	88		
SW03	=	000010	77#	87		
SW04	=	000020	76#	86		
SW05	=	000040	75#	85		
SW06	=	000100	74#	84		
SW07	=	000200	73#	83		
SW08	=	000400	72#	82		
SW09	=	001000	71#	81		
SW1	=	000002	89#			
SW10	=	002000	70#			
SW11	=	004000	69#			
SW12	=	010000	68#	7701		
SW13	=	020000	67#			
SW14	=	040000	66#			
SW15	=	100000	65#			
SW2	=	000004	80#			
SW3	=	000010	87#			
SW4	=	000020	86#			
SW5	=	000040	85#			
SW6	=	000100	84#			
SW7	=	000200	83#			
SW8	=	000400	82#			
SW9	=	001000	81#	7372	7520	
S. CLR	=	000400	267#			
S. FMT	=	001000	268#			
S. PACK	=	004000	270#			
S. RECL	=	000040	264#			
S. RTC	=	000200	266#			
S. SEEK	=	000020	263#			
S. STSP	=	000100	265#			
S. UNLD	=	002000	269#			
TBITVE	=	000014	123#			
TKVEC	=	000060	130#			
TPVEC	=	000064	131#			
TRAPPC	=	002024	654#	7369*	8280	
TRAPVE	=	000034	129#	684*	685*	
TRTVEC	=	000014	124#			
TST1	=	003072	800	812#	7454	
TST10	=	006712	1326	1363#	7461	
TST11	=	007540	1466	1487#	7462	
TST12	=	010366	1590	1611#	7463	
TST13	=	011212	1714	1735#	7464	
TST14	=	012036	1838	1859#	7465	
TST15	=	012724	1970	1991#	7466	
TST16	=	013614	2102	2123#	7467	
TST17	=	014500	2234	2255#	7468	
TST2	=	003164	847#	7455		
TST20	=	015364	2366	2379#	7469	
TST21	=	016072	2450	2469#	7470	
TST22	=	016710	2571	2588#	7471	
TST23	=	017526	2690	2707#	7472	
TST24	=	020342	2809	2826#	7473	
TST25	=	021156	2928	2947#	7474	
TST26	=	022004	3050	3069#	7475	

K16

RA611 DISKLESS CONTROLLER DIAGNOSTIC: P1 MD-11-DZR6A-B MACY11 27(732) 01-OCT-76 10:19 PAGE 184
 DZR6AB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0184

T.CS1	001700	603*	6401*	6408	6413*	6417	6528*	6532	6541*	6550	7041*	7045	7053*	7055
		7065*	7068	7084*	7087	7105*	7112	7136*	7142	7146*	7150	8292	8294	8297
T.CS2	001710	607*	6400*	6404	6414*	6421	6529*	6536	6542*	6545	7042*	7048	7054*	7058
		7066*	7071	7085*	7090	7104*	7108	7134*	7139	7147*	7154	8292	8294	8297
T.DA	001706	606*												
T.DB	001722	612*												
T.DCYL	001720	611*												
T.DS	001712	608*												
T.ECPS	001732	616*												
T.ECPT	001734	617*												
T.ER	001714	609*												
T.MR1	001724	613*												
T.MR2	001726	614*												
T.MR3	001730	615*												
T.SPAP	001736	618*												
T.WC	001702	604*												
UFE =	000400	200*	3938	4063	4188	4313								
UNLOAD =	000007	164*												
UNS =	040000	225*												
UPE =	020000	205*	3938	4063	4188	4313								
VV =	000100	236*												
WCE =	040000	206*	3938	4063	4188	4313								
WLE =	004000	222*												
WRDATA =	000023	170*												
WRHEAD =	000027	172*												
WRL =	004000	239*												
WRTCHK =	000031	173*												
WRTGAT =	040000	258*	5343	5468	5593	5718								
\$APTHD	001000	305	311*											
\$ASTAT =	***** U	7560	7575											
\$ATYC	056204	7531	7533*											
\$ATY1	056160	7529*												
\$ATY3	056166	7530*	7742											
\$ATY4	056176	7532*	7617											
\$AUTOB	001134	342*	732*	7955	8104									
\$BASE	001270	414*	740	748*	814	849	943	1034	1121	1146	1249	2381	4418	4508
		4593	4677	4761	5826	5911	6397	7033	7128	7171	7224	7261		
\$BDADR	001122	337*	825*	6171*	6172*	6173*	6185*	6186*	6187*	6189	6222*	6223*	6224*	8281
		8289												
\$BD0AT	001126	339*	853*	854	859*	864*	869*	874*	875	880*	885*	890*	895*	897
		900	905*	910*	911	916*	917	922*	923	946*	947	952*	957*	962*
		967*	968	973*	978*	983*	988*	990	993	998*	1003*	1004	1009*	1010
		1015*	1016	1039*	1040	1045*	1046	1051*	1056*	1061*	1062	1067*	1072*	1077*
		1082*	1084	1087	1091*	1096*	1097	1106*	1125*	1155*	1156	1161*	1162	1167*
		1172*	1177*	1182*	1183	1188*	1193*	1198*	1203*	1205	1208	1212*	1217*	1218
		1224*	1225	1256*	1257	1262*	1263	1268*	1273*	1278*	1283*	1284	1289*	1294*
		1299*	1304*	1306	1309	1313*	1318*	1319	1325*	1375*	1378	1383*	1384	1390*
		1396*	1402*	1408*	1409	1415*	1420*	1426*	1432*	1434	1437	1442*	1447*	1449
		1455*	1457	1499*	1502	1507*	1508	1514*	1520*	1526*	1532*	1533	1539*	1544*
		1550*	1556*	1558	1561	1566*	1571*	1573	1579*	1581	1623*	1626	1631*	1632
		1638*	1644*	1650*	1656*	1657	1663*	1668*	1674*	1680*	1682	1685	1690*	1695*
		1697	1703*	1705	1747*	1750	1755*	1756	1762*	1768*	1774*	1780*	1781	1787*
		1792*	1798*	1804*	1806	1809	1814*	1819*	1821	1827*	1829	1872*	1876	1881*
		1887*	1893*	1899*	1905*	1906	1912*	1917*	1923*	1929*	1931	1934	1939*	1944*
		1951	1957*	1959	2004*	2008	2013*	2019*	2025*	2031*	2037*	2038	2044*	2049*
		2055*	2061*	2063	2066	2071*	2076*	2083	2089*	2091	2136*	2140	2145*	2151*

2157*	2163*	2169*	2170	2176*	2181*	2187*	2193*	2195	2198	2203*	2208*	2215
2221*	2223	2268*	2272	2277*	2293*	2283*	2295*	2301*	2302	2308*	2313*	2319*
2325*	2327	2330	2335*	2340*	2347	2353*	2355	2387*	2388	2393*	2398*	2403*
2408*	2413*	2414	2419*	2424*	2429*	2434*	2426	2439	2443*	2448*	2449	2480*
2482	2487*	2488	2494*	2500*	2506*	2512*	2513	2519*	2524*	2530*	2536*	2538
2541	2546*	2551*	2553	2559*	2560	2599*	2601	2606*	2607	2613*	2619*	2625*
2631*	2632	2638*	2643*	2649*	2655*	2657	2660	2665*	2670*	2672	2678*	2679
2718*	2720	2725*	2726	2732*	2738*	2744*	2750*	2751	2757*	2762*	2768*	2774*
2776	2779	2784*	2789*	2791	2797*	2798	2837*	2839	2844*	2845	2851*	2857*
2863*	2869*	2870	2876*	2881*	2887*	2893*	2895	2898	2903*	2908*	2910	2916*
2917	2959*	2962	2967*	2968	2974*	2980*	2986*	2992*	2993	2999*	3004*	3010*
3016*	3018	3021	3026*	3031*	3033	3039*	3041	3081*	3084	3089*	3090	3096*
3102*	3108*	3114*	3115	3121*	3126*	3132*	3138*	3140	3143	3148*	3153*	3155
3161*	3163	3203*	3206	3211*	3212	3218*	3224*	3230*	3236*	3237	3243*	3248*
3254*	3260*	3262	3265	3270*	3275*	3277	3283*	3285	3325*	3328	3333*	3334
3340*	3346*	3352*	3358*	3359	3365*	3370*	3376*	3382*	3384	3387	3392*	3397*
3399	3405*	3407	3447*	3450	3455*	3456	3462*	3468*	3474*	3480*	3481	3487*
3492*	3498*	3504*	3506	3509	3514*	3519*	3521	3527*	3529	3569*	3572	3577*
3578	3584*	3590*	3596*	3602*	3603	3609*	3614*	3620*	3626*	3628	3631	3636*
3641*	3643	3649*	3651	3691*	3694	3699*	3700	3706*	3712*	3718*	3724*	3725
3731*	3736*	3742*	3748*	3750	3753	3758*	3763*	3765	3771*	3773	3813*	3816
3821*	3822	3828*	3834*	3840*	3846*	3847	3853*	3858*	3864*	3870*	3872	3875
3880*	3885*	3887	3893*	3895	3936*	3940	3945*	3946	3952*	3958*	3964*	3970*
3976*	3981*	3987*	3993*	3995	3998	4003*	4008*	4010	4016*	4018	4061*	4065
4070*	4071	4077*	4083*	4089*	4095*	4101*	4106*	4112*	4118*	4120	4123	4128*
4133*	4135	4141*	4143	4186*	4190	4195*	4196	4202*	4208*	4214*	4220*	4226*
4231*	4237*	4243*	4245	4248	4253*	4258*	4260	4266*	4268	4311*	4315	4320*
4321	4327*	4333*	4339*	4345*	4351*	4356*	4362*	4368*	4370	4373	4378*	4383*
4385	4391*	4393	4424*	4431*	4432	4441*	4446*	4451*	4456*	4457	4462*	4467*
4469*	4472*	4477*	4479	4482	4486*	4491*	4492	4514*	4515	4520*	4525*	4530*
4535*	4540*	4541	4546*	4551*	4556*	4561*	4563	4566	4570*	4575*	4576	4599*
4600	4605*	4610*	4615*	4620*	4625*	4626	4631*	4636*	4641*	4646*	4648	4651
4655*	4660*	4661	4683*	4684	4689*	4694*	4699*	4704*	4709*	4710	4715*	4720*
4725*	4730*	4732	4735	4739*	4744*	4745	4767*	4768	4773*	4778*	4783*	4788*
4793*	4794	4799*	4804*	4809*	4814*	4816	4819	4823*	4828*	4829	4861*	4864
4869*	4870	4876*	4882*	4888*	4894*	4900*	4901	4907*	4912*	4918*	4920	4923
4928*	4933*	4935	4941*	4943	4981*	4984	4989*	4990	4996*	5002*	5008*	5014*
5020*	5021	5027*	5032*	5038*	5040	5043	5048*	5053*	5055	5061*	5063	5101*
5104	5109*	5110	5116*	5122*	5128*	5134*	5140*	5141	5147*	5152*	5158*	5160
5163	5168*	5173*	5175	5181*	5183	5221*	5224	5229*	5230	5236*	5242*	5248*
5254*	5260*	5261	5267*	5272*	5278*	5280	5283	5288*	5293*	5295	5301*	5303
5341*	5345	5349	5354*	5355	5361*	5367*	5373*	5379*	5385*	5386	5392*	5397*
5403*	5409*	5414*	5416	5422*	5424	5428	5466*	5470	5474	5479*	5480	5486*
5492*	5498*	5504*	5510*	5511	5517*	5522*	5528*	5534*	5539*	5541	5547*	5549
5553	5591*	5595	5599	5604*	5605	5611*	5617*	5623*	5629*	5635*	5636	5642*
5647*	5653*	5659*	5664*	5666	5672*	5674	5678	5716*	5720	5724	5729*	5730
5736*	5742*	5748*	5754*	5760*	5761	5767*	5772*	5778*	5784*	5789*	5791	5797*
5799	5803	5832*	5833	5838*	5843*	5848*	5853*	5858*	5859	5864*	5869*	5874*
5879*	5881	5884	5888*	5893*	5894	5917*	5918	5923*	5928*	5933*	5938*	5943*
5944	5949*	5954*	5959*	5964*	5966	5969	5973*	5978*	5979	6450*	6451	6456*
6461*	6466*	6471*	6479*	6480	6485*	6490*	6495*	6500*	6502	6505	6509*	6514*
6515	6520*	6521	6589*	6590	6595*	6600*	6605*	6610*	6618*	6619	6624*	6629*
6634*	6639*	6641	6644	6648*	6653*	6654	6659*	6660	6705*	6706	6711*	6716*
6721*	6726*	6734*	6735	6740*	6745*	6750*	6755*	6757	6760	6764*	6769*	6770
6775*	6776	6821*	6822	6827*	6832*	6837*	6842*	6850*	6851	6856*	6861*	6866*
6871*	6873	6876	6880*	6885*	6886	6891*	6892	6937*	6938	6943*	6948*	6953*

M16

		6958*	6966*	6967	6972*	6977*	6982*	6987*	6989	6992	6996*	7001*	7002	7007*
\$BELL	001204	7008	7077*	7083*	7093	8282	8294	8286	8297					
\$CDW1	001274	365#	7604	7637										
\$CDW2	001276	416#												
\$CHARC	057366	417#												
\$CKSWR	060044	7759*	7769*	7776	7785*	7790#								
\$CNTAG	001100	7947#	8271											
\$CM3 =	000000	325#	673	674	682	688	689	690						
\$CM4 =	000010	355#												
\$CNTLG	060717	355#	356#	357#	358#	359#	360#	361#	362#	363#				
\$CNTLU	060712	7958	8099#											
\$CPJOP	001242	7975	8073	8098#										
\$CRLF	001211	388#												
		367#	7332	7612	7637	7658	7663	7671	7694	7697	7758	7793	7986	8078
		8098	8157											
\$DBLK	060034	7894	7928	7936#										
\$DEVCT	001224	379#												
\$DEVM	001272	415#												
\$DOAGN	055264	7314	7335	7341#										
\$DTBL	060024	7897	7932#											
\$ENDAD	055254	291	720	7337*	7632									
\$ENDCT	055120	688	7316#											
\$ENULL	055270	7344#												
\$ENV	001234	384#	726	7538	7562	7614	7737							
\$ENVM	001235	385#	711	7540	7739	7744								
\$EOP	055064	7306#	7711											
\$EOPCT	055112	688#	7313#	7317	7709*									
\$ERFLG	001103	328#	7384	7423	7425	7431*	7453	7522	7599*	7637				
\$ERMAX	001115	334#	691*	7425	7448*	7453								
\$ERROR	056426	682	7597#											
\$ERRPC	001116	335#	7606*	7607*	7608	7637	8281	8282	8284	8286	8288	8289	8291	8292
		8294	8297											
\$ERRTB	001300	434#	7655											
\$ERTTL	001112	332#	7329	7333*	7605*	7637								
\$ESCAP	001202	364#	690*	7368*	7371*	7447*	7628	7630	7637					
\$ETABL	001234	383#												
\$ETEND	001300	317	418#											
\$FATAL	001216	376#	7566*											
\$FFLG	056424	7529*	7532*	7560	7569*	7577*								
\$FILLC	001156	353#	7762	7793										
\$FILLS	001155	352#	7793											
\$GDADR	001120	336#												
\$GDADR	001124	338#	856*	861*	866*	871*	877*	882*	887*	892*	896*	899*	900	902*
		907*	913*	919*	925*	949*	954*	959*	964*	970*	975*	980*	985*	989*
		992*	993	995*	1000*	1006*	1012*	1018*	1042*	1048*	1053*	1058*	1064*	1069*
		1074*	1079*	1083*	1086*	1087	1093*	1099*	1108*	1128*	1158*	1164*	1169*	1174*
		1179*	1185*	1190*	1195*	1200*	1204*	1207*	1208	1214*	1220*	1228*	1259*	1265*
		1270*	1275*	1280*	1286*	1291*	1296*	1301*	1305*	1308*	1309	1315*	1321*	1327*
		1376*	1377*	1378	1386*	1392*	1398*	1404*	1411*	1417*	1422*	1428*	1433*	1436*
		1437	1444*	1448*	1449	1456*	1457	1500*	1501*	1502	1510*	1516*	1522*	1528*
		1535*	1541*	1546*	1552*	1557*	1560*	1561	1568*	1572*	1573	1580*	1581	1624*
		1625*	1626	1634*	1640*	1646*	1652*	1659*	1665*	1670*	1676*	1681*	1684*	1685
		1692*	1696*	1697	1704*	1705	1748*	1749*	1750	1758*	1764*	1770*	1776*	1783*
		1789*	1794*	1800*	1805*	1808*	1809	1816*	1820*	1821	1828*	1829	1873*	1874*
		1875*	1876	1883*	1889*	1895*	1901*	1908*	1914*	1919*	1925*	1930*	1933*	1934
		1941*	1947*	1950*	1951	1958*	1959	2005*	2006*	2007*	2008	2015*	2021*	2027*

2033*	2040*	2046*	2051*	2057*	2062*	2065*	2066	2073*	2075*	2082*	2083	2090*
2091	2137*	2138*	2139*	2140	2147*	2153*	2159*	2165*	2172*	2178*	2183*	2189*
2194*	2197*	2198	2205*	2211*	2214*	2215	2222*	2223	2269*	2270*	2271*	2272
2279*	2285*	2291*	2297*	2304*	2310*	2315*	2321*	2325*	2329*	2330	2337*	2343*
2346*	2347*	2354*	2355	2390*	2395*	2400*	2405*	2410*	2416*	2421*	2426*	2431*
2435*	2438*	2439	2445*	2451*	2481*	2482	2490*	2496*	2502*	2508*	2515*	2521*
2526*	2532*	2537*	2540*	2541	2548*	2552*	2553	2563*	2600*	2601	2609*	2615*
2621*	2627*	2634*	2640*	2645*	2651*	2656*	2653*	2660	2667*	2671*	2672	2682*
2719*	2720	2728*	2734*	2740*	2746*	2753*	2759*	2764*	2770*	2775*	2778*	2779
2786*	2790*	2791	2801*	2838*	2839	2847*	2853*	2859*	2855*	2872*	2878*	2883*
2889*	2894*	2897*	2898	2905*	2909*	2910	2920*	2960*	2961*	2962	2970*	2976*
2982*	2989*	2995*	3001*	3006*	3012*	3017*	3020*	3021	3028*	3032*	3033	3040*
3041	3082*	3083*	3084	3092*	3098*	3104*	3110*	3117*	3123*	3128*	3134*	3139*
3142*	3143	3150*	3154*	3155	3162*	3163	3204*	3205*	3206	3214*	3220*	3226*
3232*	3239*	3245*	3250*	3256*	3261*	3264*	3265	3272*	3276*	3277	3284*	3285
3226*	3327*	3328	3336*	3342*	3348*	3354*	3361*	3367*	3372*	3378*	3383*	3386*
3387	3394*	3398*	3399	3406*	3407	3448*	3449*	3450	3458*	3464*	3470*	3476*
3483	3489*	3494*	3500*	3505*	3508*	3509	3516*	3520*	3521	3528*	3529	3570*
3571*	3572	3580*	3586*	3592*	3598*	3605*	3611*	3616*	3622*	3627*	3630*	3631
3638*	3642*	3643	3650*	3651	3692*	3693*	3694	3702*	3708*	3714*	3720*	3727*
3733*	3738*	3744*	3749*	3752*	3753	3760*	3764*	3765	3772*	3773	3814*	3815*
3816	3824*	3830*	3836*	3842*	3849*	3855*	3860*	3866*	3871*	3874*	3875	3892*
3886*	3887	3894*	3895	3937*	3938*	3939*	3940	3948*	3954*	3960*	3966*	3972*
3978*	3983*	3989*	3994*	3997*	3998	4005*	4007*	4010	4017*	4018	4062*	4063*
4064*	4065	4073*	4079*	4085*	4091*	4097*	4103*	4108*	4114*	4119*	4122*	4123
4130*	4134*	4135	4142*	4143	4187*	4188*	4189*	4190	4198*	4204*	4210*	4216*
4222*	4228*	4233*	4239*	4244*	4247*	4248	4255*	4259*	4260	4267*	4268	4312*
4313*	4314*	4315	4323*	4329*	4335*	4341*	4347*	4353*	4358*	4364*	4369*	4372*
4373	4380*	4384*	4385	4392*	4393	4427*	4436*	4443*	4448*	4453*	4459*	4464*
4474*	4478*	4481*	4482	4488*	4494*	4517*	4522*	4527*	4532*	4537*	4543*	4548*
4553*	4558*	4562*	4565*	4566	4572*	4578*	4602*	4607*	4612*	4617*	4622*	4628*
4633*	4638*	4643*	4647*	4650*	4651	4657*	4663*	4686*	4691*	4696*	4701*	4706*
4712*	4717*	4722*	4727*	4731*	4734*	4735	4741*	4747*	4770*	4775*	4780*	4785*
4790*	4796*	4801*	4806*	4811*	4815*	4818*	4819	4825*	4831*	4862*	4863*	4864*
4872*	4878*	4884*	4890*	4896*	4903*	4909*	4914*	4919*	4922*	4923	4930*	4934*
4935	4942*	4943	4982*	4983*	4984	4992*	4998*	5004*	5010*	5016*	5023*	5029*
5034*	5039*	5042*	5043	5050*	5054*	5055	5062*	5063	5102*	5103*	5104	5112*
5118*	5124*	5130*	5136*	5143*	5149*	5154*	5159*	5162*	5163	5170*	5174*	5175
5182*	5183	5222*	5223*	5224	5232*	5238*	5244*	5250*	5256*	5263*	5269*	5274*
5279*	5282*	5283	5290*	5294*	5295	5302*	5303	5342*	5343*	5344*	5347*	5349
5357*	5363*	5369*	5375*	5381*	5388*	5394*	5399*	5405*	5411*	5415*	5416	5423*
5426*	5428	5467*	5468*	5469*	5472*	5474	5482*	5488*	5494*	5500*	5506*	5513*
5519*	5524*	5530*	5536*	5540*	5541	5548*	5551*	5553	5592*	5593*	5594*	5597*
5599	5607*	5613*	5619*	5625*	5631*	5638*	5644*	5649*	5655*	5661*	5665*	5666
5673*	5676*	5678	5717*	5718*	5719*	5722*	5724	5732*	5738*	5744*	5750*	5756*
5763*	5769*	5774*	5780*	5786*	5790*	5791	5798*	5801*	5803	5835*	5840*	5845*
5850*	5855*	5861*	5866*	5871*	5876*	5880*	5883*	5884	5890*	5896*	5920*	5925*
5930*	5935*	5940*	5946*	5951*	5956*	5961*	5965*	5968*	5969	5975*	5981*	6453*
6458*	6463*	6468*	6473*	6482*	6487*	6492*	6497*	6501*	6504*	6505	6511*	6517*
6525*	6592*	6597*	6602*	6607*	6612*	6621*	6626*	6631*	6636*	6640*	6643*	6644
6650*	6656*	6662*	6708*	6713*	6718*	6723*	6728*	6737*	6742*	6747*	6752*	6756*
6759*	6760	6766*	6772*	6778*	6824*	6829*	6834*	6839*	6844*	6853*	6858*	6863*
6868*	6872*	6875*	6876	6882*	6888*	6894*	6940*	6945*	6950*	6955*	6960*	6969*
6974*	6979*	6984*	6988*	6991*	6992	6998*	7004*	7010*	7086*	7093	8282	8284
8286	8297											
7334*												

\$ROCHR	060326	8018	8272															
\$RODEC=	***** U	8275																
\$ROLIN	060446	8046	8273															
\$RODOCT	060746	8118	8274															
\$ROSZ =	000C10	8039																
\$RESRE	061144	8190	8276															
\$RATNAD	055266	7343																
\$R2A =	***** U	8277																
\$SAVRE	061106	8174	8275															
\$SCOPE	055426	680	7393															
\$SETUP=	000137	658	679	680	682	684	686	688	689	690	692	720	723	7309				
		7394	7598	7624	7632	7942	8104											
\$STUP =	177777	658																
\$SVLAD	055666	7404	7443															
\$SVPC =	000220	289	294															
\$SWR =	167400	1	11	15	16	17	18	19	20	21	22	363	364	365				
		689	690	692	693	813	848	942	1033	1120	1145	1248	1364	1488				
		1612	1735	1860	1992	2124	2256	2380	2470	2589	2708	2827	2948	3070				
		3192	3314	3436	3558	3680	3802	3924	4049	4174	4299	4417	4507	4592				
		4676	4760	4850	4970	5090	5210	5330	5455	5580	5705	5825	5910	6000				
		6250	6327	6360	6396	6445	6578	6694	6810	6926	7032	7127	7170	7223				
		7260	7303	7309	7336	7342	7344	7385	7386	7387	7388	7389	7395	7407				
		7409	7410	7423	7424	7425	7432	7433	7434	7446	7449	7452	7589	7590				
		7591	7592	7593	7602	7609	7621	7625	7637									
\$SWREG	001236	386	713															
\$SWRMC=	000000	22	23	7389	7390	7413												
\$SWOBT	055740	7419	7453															
\$TESTN	001220	377	7444	8280	8281	8282	8284	8286	8288	8289	8291	8292	8294	8297				
\$TIMES	001200	363	689	813	848	942	1033	1120	1145	1248	1364	1488	1612	1736				
		1860	1992	2124	2256	2380	2470	2589	2708	2827	2948	3070	3192	3314				
		3436	3558	3680	3802	3924	4049	4174	4299	4417	4507	4592	4676	4760				
		4850	4970	5090	5210	5330	5455	5580	5705	5825	5910	6000	6250	6327				
		6360	6396	6445	6578	6694	6810	6926	7032	7127	7170	7223	7260	7309				
		7432	7439	7442	7452													
\$TKB	001146	348	7940	7951	7968	8022	8028											
\$TKS	001144	347	7940	7949	7965	7989	8020	8026										
\$TMP0	001160	355	744	746	748	755	757	760	772	774	776	778	779	780				
		781	782	784														
\$TMP1	001162	356																
\$TMP2	001164	357																
\$TMP3	001166	358																
\$TMP4	001170	359																
\$TMP5	001172	360																
\$TMP6	001174	361																
\$TMP7	001176	362																
\$TN =	000077	1	11	802	813	833	848	924	929	942	1017	1022	1033	1107				
		1112	1126	1133	1145	1234	1238	1248	1326	1345	1364	1466	1469					
		1488	1540	1593	1612	1714	1717	1736	1838	1841	1860	1970	1973	1992				
		2102	2105	2124	2234	2237	2256	2366	2369	2380	2450	2455	2470	2571				
		2574	2589	2690	2693	2708	2809	2812	2827	2928	2931	2948	3050	3053				
		3070	3172	3175	3192	3294	3297	3314	3416	3419	3436	3538	3541	3558				
		3660	3663	3680	3782	3785	3802	3904	3907	3924	4029	4032	4049	4154				
		4157	4174	4279	4282	4299	4404	4407	4417	4493	4497	4507	4577	4582				
		4592	4662	4667	4676	4746	4751	4760	4830	4835	4850	4952	4955	4970				
		5072	5075	5090	5192	5195	5210	5312	5315	5330	5437	5440	5455	5562				
		5565	5580	5687	5690	5705	5812	5815	5825	5895	5900	5910	5980	5988				

MSG	802#	804	833#	835	929#	931	1022#	1024	1112#	1114	1133#	1135	1238#	1240	1345#
	1347	1469#	1471	1593#	1595	1717#	1719	1841#	1843	1973#	1975	2105#	2107	2237#	2239
	2369#	2371	2455#	2457	2574#	2576	2693#	2695	2812#	2814	2931#	2933	3053#	3055	3175#
	3177	3297#	3299	3419#	3421	3541#	3543	3663#	3665	3785#	3787	3907#	3909	4032#	4034
	4157#	4159	4282#	4284	4407#	4409	4497#	4499	4582#	4584	4667#	4669	4751#	4753	4835#
	4837	4955#	4957	5075#	5077	5195#	5197	5315#	5317	5440#	5442	5555#	5567	5690#	5692
	5815#	5817	5900#	5902	5988#	5990	6237#	6239	6318#	6320	6350#	6352	6388#	6390	6426#
	6428	6556#	6558	6672#	6674	6788#	6790	6904#	6906	7020#	7022	7118#	7120	7159#	7161
	7212#	7214	7249#	7251											
MULT	1#	133#													
NEWST	1#	133#	802	833	929	1022	1112	1133	1238	1345	1469	1593	1717	1841	1973
	2105	2237	2369	2455	2574	2693	2812	2931	3053	3175	3297	3419	3541	3663	3785
	3907	4032	4157	4282	4407	4497	4582	4667	4751	4835	4955	5075	5195	5315	5440
	5565	5690	5815	5900	5988	6237	6318	6350	6388	6426	6556	6672	6788	6904	7020
	7118	7159	7212	7249											
POP	1#	133#	7572	7573	7923	8146	8195								
PUSH	1#	133#	7533	7535	7556	7882	8120	8175							
REPORT	1#	133#													
SCC.E	28#	812	847	941	1032	1119	1144	1247	1363	1487	1611	1735	1859	1991	2123
	2255	2379	2469	2588	2707	2826	2947	3069	3191	3313	3435	3557	3679	3801	3923
	4048	4173	4298	4416	4506	4591	4675	4759	4849	4969	5089	5209	5329	5454	5579
	5704	5824	5909	5999	6249	6326	6359	6395	6444	6577	6693	6809	6925	7031	7126
	7169	7222	7259	7307											
SETPRI	1#	133#													
SETTRA	8255#	8264	8265	8266	8267	8269	8271	8272	8273	8274	8275	8276	8277		
SETUP	1#	133#	671												
SKIP	1#	133#	924	1017	1107	1126	1234	1326	1466	1590	1714	1838	1970	2102	2234
	2366	2450	2571	2690	2809	2928	3050	3172	3294	3416	3538	3660	3782	3904	4029
	4154	4279	4404	4493	4577	4662	4746	4830	4952	5072	5192	5312	5437	5562	5687
	5812	5895	5980	6183	6199	6220	6422	6669	6785	6901	7017	7155			
SLASH	1#	133#													
SPACE	133#														
STARS	1#	133#	287	298	300	307	320	369	372	802	811	833	846	929	940
	1022	1031	1112	1118	1133	1143	1238	1246	1334	1343	1345	1362	1469	1486	1593
	1610	1717	1734	1841	1858	1973	1990	2105	2122	2237	2254	2369	2378	2455	2468
	2574	2587	2693	2706	2812	2825	2931	2946	3053	3068	3175	3190	3297	3312	3419
	3434	3541	3556	3663	3678	3785	3800	3907	3922	4032	4047	4157	4172	4282	4297
	4407	4415	4497	4505	4582	4590	4667	4674	4751	4758	4835	4848	4955	4968	5075
	5088	5195	5208	5315	5328	5440	5453	5565	5578	5690	5703	5815	5823	5900	5908
	5988	5998	6237	6248	6318	6325	6350	6358	6388	6394	6426	6443	6556	6576	6672
	6692	6788	6808	6904	6924	7020	7030	7118	7125	7159	7168	7212	7221	7249	7258
	7299	7381	7516	7528	7585	7638	7647	7716	7795	7872	7939	7942	8010	8039	8106
	8159	8205	8213	8234											
SWRSU	1#	133#	694#												
TRMTRP	8255#														
TYPBIN	1#	133#													
TYPDEC	1#	133#	7322	7329											
TYPNAM	1#	133#	716												
TYPNUM	1#	133#													
TYPOCS	1#	133#													
TYPOCT	1#	133#	740	7960											
TYPTXT	1#	133#	7318	7325											
WRTDB	318#	6579	6695	6811	6927										
WTRG1	318#	2381	4508	4593	4677	4761	5826	5911							
WTRG2	318#	1365	1489	1613	1737	1861	1993	2125	2257	2471	2590	2709	2828	2949	3071
	3193	3315	3437	3559	3681	3803	3925	4050	4175	4300	4851	4971	5091	5211	5331

JO1

BGE	7440														
BGT	7314	7855			8036	8131									
BHI	747	777	7098	7426											
BHIS	7704														
BIC	751	759	783	786	1377	1501	1625	1749	1874	1968	2006	2100	2138	2232	2270
	2364	2961	3083	3205	3327	3449	3571	3693	3815	3938	4027	4063	4152	4188	4277
	4313	4402	4863	4983	5103	5223	5343	5468	5593	5718	7311	7650	7845	7952	7969
	7996	8023	8029	8037	8140										
BIS	760	899	992	1086	1207	1308	1436	1560	1684	1808	1875	1933	2007	2065	2139
	2197	2271	2329	2438	2540	2659	2778	2897	3020	3142	3264	3386	3508	3630	3752
	3874	3939	3997	4064	4122	4189	4247	4314	4372	4481	4565	4650	4734	4818	4922
	5042	5162	5282	5344	5347	5426	5469	5472	5551	5594	5597	5676	5719	5722	5801
	5883	5968	6504	6643	6759	6875	6991	7850	7851	7910	7911	8003			
BISB	784														
BIT	897	990	1084	1205	1306	1434	1558	1682	1806	1931	1945	2063	2077	2195	2209
	2327	2341	2436	2538	2657	2776	2895	3018	3140	3262	3384	3506	3628	3750	3872
	3995	4120	4245	4370	4479	4563	4648	4732	4816	4920	5040	5160	5280	5345	5424
	5470	5549	5595	5674	5720	5799	5881	5966	6502	6641	6757	6873	6989	7372	7395
	7409	7427	7434	7520	7602	7609	7625	7701							
BITB	711	7540	7739	7744	7776										
BLE	7689														
BLO	775	7099	8062												
BLOS	758	8050													
BLT	7415	7767	7856	7899	7915	7993	8034	8133							
BMI	7906														
BNE	677	701	719	725	729	738	795	821	1234	6006	6015	6036	6180	6196	6218
	6231	6478	6617	6733	6849	6965	7040	7062	7080	7133	7358	7373	7396	7435	7539
	7545	7548	7565	7610	7615	7633	7677	7738	7745	7747	7755	7763	7777	7784	7846
	7904	7948	7954	7974	7981	7988	8025	8031	8054	8056	8072	8076	8086	8218	8220
BPL	7622	7732	7781	7844	7890	7920	7950	7966	8021	8027					
BR	659	662	703	731	734	822	1948	2080	2212	2344	6183	6199	6220	6273	6304
	6309	7081	7101	7189	7238	7283	7289	7319	7326	7359	7398	7404	7407	7421	7430
	7433	7531	7553	7620	7683	7698	7734	7760	7770	7779	7786	7822	7837	7858	7901
	7918	7977	8004	8006	8032	8065	8074	8080	8082	8142	8155	8211			
CLC	1463	1965	2568	3047	3535	4024	4949	5434	6666						
CLR	664	675	689	690	710	715	762	791	796	828	861	866	871	882	887
	892	907	954	959	964	975	980	985	1000	1053	1058	1069	1074	1079	1093
	1108	1128	1147	1169	1174	1179	1190	1195	1200	1214	1253	1270	1275	1280	1291
	1296	1301	1315	1327	1373	1342	1398	1404	1417	1422	1428	1444	1456	1497	1516
	1522	1528	1541	1546	1552	1568	1580	1614	1621	1640	1646	1652	1665	1670	1676
	1692	1704	1738	1745	1764	1770	1776	1789	1794	1800	1816	1828	1862	1870	1883
	1889	1895	1901	1914	1919	1925	1941	2002	2015	2021	2027	2033	2046	2051	2057
	2073	2126	2127	2134	2147	2153	2159	2165	2178	2183	2189	2205	2258	2259	2266
	2279	2285	2291	2297	2310	2315	2321	2337	2385	2395	2400	2405	2410	2421	2426
	2431	2445	2496	2502	2508	2521	2526	2532	2548	2615	2621	2627	2640	2645	2651
	2667	2710	2734	2740	2746	2759	2764	2770	2786	2829	2853	2859	2865	2878	2883
	2889	2905	2957	2976	2982	2988	3001	3006	3012	3028	3040	3079	3098	3104	3110
	3123	3128	3134	3150	3162	3194	3201	3220	3226	3232	3245	3250	3256	3272	3284
	3316	3323	3342	3348	3354	3367	3372	3378	3394	3406	3445	3464	3470	3476	3489

	7899	7894	7897	7917	7923	7924	7925	7926	7927	7929	7930	7960	7984	7989	8018
	8019	8046	8048	8059	8090	8091	8092	8093	8112	8119	8120	8121	8122	8124	8125
	8144	8145	8146	8147	8148	8175	8176	8177	8178	8179	8180	8181	8182	8183	8184
	8191	8192	8193	8194	8195	8196	8197	8198	8199	8200	8207	8208	8209	8216	8221
	8222	8223	8226	8240	8241	8245	8251	8252							
MOVE	691	732	763	787	7412	7444	7448	7529	7530	7532	7608	7616	7649	7672	7746
	7774	7782	7819	7820	7823	7824	7825	7829	7832	7833	7852	7892	7895	7909	7912
	7921	7951	7968	8022	8028	8052	8057	8063	8068	8083	8128	8243			
NEG	7833	7991													
NOP	6023	6025	6212	6266	6289	6297	6338	6373	7183	7199	7237	7272	7275	7238	7339
	7340														
RESET	665	851	1104	7336											
ROL	1464	1588	1712	1966	2098	2230	2569	2688	2807	3048	3170	3292	3414	3536	3653
	3780	4025	4150	4275	4950	5070	5190	5435	5560	5685	6667	6783	6899	7836	7838
	7839	7840	7842	8135	8137	8139									
ROR	1836	2362	2926	3902	4400	5310	5810	7015							
RTI	669	705	801	6020	6029	6204	6210	6263	6269	6286	6294	6300	6334	6341	6370
	6376	7178	7186	7195	7201	7234	7268	7278	7374	7451	7525	7636	7751	7864	7931
	7930	8038	8094	8149	8185	8201	8253								
RTS	7364	7574	7713	7791	8246										
SEC	1587	1711	1835	2097	2229	2361	2687	2806	2925	3169	3291	3413	3657	3779	3901
	4149	4274	4399	5069	5189	5309	5559	5684	5809	6782	6898	7014			
SUB	6172	6186	6223	6254	7549	7607	7898								
TRAP	8255	8264	8265	8266	8267	8269	8271	8272	8273	8274	8275	8276	8277		
TST	724	788	818	819	1233	6399	6540	7103	7182	7226	7271	7402	7422	7436	7544
	7562	7564	7621	7628	7674	7676	7685	7692	7707	7748	7756	7778	7847	7903	7913
	7982	7997	8055	8066	8099	8143	8150	8242							
TSTB	7423	7522	7536	7547	7560	7673	7731	7780	7905	7919	7949	7965	8020	8026	
.ASCII	366	367	8430	8561	8653	8661	8677	8803	8824						
.ASCIZ	365	368	736	7321	7328	8098	8099	8100	8102	8230	8398	8402	8404	8409	8413
	8414	8425	8428	8433	8435	8438	8441	8444	8448	8452	8456	8460	8463	8466	8470
	8476	8481	8488	8495	8505	8516	8523	8531	8539	8548	8553	8566	8571	8577	8582
	8588	8595	8600	8606	8612	8618	8624	8629	8637	8641	8647	8658	8665	8668	8683
	8687	8595	8700	8705	8714	8722	8729	8736	8743	8750	8756	8762	8770	8775	8780
	8787	8794	8811	8815	8831	8838	8841	8845	8849	8853	8956	8861	8865	8871	8875
	8879	8882	8885	8891	8897	8900	8903	8908	8912	8917	8920	8926	8930	8937	8941
	8946	8950													
.BLKB	8097														
.BLKW	7936	8955													
.BYTE	327	328	333	334	342	343	351	352	353	354	384	385	395	396	403
	404	406	407	409	410	7344	7575	7576	7577	7618	7619	7865	7866	7867	7869
	8095	8096	8303	8305	8307	8309	8311	8313	8315	8317	8319	8321	8323	8325	8327
	8329	8331	8333	8335	8337	8339	8341	8343	8345	8347	8349	8351	8353	8355	8357
	8359	8361	8363	8365	8367	8369	8371	8373	8375	8377	8379	8381	8383	8385	8387
	8389	8391	8393	8395	8956	8957	8958	8959	8960	8961	8962	8963	8964	8965	8966
	8967	8968	8969	8970	8971	8972	8973	8974	8975	8976	8977	8978	8979	8980	8981
	8982	8983	8984	8985	8986	8987	8988	8989	8990	8991	8992	8993	8994	8995	8996
	8997	8998	8999	9000	9001	9002	9003	9004	9005	9006	9007	9008	9009	9010	9011
	9012	9013	9014	9015	9016	9017	9018	9019	9020	9021					
.DSABL	8007														
.ENABL	1	7940													
.END	9022														
.ENDC	6														
	321	325	21	22	23	27	119	133	282	288	292	294	299	301	308
	413	414	327	355	363	364	365	366	370	373	395	403	406	409	412
	692	694	415	416	417	420	658	678	679	682	684	686	688	689	690
			715	720	722	728	734	736	803	804	811	812	813	814	834

835	846	847	848	849	853	925	930	931	940	941	942	943	946	1018
1023	1024	1031	1032	1033	1034	1108	1113	1114	1118	1119	1120	1121	1127	1134
1135	1143	1144	1145	1146	1235	1239	1240	1246	1247	1248	1249	1327	1335	1344
1346	1347	1362	1363	1364	1365	1366	1367	1374	1377	1378	1390	1396	1402	1408
1415	1432	1442	1448	1457	1462	1463	1464	1465	1467	1470	1471	1486	1487	1488
1499	1490	1491	1498	1501	1502	1514	1520	1526	1532	1539	1556	1566	1572	1581
1586	1587	1588	1589	1591	1594	1595	1610	1611	1612	1613	1614	1615	1622	1625
1626	1638	1644	1650	1656	1663	1680	1690	1696	1705	1710	1711	1712	1713	1715
1718	1719	1734	1735	1736	1737	1738	1739	1746	1749	1750	1762	1768	1774	1780
1787	1804	1814	1820	1829	1834	1835	1837	1839	1842	1843	1858	1859	1860	1861
1864	1871	1876	1881	1887	1893	1899	1905	1912	1923	1939	1950	1959	1964	1965
1967	1969	1971	1974	1975	1990	1991	1992	1993	1996	2003	2008	2013	2019	2025
2031	2037	2044	2061	2071	2082	2091	2096	2098	2099	2101	2103	2106	2107	2122
2123	2124	2125	2128	2135	2140	2145	2151	2157	2163	2169	2176	2193	2203	2214
2223	2228	2230	2231	2233	2235	2238	2239	2254	2255	2256	2257	2260	2267	2272
2277	2283	2289	2295	2301	2308	2325	2335	2346	2355	2360	2363	2365	2367	2370
2371	2378	2379	2380	2381	2451	2456	2457	2468	2469	2470	2471	2472	2473	2479
2482	2494	2500	2506	2512	2519	2536	2546	2552	2567	2568	2569	2570	2572	2575
2576	2587	2588	2589	2590	2591	2592	2598	2601	2613	2619	2625	2631	2638	2655
2665	2671	2686	2687	2688	2689	2691	2694	2695	2706	2707	2708	2709	2710	2711
2717	2720	2732	2738	2744	2750	2757	2774	2784	2790	2805	2806	2807	2808	2810
2813	2814	2825	2826	2827	2828	2829	2830	2836	2839	2851	2857	2863	2869	2876
2893	2903	2909	2924	2925	2927	2929	2932	2933	2946	2947	2948	2949	2950	2951
2958	2961	2962	2974	2980	2986	2992	2999	3016	3026	3032	3041	3046	3047	3048
3049	3051	3054	3055	3068	3069	3070	3071	3072	3073	3080	3083	3084	3096	3102
3108	3114	3121	3138	3148	3154	3163	3168	3169	3170	3171	3173	3176	3177	3190
3191	3192	3193	3194	3195	3202	3205	3206	3218	3224	3230	3236	3243	3260	3270
3276	3285	3290	3291	3292	3293	3295	3298	3299	3312	3313	3314	3315	3316	3317
3324	3327	3328	3340	3346	3352	3358	3365	3382	3392	3398	3407	3412	3413	3414
3415	3417	3420	3421	3434	3435	3436	3437	3438	3439	3446	3449	3450	3462	3468
3474	3480	3487	3504	3514	3520	3529	3534	3535	3536	3537	3539	3542	3543	3556
3557	3558	3559	3560	3561	3568	3571	3572	3584	3590	3596	3602	3609	3626	3636
3642	3651	3656	3657	3658	3659	3661	3664	3665	3678	3679	3680	3681	3682	3683
3690	3693	3694	3706	3712	3718	3724	3731	3748	3758	3764	3773	3778	3779	3780
3781	3783	3786	3787	3800	3801	3802	3803	3804	3805	3812	3815	3816	3828	3834
3840	3846	3853	3870	3880	3886	3895	3900	3901	3903	3905	3908	3909	3922	3923
3924	3925	3926	3928	3935	3938	3940	3952	3958	3964	3970	3976	3993	4003	4009
4018	4023	4024	4025	4026	4028	4030	4033	4034	4047	4048	4049	4050	4051	4053
4060	4063	4065	4077	4083	4089	4095	4101	4118	4128	4134	4143	4148	4149	4150
4151	4153	4155	4158	4159	4172	4173	4174	4175	4176	4178	4185	4188	4190	4202
4208	4214	4220	4226	4243	4253	4259	4268	4273	4274	4275	4276	4278	4280	4283
4284	4297	4298	4299	4300	4301	4303	4310	4313	4315	4327	4333	4339	4345	4351
4368	4378	4384	4393	4398	4399	4401	4403	4405	4408	4409	4415	4416	4417	4418
4494	4498	4499	4505	4506	4507	4508	4578	4583	4584	4590	4591	4592	4593	4663
4668	4669	4674	4675	4676	4677	4747	4752	4753	4758	4759	4760	4761	4831	4836
4837	4848	4849	4850	4851	4852	4853	4860	4863	4864	4876	4882	4888	4894	4900
4907	4918	4928	4934	4943	4948	4949	4950	4951	4953	4956	4957	4963	4969	4970
4971	4972	4973	4980	4983	4994	4996	5002	5008	5014	5020	5027	5038	5048	5054
5063	5068	5069	5070	5071	5073	5076	5077	5088	5089	5090	5091	5092	5093	5100
5103	5104	5116	5122	5128	5134	5140	5147	5158	5168	5174	5183	5188	5189	5190
5191	5193	5196	5197	5208	5209	5210	5211	5212	5213	5220	5223	5224	5236	5242
5248	5254	5260	5267	5278	5288	5294	5303	5308	5309	5311	5313	5316	5317	5328
5329	5330	5331	5332	5333	5340	5343	5349	5361	5367	5373	5379	5385	5392	5409
5415	5428	5433	5434	5435	5436	5438	5441	5442	5453	5454	5455	5456	5457	5458
5465	5468	5474	5486	5492	5498	5504	5510	5517	5534	5540	5553	5558	5559	5560
5561	5563	5566	5567	5578	5579	5580	5581	5582	5583	5590	5593	5599	5611	5617

	5182	5189	5193	5196	5197	5209	5210	5212	5302	5309	5311	5313	5316	5317	5329
	5330	5332	5333	5423	5428	5434	5435	5438	5441	5442	5454	5455	5457	5458	5548
	5553	5559	5563	5566	5567	5579	5580	5582	5673	5678	5684	5688	5691	5692	5704
	5705	5707	5798	5803	5809	5811	5813	5816	5817	5824	5825	5836	5901	5902	5909
	5910	5981	5989	5990	5999	6000	6184	6200	6221	6238	6239	6249	6250	6319	6320
	6326	6327	6351	6352	6359	6360	6389	6390	6395	6396	6423	6427	6428	6444	6445
	6557	6558	6577	6578	6581	6666	6667	6670	6673	6674	6693	6694	6696	6697	6782
	6786	6789	6790	6809	6810	6812	6898	6902	6905	6906	6925	6926	6928	7016	7018
	7021	7022	7031	7032	7119	7120	7126	7127	7156	7160	7161	7169	7170	7213	7214
	7222	7223	7250	7251	7259	7260	7300	7304	7308	7314	7317	7344	7382	7408	7411
	7423	7425	7452	7453	7517	7529	7586	7588	7602	7632	7637	7639	7648	7717	7796
	7873	7940	7943	8011	8013	8018	8039	8040	8050	8081	8098	8107	8160	8206	8214
	8235	8241													
.IFT	736	7321	7328	7433	7612	8013	8018	8130	8150	8157					
.IFTF	736	7321	7328	7431	7611	7958	8011	8014	8126	8134	8156				
.IIF	1	6	11	15	16	17	19	22	23	277	369	373	679	682	688
	689	690	692	693	720	741	7301	7308	7309	7323	7330	7344	7345	7385	7386
	7387	7388	7389	7390	7394	7413	7432	7433	7449	7452	7453	7589	7590	7591	7592
	7593	7598	7624	7632	7637	7793	7940	7961	8089	8098	8104	8157	8263	8264	8265
.IRP	8266	8267	8269	8271	8272	8273	8274	8275	8276	8277					
	658	802	833	929	1022	1112	1133	1238	1345	1469	1593	1717	1841	1973	2105
	2237	2369	2455	2574	2693	2812	2931	3053	3175	3297	3419	3541	3663	3785	3907
	4032	4157	4292	4407	4497	4582	4667	4751	4835	4955	5075	5195	5315	5440	5565
	5690	5815	5900	5988	6237	6318	6350	6388	6426	6556	6672	6788	6904	7020	7118
	7159	7212	7249	7414	7454	7455	7456	7457	7458	7459	7460	7461	7462	7463	7464
	7465	7466	7467	7468	7469	7470	7471	7472	7473	7474	7475	7476	7477	7478	7479
	7480	7481	7482	7483	7484	7485	7486	7487	7488	7489	7490	7491	7492	7493	7494
	7495	7496	7497	7498	7499	7500	7501	7502	7503	7504	7505	7506	7507	7508	7509
	7510	7511	7512	7513	7514	7515	7534	7535	7556	7572	7573	7883	7923	8120	8146
	8175	8195													
.LIST	1	22	133	277	355	356	357	358	359	360	361	362	363	370	373
	658	694	720	723	736	802	813	833	848	929	942	1022	1033	1112	1120
	1133	1145	1238	1248	1345	1364	1469	1488	1593	1612	1717	1736	1841	1860	1973
	1992	2105	2124	2237	2256	2369	2380	2455	2470	2574	2589	2693	2708	2812	2827
	2931	2948	3053	3070	3175	3192	3297	3314	3419	3436	3541	3558	3663	3680	3785
	3802	3907	3924	4032	4049	4157	4174	4282	4299	4407	4417	4497	4507	4582	4592
	4667	4676	4751	4760	4835	4850	4955	4970	5075	5090	5195	5210	5315	5330	5440
	5455	5565	5580	5690	5705	5815	5825	5900	5910	5988	6000	6237	6250	6318	6327
	6350	6360	6388	6396	6426	6445	6556	6578	6672	6694	6788	6810	6904	6926	7020
	7032	7118	7127	7159	7170	7212	7223	7249	7260	7308	7321	7328	7336	7389	7453
	7455	7456	7457	7458	7459	7460	7461	7462	7463	7464	7465	7466	7467	7468	7469
	7470	7471	7472	7473	7474	7475	7476	7477	7478	7479	7480	7481	7482	7483	7484
	7485	7486	7487	7488	7489	7490	7491	7492	7493	7494	7495	7496	7497	7498	7499
	7500	7501	7502	7503	7504	7505	7506	7507	7508	7509	7510	7511	7512	7513	7514
	7515	7516	7632	8039	8255	8263	8264	8265	8266	8267	8268	8269	8270	8271	8272
	8273	8274	8275	8276	8277	8278									
.MACRO	1	23	318	710	802	833	929	1022	1112	1133	1238	1345	1469	1593	1717
	1841	1973	2105	2237	2369	2455	2574	2693	2812	2931	3053	3175	3297	3419	3541
	3663	3785	3907	4032	4157	4282	4407	4497	4582	4667	4751	4835	4955	5075	5195
	5315	5440	5565	5690	5815	5900	5988	6237	6318	6350	6388	6426	6556	6672	6788
	6904	7020	7118	7159	7212	7249	8255								
.MCALL	1	133	370	694	723										
.MEXIT	419														
.MLIST	1	22	133	277	355	356	357	358	359	360	361	362	363	370	373
	658	694	720	723	736	802	813	833	848	929	942	1022	1033	1112	1120
	1133	1145	1238	1248	1345	1364	1469	1488	1593	1612	1717	1736	1841	1860	1973

	1992	2105	2124	2237	2256	2369	2380	2455	2470	2574	2589	2633	2708	2812	2827
	2931	2948	3053	3070	3175	3192	3297	3314	3419	3436	3541	3558	3663	3680	3785
	3902	3907	3924	4032	4049	4157	4174	4282	4299	4407	4417	4497	4507	4582	4592
	4667	4676	4751	4760	4835	4850	4955	4970	5075	5090	5195	5210	5315	5330	5440
	5455	5565	5580	5690	5705	5815	5825	5900	5910	5988	6000	6237	6250	6318	6327
	6350	6360	6388	6396	6426	6445	6556	6578	6672	6694	6788	6610	6904	6926	7020
	7032	7118	7127	7159	7170	7212	7223	7249	7260	7308	7321	7328	7336	7389	7453
	7455	7456	7457	7458	7459	7460	7461	7462	7463	7464	7465	7466	7467	7468	7469
	7470	7471	7472	7473	7474	7475	7476	7477	7478	7479	7480	7481	7482	7483	7484
	7485	7486	7487	7488	7489	7490	7491	7492	7493	7494	7495	7496	7497	7498	7499
	7500	7501	7502	7503	7504	7505	7506	7507	7508	7509	7510	7511	7512	7513	7514
	7515	7516	7632	8039	8255	8263	8264	8265	8266	8267	8268	8269	8270	8271	8272
	8273	8274	8275	8276	8277	8278	8278	8300	8396	8423	8514				
.PAGE	318	420	601	656	802	7297	8278	8300	8396	8423	8514				
.REPT	277	355	6042	7454											
.SBTTL	11	23	140	159	176	192	209	228	244	261	271	280	285	296	318
	370	420	601	620	639	656	672	716	723	802	831	833	929	1022	1112
	1133	1238	1335	1345	1469	1593	1717	1841	1973	2105	2237	2369	2455	2574	2693
	2812	2931	3053	3175	3297	3419	3541	3663	3785	3907	4032	4157	4282	4407	4497
	4582	4667	4751	4835	4955	5075	5195	5315	5440	5565	5690	5815	5900	5986	5988
	6237	6318	6350	6386	6386	6426	6556	6672	6788	6904	7020	7118	7159	7212	7249
	7297	7347	7366	7379	7518	7526	7583	7639	7714	7793	7870	7937	8104	8157	8203
	8232	8255	8278	8300	8396	8423	8514								
.TITLE	1														
.WORD	277	278	279	293	312	313	314	315	316	317	326	329	330	331	332
	335	336	337	338	339	340	341	344	345	346	355	356	357	358	359
	360	361	362	375	376	377	378	379	380	381	382	386	387	388	401
	405	408	411	412	413	414	415	416	417	603	604	605	606	607	608
	609	610	611	612	613	614	615	616	617	618	622	623	624	625	626
	627	628	629	630	631	632	633	634	635	636	637	641	642	643	647
	648	649	650	651	652	653	654	655	7313	7316	7343	7454	7455	7456	7457
	7458	7459	7460	7461	7462	7463	7464	7465	7466	7467	7468	7469	7470	7471	7472
	7473	7474	7475	7476	7477	7478	7479	7480	7481	7482	7483	7484	7485	7486	7487
	7488	7489	7490	7491	7492	7493	7494	7495	7496	7497	7498	7499	7500	7501	7502
	7503	7504	7505	7506	7507	7508	7509	7510	7511	7512	7513	7514	7515	7558	7660
	7665	7696	7743	7790	7869	8153	8156	8229	8262	8280	8281	8282	8284	8286	8288
	8289	8291	8292	8294	8297	8302	8304	8306	8308	8310	8312	8314	8316	8318	8320
	8322	8324	8326	8328	8330	8332	8334	8336	8338	8340	8342	8344	8346	8348	8350
	8352	8354	8356	8358	8360	8362	8364	8366	8368	8370	8372	8374	8376	8378	8380
	8382	8384	8386	8388	8390	8392	8394								

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DZR6AB/CRF/SOL=SYSMAC.C1 DZR6AB.P11
RUN-TIME: 90 127 21 SECONDS
RUN-TIME RATIO: 337/239=1.4
CORE USED: 40K (79 PAGES)

