

# RK611

DISKLESS CONTR DIAG 2  
MD-11-DZR6B-B

EP-DZR6B-B-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 2

NOV 1976  
**digital**  
MADE IN USA

The microfiche card contains 24 frames arranged in a 4x6 grid. Each frame displays a technical diagram or data table. The diagrams are high-contrast and appear to be organized into columns and rows, possibly representing different stages or components of a system. The card is dark blue with white markings.

**RK611**

DIAGNOSTIC PART 2  
MD-11-DZR6B-B

EP-DZR6B-B-DL-A  
COPYRIGHT © 1976  
FICHE 2 OF 2

NOV 1976  
**digital**  
MADE IN USA

B01

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZR6B-B  
PRODUCT NAME: RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 2  
DATE: AUG, 1976  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: ROY 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 LICENCE 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 2 TEST THE LOADING OF THE DRIVE BUS MESSAGES BY EXECUTING CLASS A COMMANDS. SOME TESTS EXECUTE COMMANDS PARTIALLY MAINTENANCE MODE AND PARTIALLY AT NORMAL SPEED TO FOOL THE CONTROLLER AND FORCE ERRORS. THIS PROGRAM DOES NOT REQUIRE THE PRESENCE OF AN RK06 DRIVE.

## 2.0 REQUIREMENTS

## 2.1 EQUIPMENT

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

## 2.2 PRELIMINARY PROGRAMS

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 1  
MAINDEC-11-DZR6A

## 3.0 OPERATING PROCEDURES

## 3.1 LOADING PROCEDURE

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

## 3.2 STARTING PROCEDURE

LOCATION 200	- START PROGRAM
LOCATION 204	- RESTART PROGRAM
LOCATION 214	- REQUEST BUS ADDRESS, VECTOR ADDRESS, AND PRIORITY MODIFICATION

## 3.3 OPTIONAL SWITCH SETTINGS

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

## 3.5 RUN TIME

FIRST PASS                   7 SECONDS  
SUBSEQUENT PASSES           2 MINUTES

## 4.0 OPERATING PROCEDURES

THE PROGRAM IS EXECUTED BY STARTING AT THE APPROPRIATE ADDRESS.

## 5.0 PROGRAM DESCRIPTION

## \*\*DRIVE MESSAGE LOADING

## TEST 1 FIRST COMMAND IN MAINT MODE

INITIALIZE RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN MODE. ISSUE SELECT DRIVE. WAIT AND MAKE SURE CS1 REMAINS THE SAME. CLOCK IN MESSAGES A AND B. MAKE SURE CORRECT MSG ARE LOADED. CHECKING IS DONE A FIELD AT A TIME.

## TEST 2 DRIVE SELECT BITS LOADING FOR DRIVE MESS.

INITIALIZE RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 2 WITH ZERO. LOAD COMMAND AND STATUS REGISTER WITH A SELECT COMMAND. CLOCK IN MESSAGES A AND B INTO SHIFT REGISTER. MAKE SURE CORRECT MESSAGES ARE LOADED. REPEAT FOR DRIVE SELECT = 1-17.

# F01

## TEST 3 FORMAT BIT LOADING TO FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND AND 24 SECTOR MODE FORMAT. MAKE SURE CORRECT MESSAGE IS LOADED.

## TEST 4 HEAD SELECT BITS LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD TRACK ADDRESS WITH ZERO. LOAD COMMAND AND STATUS REGISTER 2 WITH ZERO. LOAD COMMAND AND STATUS REGISTER WITH SELECT COMMAND. CLOCK IN MESSAGE A AND B INTO SHIFT REGISTER. MAKE SURE CORRECT MESSAGE IS LOADED. REPEAT FOR TRACK ADDRESS = 1-7.

## TEST 5 MESSAGE SELECT BITS LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE AND ZERO IN MESSAGE SELECT BITS. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND. CLOCK IN MESSAGE A AND B INTO SHIFT REGISTER. MAKE SURE CORRECT MESSAGE IS LOADED. REPEAT FOR MESSAGE SELECT = 1-17.

## TEST 6 CLEAR DRIVE COMMAND LOADING FOR DRIVE MESS

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH A DRIVE CLEAR. CLOCK MESSAGE A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY. REPEAT FOR 24 SECTOR FORMAT.

## TEST 7 UNLOAD COMMAND LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH AN UNLOAD COMMAND. CLOCK IN MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY. REPEAT FOR 24 SECTOR FORMAT.

## TEST 10 PACK ACKNOWLEDGE COMMAND LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH A PACK ACKNOWLEDGE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY. REPEAT FOR 24 SECTOR FORMAT.

# GO1

## TEST 11 RECALIBRATE COMMAND LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH A RECALIBRATE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY.

## TEST 12 START SPINDLE COMMAND LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH A START SPINDLE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY.

## TEST 13 SEEK AND CYLINDER ADD 0-777 LOADING FOR DRIVE MESS

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD ZERO IN CYLINDER ADDRESS. LOAD COMMAND AND STATUS REGISTER 1 WITH A SEEK COMMAND. CLOCK IN MESSAGE A AND B INTO SHIFT REGISTER. MAKE SURE CORRECT MESSAGE IS LOADED. REPEAT FOR CYLINDER = 1-777.

## TEST 14 SEEK AND CYLINDER BIT 9 AND RK06 FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD 1000 IN CYLINDER ADDRESS. LOAD COMMAND AND STATUS REGISTER 1 WITH A SEEK COMMAND. CLOCK IN MESSAGE A AND B INTO SHIFT REGISTERS. MAKE SURE CYLINDER BIT 9 IN MESSAGE IN RESET. REPEAT FOR CYLINDER = 1400.

## TEST 15 SEEK AND CYLINDER ADD 0,777-1777 LOADING FOR DRIVE MESS

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD 0 IN CYLINDER ADDRESS. LOAD COMMAND AND STATUS REGISTER 1 WITH SEEK COMMAND AND CDT SET. CLOCK IN MESSAGE A AND B INTO SHIFT REGISTER. MAKE SURE CYLINDER CORRECT. REPEAT FOR CYLINDER = 777-1777.

## TEST 16 OFFSET COMMAND LOADING FOR DRIVE MESS.

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD OFFSET REGISTER WITH 0. LOAD COMMAND AND STATUS REGISTER 1 WITH AN OFFSET. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY. REPEAT FOR OFFSET REGISTER = 1-377.

2  
↑

# H01

## TEST 17 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 1)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

## TEST 20 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 2)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND AND STATUS REGISTER 1 WITH A PACK ACKNOWLEDGE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

## TEST 21 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 3)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND AND STATUS REGISTER 1 WITH A CLEAR DRIVE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

## TEST 22 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 4)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND AND STATUS REGISTER 1 WITH AN UNLOAD. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

## TEST 23 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 5)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND AND STATUS REGISTER 1 WITH A START SPINDLE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

# I01

## TEST 24 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 6)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND AND STATUS REGISTER 1 WITH A RECALIBRATE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

## TEST 25 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 1)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH A PACK ACKNOWLEDGE. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

## TEST 26 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 2)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH A DRIVE CLEAR. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

## TEST 27 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 3)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH AN UNLOAD. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

## TEST 30 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 4)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH A START SPINDLE. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

# J01

## TEST 31 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 5)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH A RECALIBRATE. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

## TEST 32 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 6)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH A OFFSET. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

## TEST 33 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 7)

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD COMMAND AND STATUS REGISTER 1 WITH A SEEK. CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE MESSAGE SELECT BITS ARE CLEARED.

## \*\*DRIVE MESSAGE LOOPBACK AND PARITY GENERATION TESTS

### TEST 34 DRIVE MESSAGE LOOPBACK

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE INDICATING MESSAGE 3. LOAD COMMAND STATUS REGISTER FOR DRIVE 5. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND. CLOCK 4 BITS THROUGH THE DRIVE MESSAGE LOOPBACK. VERIFY THAT BITS ARE INDEED LOOPED BACK.

### TEST 35 DRIVE MESSAGE SHIFT

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS WITH 441. LOAD HEAD ADDRESS WITH 1. LOAD COMMAND AND STATUS REGISTER 1 WITH A SEEK IN 24 SECTOR MODE. CLOCK 8 BITS THROUGH THE DRIVE MESSAGE LOOPBACK. VERIFY THAT BITS ARE SHIFTED PROPERLY.

# K01

## TEST 36 DRIVE MESSAGE PARITY PRECONDITIONING

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND. CLOCK ALL 16 BITS THROUGH THE DRIVE MESSAGE LOOPBACK. VERIFY PARITY HAS BEEN PRECONDITIONED PROPERLY. REPEAT FOR BAD PARITY GENERATION.

## TEST 37 ODD DRIVE MESSAGE PARITY GENERATION

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE AND MESSAGE SELECT = 1. LOAD COMMAND AND STATUS REGISTER 2 WITH DRIVE SELECT = 1. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND. VERIFY THAT PARITY HAS BEEN GENERATED CORRECTLY. REPEAT FOR MESSAGE SELECT = DRIVE SELECT = 2-17.

## TEST 40 DRIVE MESSAGE PARITY INTERACTION

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 2 WITH DRIVE SELECT = 1. LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND. VERIFY THAT THE CORRECT PARITY IS GENERATED FOR BOTH MESSAGES. REPEAT FOR MESSAGE SELECT = 1 AND DRIVE SELECT = 0.

## TEST 41 EVEN DRIVE MESSAGE PARITY GENERATION

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE AND MESSAGE SELECT = 1 AND BAD PARITY SET. LOAD COMMAND AND STATUS REGISTER 2 WITH DRIVE SELECT = 1. LOAD COMMAND AND STATUS REGISTER SELECT COMMAND. VERIFY THAT EVEN PARITY IS GENERATED. REPEAT FOR MESSAGE SELECT = DRIVE SELECT = 2-17.

# L01

## \*\*CLASS A COMMAND EXECUTION

### TEST 42 RELEASE COMMAND IN DIAGNOSTIC MODE

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND  
STATUS REGISTER 2 WITH DRIVE SELECT = 10. LOAD  
COMMAND AND STATUS REGISTER 1 WITH A SELECT.  
CLOCK COMMAND TO COMPLETION. MAKE SURE UNIT  
FIELD ERROR DOES NOT SET (SACK HIGH). REPEAT FOR  
DRIVE SELECT = 11-17.

### TEST 43 SELECT COMMAND IN DIAGNOSTIC MODE

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND  
STATUS REGISTER 2 WITH DRIVE SELECT = 0. LOAD  
COMMAND AND STATUS REGISTER 1 WITH A SELECT.  
CLOCK COMMAND TO COMPLETION. MAKE SURE MESSAGE SHIFT IS  
NOT DONE DURING THE RECEIVE CYCLE OF DRIVE MESSAGE.  
MAKE SURE NO ERRORS SET. REPEAT FOR DRIVE SELECT = 1-7.

### TEST 44 RELEASE COMMAND IN NORMAL MODE

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
LOAD COMMAND AND STATUS REGISTER 2 WITH DRIVE SELECT = 10.  
LOAD COMMAND AND STATUS REGISTER 1 WITH A SELECT.  
MAKE SURE NO ERRORS OCCUR. REPEAT FOR DRIVE  
SELECT = 11-17

### TEST 45 INTERRUPT AT COMMAND COMPLETION

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
LOWER PROCESSOR PRIORITY TO ZERO. ISSUE A RELEASE  
COMMAND WITH INTERRUPT ENABLE SET. MAKE SURE  
INTERRUPT OCCURS. LOWER PRIORITY AFTER INTERRUPT  
AND MAKE SURE INTERRUPT HAS CLEARED.

LOWER PROCESSOR PRIORITY TO ZERO. REISSUE RELEASE  
WITH INTERRUPT ENABLE RESET. MAKE SURE NO INTERRUPT  
OCCURS. SET INTERRUPT ENABLE AND MAKE SURE NO  
INTERRUPT OCCURS.

# MO1

## TEST 46 GO CLEAR OF SILO

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
WRITE ONE WORD INTO THE SILO. ISSUE A RELEASE COMMAND  
WITH INTERRUPT ENABLE RESET. WAIT FOR READY.  
READ THE DATA BUFFER TO MAKE SURE THE SILO HAS BEEN  
CLEARED. (DATA LATE SET AFTER READ OF DATA BUFFER)

## TEST 47 SEEK COMMAND IN DIAGNOSTIC MODE

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET  
24 SECTOR FORMAT TO CYLINDER 1714, HEAD 7, DRIVE 0.  
MAKE SURE NO STATUS BITS ARE SET AND NO ERROR  
BITS ARE SET.

## \*\*ERROR AND STATUS BIT FORCING WITH DRIVE MESSAGES

### TEST 50 DRIVE STATUS FROM SHIFT REGISTER

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT  
RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 757, HEAD 1,  
DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS  
6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE SPEED LOSS,  
DRIVE AVAILABLE, VOLUME VALID, OFFSET, DRIVE READY,  
AND WRITE LOCK ARE SET.

### TEST 51 DRIVE AVAILABLE SETTING

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO A RK06,  
26 SECTOR FORMAT TO CYLINDER 2, HEAD 0, DRIVE 0.  
CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.  
TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE  
AVAILABLE SETS.

### TEST 52 DRIVE BUS PARITY ERROR

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
TO A RK06, 26 SECTOR FORMAT TO CYLINDER 3, HEAD 0,  
DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.  
TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE BUS  
PARITY, DRIVE AVAILABLE, AND CONTROLLER ERROR ARE SET.

# NO1

## TEST 53 DRIVE AVAILABLE RESET ERROR

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
TO A RK06, 26 SECTOR FORMAT, AND DRIVE 0.  
CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.  
TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE  
IS RESET AND CONTROLLER ERROR IS SET.

## TEST 54 CDT SET DRIVE TYPE

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
WITH CDT SET, 26 SECTOR FORMAT, TO CYLINDER 23,  
HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE  
UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE  
AND MAKE SURE ONLY DRIVE AVAILABLE SETS.

## TEST 55 CDT SET AND DRIVE TYPE ERROR

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
WITH CDT SET, 26 SECTOR FORMAT, TO CYLINDER 2,  
HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE  
UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE  
AND MAKE SURE DRIVE AVAILABLE, DRIVE TYPE ERROR,  
AND CONTROLLER ERROR SET.

## TEST 56 RK06 AND DRIVE TYPE ERROR

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
TO A RK06, 26 SECTOR FORMAT, TO CYLINDER 23,  
HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE  
UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC  
MODE AND MAKE SURE DRIVE AVAILABLE, DRIVE TYPE ERROR,  
AND CONTROLLER ERROR SETS.

## TEST 57 SPEED LOSS FROM SHIFT REG.

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO A RK06,  
26 SECTOR FORMAT, TO CYLINDER 3, HEAD 1, DRIVE 0.  
CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN  
OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE AND  
SPEED LOSS ARE SET.

## B02

### TEST 60 DRIVE OFF TRACK FROM SHIFT REG.

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO A RK06,  
26 SECTOR FORMAT, TO CYLINDER 3, HEAD 2, DRIVE 0.  
CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.  
TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE  
AND DRIVE OFF TRACK ARE SET.

### TEST 61 WRITE LOCK ERROR FROM SHIFT REG.

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A PACK ACKNOWLEDGE  
TO A RK06, 26 SECTOR FORMAT, WITH CYLINDER 0,  
HEAD 1, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL  
PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE  
SURE SPEED LOSS, WRITE LOCK ERROR AND CONTROLLER ERROR  
ARE SET WITH DRIVE AVAILABLE RESET.

### TEST 62 SEEK INCOMPLETE

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE AN UNLOAD  
TO A RK06, 26 SECTOR FORMAT, WITH CYLINDER 0,  
HEAD 1, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL  
PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE  
SURE SPEED LOSS, SEEK INCOMPLETE, AND CONTROLLER ERROR  
ARE SET WITH DRIVE AVAILABLE RESET.

### TEST 63 NON-EXECUTABLE DRIVE FUNCTION FROM SHIFT REG.

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR  
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE  
A DRIVE CLEAR TO A RK06, 26 SECTOR FORMAT,  
WITH CYLINDER 0, HEAD 1, DRIVE 0. CLOCK IN DIAGNOSTIC  
MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC  
MODE AND MAKE SURE SPEED LOSS, NON-EXECUTABLE DRIVE FUNCTION, AND  
CONTROLLER ERROR ARE SET WITH DRIVE AVAILABLE RESET.

### TEST 64 AC LOW AND C-D PARITY FROM SHIFT REG.

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611  
CONTROLLER IN DIAGNOSTIC MODE. ISSUE A START SPINDLE  
TO AN RK06, IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0,  
DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.  
TURN OFF DIAGNOSTIC MODE AND MAKE SURE AC LOW, DRIVE

C02

DETECTED SERCOM PARITY, AND CONTROLLER ERROR SET WITH  
DRIVE AVAILABLE RESET.

TEST 65 ILLEGAL DISK ADDRESS ERROR FROM SHIFT REG.

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT  
RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A RECALIBRATE  
TO AN RK06, IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 1,  
DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE  
ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE  
SPEED LOSS, ILLEGAL DISK ADDRESS ERROR, AND CONTROLLER  
ERROR ARE SET WITH DRIVE AVAILABLE RESET.

TEST 66 IDAE DETECTION IN RK611 CONTROLLER (PART 1)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT  
RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A  
SEEK TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 1003,  
HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL  
PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE  
SURE DRIVE AVAILABLE, ILLEGAL DISK ADDRESS ERROR,  
AND CONTROLLER ERROR ARE SET.

TEST 67 IDAE DETECTION IN RK611 CONTROLLER (PART 2)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT  
RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 1022, HEAD  
0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE  
ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE  
DRIVE AVAILABLE AND POSITIONING IN PROGRESS ARE SET  
WITH ILLEGAL DISK ADDRESS ERROR RESET.

TEST 70 IDAE DETECTION IN RK611 CONTROLLER (PART 3)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT  
RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK  
TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 2,  
HEAD 3, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL  
PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE  
SURE DRIVE AVAILABLE, DRIVE OFF TRACK, SPEED LOSS,  
ILLEGAL DISK ADDRESS ERROR, AND CONTROLLER ERROR ARE  
SET.

## TEST 71 IDAE DETECTION IN RK611 CONTROLLER (PART 4)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 23, HEAD 3, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE, DRIVE OFF TRACK, AND SPEED LOSS ARE SET WITH ILLEGAL DISK ADDRESS ERROR RESET.

## TEST 72 IDAE DETECTION IN RK611 CONTROLLER (PART 5)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 22, HEAD 4, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE AND UNSAFE ARE SET WITH ILLEGAL DISK ADDRESS ERROR RESET.

## TEST 73 IDAE DETECTION IN RK611 CONTROLLER (PART 6)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 3, HEAD 4, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE, UNSAFE, ILLEGAL DISK ADDRESS ERROR AND CONTROLLER ERROR ARE SET.

## TEST 74 IDAE DETECTION IN RK611 CONTROLLER (PART 7)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 23, HEAD 5, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE, UNSAFE, SPEED LOSS, ILLEGAL DISK ADDRESS ERROR, AND CONTROLLER ERROR ARE SET.

## TEST 75 IDAE DETECTION IN RK611 CONTROLLER (PART 8)

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 23, HEAD 6, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE, UNSAFE, DRIVE OFF TRACK, ILLEGAL DISK ADDRESS ERROR, AND CONTROLLER CLEAR ARE SET.

# E02

## TEST 76 NON-STANDARD MESSAGE RECEIVING

CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET IN 24 SECTOR FORMAT, CYLINDER 1757, HEAD 7, DRIVE 1. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE NO ERRORS SET AND DRIVE STATUS IS NOT REPORTED. REPEAT FOR DRIVES 2 AND 4.

## TEST 77 DRIVE BUS PARITY ON NON-STANDARD MESSAGE

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 2, HEAD 0, DRIVE 1. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE BUS PARITY ERROR AND CONTROLLER ERROR SETS.

## TEST 100 NON-EXISTENT DRIVE (DRIVE MESSAGE TIME OUT)

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SELECT TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 5. TURN OFF DIAGNOSTIC MODE AND MAKE SURE NON-EXISTENT DRIVE AND CONTROLLER ERROR ARE SET. THIS TEST CHECKS NON-EXISTENT DRIVE DUE TO DRIVE MESSAGE TIME OUT.

## TEST 101 NON-EXISTENT DRIVE AND NO SACK

CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SELECT TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 4. TURN OFF DIAGNOSTIC MODE AND MAKE SURE NON-EXISTENT DRIVE AND CONTROLLER ERROR ARE SET.

THIS TEST EXERCISES THE NON-EXISTENT DRIVE LOGIC DUE TO RELEASE BIT RESET AND SACK RESET BUT THE PASSING

OF THIS TEST DOES GUARENTEE THAT THIS SITUATION DID INDEED CAUSE A NON-EXISTENT DRIVE.

## \*\*ILLEGAL FUNCTION CODE TEST

## TEST 102 ILLEGAL FUNCTION CODE

CLEAR RK611 WITH A CONTROLLER CLEAR. ISSUE AN ILLEGAL COMMAND IN NORMAL MODE AND MAKE SURE COMMAND FINISHES SETTING CONTROLLER READY WITH PROPER ERROR CONDITIONS.

## 6.0 ERROR REPORTING

THE GENERAL FORMAT OF ERROR REPORTS IS:

OPERATION DESCRIPTION AND ERROR DESCRIPTION

TEST NUM	ERROR PC	
XXXXXX	YYYYYY	
EXPECT. REG	ACTUAL REG	OTHER PERTENANT INFORMATION
ZZZZZZ	WWWWW	AAAAAA

NOTE: MOVE 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
1480	TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
1499	EXPECTED RK611 CONTROLLER REGISTERS
1518	PREVIOUS RK611 CONTROLLER REGISTERS
1526	PROGRAM DEFINED VARIABLES
1548	PROGRAM SETUP
1564	INITIALIZE THE COMMON TAGS
1608	TYPE PROGRAM NAME
1615	GET VALUE FOR SOFTWARE SWITCH REGISTER
1685	*DRIVE MESSAGE LOADING
1687	T1 FIRST COMMAND IN MAINT MODE
1753	T2 DRIVE SELECT BITS LOADING FOR DRIVE MESS.
1812	T3 FORMAT BIT LOADING TO FOR DRIVE MESS.
1862	T4 HEAD SELECT BITS LOADING FOR DRIVE MESS.
1928	T5 MESSAGE SELECT BITS LOADING FOR DRIVE MESS.
1998	T6 CLEAR DRIVE COMMAND LOADING FOR DRIVE MESS
2062	T7 UNLOAD COMMAND LOADING FOR DRIVE MESS.
2126	T10 PACK ACKNOWLEDGE COMMAND LOADING FOR DRIVE MESS.
2190	T11 RECALIBRATE COMMAND LOADING FOR DRIVE MESS.
2240	T12 START SPINDLE COMMAND LOADING FOR DRIVE MESS.
2290	T13 SEEK AND CYLINDER ADD 0-777 LOADING FOR DRIVE MESS
2357	T14 SEEK AND CYLINDER BIT 9 AND RK06 FOR DRIVE MESS.
2423	T15 SEEK AND CYLINDER ADD 0,777-1777 LOADING FOR DRIVE MESS
2493	T16 OFFSET COMMAND LOADING FOR DRIVE MESS.
2563	T17 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 1)
2617	T20 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 2)
2678	T21 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 3)
2739	T22 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 4)
2800	T23 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 5)
2861	T24 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 6)
2922	T25 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 1)
2990	T26 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 2)
3058	T27 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 3)
3126	T30 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 4)
3194	T31 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 5)
3262	T32 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 6)
3323	T33 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 7)
3392	*DRIVE MESSAGE LOOPBACK AND PARITY GENERATION TESTS
3394	T34 DRIVE MESSAGE LOOPBACK
3450	T35 DRIVE MESSAGE SHIFT

3507	T36	DRIVE MESSAGE PARITY PRECONDITIONING
3581	T37	ODD DRIVE MESSAGE PARITY GENERATION
3655	T40	DRIVE MESSAGE PARITY INTERACTION
3721	T41	EVEN DRIVE MESSAGE PARITY GENERATION
3797	**CLASS	A COMMAND EXECUTION
3799	T42	RELEASE COMMAND IN DIAGNOSTIC MODE
3855	T43	SELECT COMMAND IN DIAGNOSTIC MODE
3963	T44	RELEASE COMMAND IN NORMAL MODE
4016	T45	INTERRUPT AT COMMAND COMPLETION
4109	T46	GO CLEAR OF SILO
4154	T47	SEEK COMMAND IN DIAGNOSTIC MODE
4240	**ERROR	AND STATUS BIT FORCING WITH DRIVE MESSAGES
4242	T50	DRIVE STATUS FROM SHIFT REGISTER
4329	T51	DRIVE AVAILABLE SETTING
4407	T52	DRIVE BUS PARITY ERROR
4485	T53	DRIVE AVAILABLE RESET ERROR
4561	T54	CDT SET DRIVE TYPE
4639	T55	CDT SET AND DRIVE TYPE ERROR
4718	T56	RK06 AND DRIVE TYPE ERROR
4797	T57	SPEED LOSS FROM SHIFT REG.
4875	T60	DRIVE OFF TRACK FROM SHIFT REG.
4953	T61	WRITE LOCK ERROR FROM SHIFT REG.
5032	T62	SEEK INCOMPLETE
5111	T63	NON-EXECUTABLE DRIVE FUNCTION FROM SHIFT REG.
5190	T64	AC LOW AND C-D PARITY FROM SHIFT REG.
5267	T65	ILLEGAL DISK ADDRESS ERROR FROM SHIFT REG.
5346	T66	IDAE DETECTION IN RK611 CONTROLLER (PART 1)
5431	T67	IDAE DETECTION IN RK611 CONTROLLER (PART 2)
5516	T70	IDAE DETECTION IN RK611 CONTROLLER (PART 3)
5602	T71	IDAE DETECTION IN RK611 CONTROLLER (PART 4)
5687	T72	IDAE DETECTION IN RK611 CONTROLLER (PART 5)
5772	T73	IDAE DETECTION IN RK611 CONTROLLER (PART 6)
5857	T74	IDAE DETECTION IN RK611 CONTROLLER (PART 7)
5942	T75	IDAE DETECTION IN RK611 CONTROLLER (PART 8)
6027	T76	NON-STANDARD MESSAGE RECEIVING
6091	T77	DRIVE BUS PARITY ON NON-STANDARD MESSAGE
6168	T100	NON-EXISTENT DRIVE (DRIVE MESSAGE TIME OUT)
6251	T101	NON-EXISTENT DRIVE AND NO SACK
6331	**ILLEGAL	FUNCTION CODE TEST
6333	T102	ILLEGAL FUNCTION CODE
6380		END OF PASS ROUTINE
6430		CHECK FOR MEMORY CHECK ENABLE OPTION
6449		MEMORY CHECK ENABLE TRAP
6462		SCOPE HANDLER ROUTINE
6605		LOOP ON INTERNAL ERROR
6613		APT COMMUNICATIONS ROUTINE
6670		ERROR HANDLER ROUTINE
6726		TYPE ERROR ROUTINE
6802		TYPE ROUTINE
6881		BINARY TO OCTAL (ASCII) AND TYPE
6958		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7025		TTY INPUT ROUTINE
7192		READ AN OCTAL NUMBER FROM THE TTY
7245		SAVE AND RESTORE RD-R5 ROUTINES
7291		POWER DOWN AND UP ROUTINES
7321		TRAP DECODER

7344	TRAP TABLE
7367	DATA PRINTED BY ERROR ROUTINES
7423	DATA FORMATS
7633	ASCII MESSAGES
7660	DATA HEADERS
7826	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: P2 MD-11-DZR6B-B
.*COPYRIGHT (C) 1976
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY ROY SPITZER
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDECSYSTEMAC
.*PACKAGE (MAINDEC-11-DZGAC-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

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

.\*GENERAL PURPOSE REGISTER DEFINITIONS

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

.\*PRIORITY LEVEL DEFINITIONS

```
000000 PRO= 0      ;;PRIORITY LEVEL 0
000040 PRI= 40    ;;PRIORITY LEVEL 1
```

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

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

:"\*DATA BIT DEFINITIONS (BIT00 TO BIT15)

92	100000	BIT15=	100000
93	040000	BIT14=	40000
94	020000	BIT13=	20000
95	010000	BIT12=	10000
96	004000	BIT11=	4000
97	002000	BIT10=	2000
98	001000	BIT09=	1000
99	000400	BIT08=	400
100	000200	BIT07=	200
101	000100	BIT06=	100
102	000040	BIT05=	40
103	000020	BIT04=	20
104	000010	BIT03=	10
105	000004	BIT02=	4
106	000002	BIT01=	2
107	000001	BIT00=	1
108		.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     ;: VECTOR FOR MEMORY CHECK ENABLE
134      172100  MEMBAS= 172100 ;: BUS ADDRESS FOR MEMORY CHECK ENABLE
135      000001  PAR.EN= 1       ;: MEMORY ENABLE PARITY CHECKING
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	RDDATA= 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	DYE= 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      SPDLSS= 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=   BIT5      ;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= BIT5      ;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
278      000174      000000      .=174
279      000176      000000      DISPREG: .WORD 0      ;SOFTWARE DISPLAY REGISTER
280                      SWREG:  .WORD 0      ;SOFTWARE SWITCH REGISTER
281      .SBTTL STARTING ADDRESS(ES)
    
```

```

281 000200 000137 004316
282 000204 000137 004306
283
284 000214 000137 004276
285
286
287
288
289
290
291 000046 043502
292
293
294 000052 000000
295 000220
296 001000
297
298
299
300
301 001000
302 000024 000200
303 000044 000044
304 000044 001000
305 001000
306
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 @START ;; JUMP TO STARTING ADDRESS OF PROGRAM
JMP RESTRT ;; JUMP TO RESTART ROUTINE
.=214
JMP PARM ;; JUMP TO OPERATOR ASSIGNED PARAMETERS
.SBTTL ACT11 HOOKS

*****
;HOOKS REQUIRED BY ACT11
$SVPC=. ;SAVE PC
.=46
$ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
.=52
.WORD 0 ;;2)SET LOC.52 TO ZERO
.= $SVPC ;; RESTORE PC
.=1000
.SBTTL APT PARAMETER BLOCK

*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
*****
.$X=. ;;SAVE CURRENT LOCATION
.=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200 ;;FOR APT START UP
.=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR ;;POINT TO APT HEADER BLOCK
.=.$X ;;RESET LOCATION COUNTER

*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.

$APTHD:
$HIBTS: .WORD 0 ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;; ADDRESS OF APT MAILBOX (BITS 0-15)
$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 000377  
001210 077  
001211 015  
001212 000012

.WORD 0  
\$TSTNM: .BYTE 00  
SERFLG: .BYTE 00  
\$ICNT: .WORD 00  
\$LPADR: .WORD 00  
\$LPERR: .WORD 00  
\$ERTTL: .WORD 00  
\$ITEMB: .BYTE 0  
\$ERMAX: .BYTE 1  
\$ERRPC: .WORD 0  
\$GDADR: .WORD 0  
\$BDADR: .WORD 0  
\$GDDAT: .WORD 0  
\$BDDAT: .WORD 0  
\$WORD 0  
\$WORD 0  
SAUTOB: .BYTE 0  
SINTAG: .BYTE 0  
\$WORD 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

\*\*\*\*\*  
.SBTTL APT MAILBOX-ETABLE

\*\*\*\*\*  
.EVEN

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	\$DEVN: .WORD	ADEVN	:: 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          .SBTTL  ERROR POINTER TABLE
421
422          ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
423          ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
424          ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
425          ;*NOTE1:      IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
426          ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
427
428          ;*          EM          ;;POINTS TO THE ERROR MESSAGE
429          ;*          DH          ;;POINTS TO THE DATA HEADER
430          ;*          DT          ;;POINTS TO THE DATA
431          ;*          DF          ;;POINTS TO THE DATA FORMAT
432
433
434          001300          $ERRTB:
435          001300          000000          : ERROR 1: ATTEMPTING TO SET CMD BIT DRIVE MESS A
436          001302          000000          : EMIN: 0
437          001304          047666          : DT001
438          001306          050322          : DF001
439          ;
440          ; ERROR 2: ATTEMPTING A SELECT OF DRIVE NUM - CS1 INCORRECT
441          001310          053556          : EM106
442          001312          060307          : EM2003
443          001314          047706          : DT002
444          001316          050356          : DF002
445          ;
446          ; ERROR 3: ATTEMPTING A SELECT OF DRIVE NUM - DRIVE NUM INCORRECT
447          001320          053556          : EM106
448          001322          060352          : EM2004
449          001324          047706          : DT002
450          001326          050356          : DF002
451          ;
452          ; ERROR 4: ATTEMPTING A SELECT OF DRIVE NUM - MESSAGE A INCORRECT
453          001330          053556          : EM106
454          001332          060231          : EM2001
455          001334          047706          : DT002
456          001336          050356          : DF002
457          ;
458          ; ERROR 5: ATTEMPTING A SELECT OF DRIVE NUM - MESSAGE B INCORRECT
459          001340          053556          : EM106
460          001342          060260          : EM2002
461          001344          047706          : DT002
462          001346          050356          : DF002
463          ;
464          ; ERROR 6: ATTEMPTING A SELECT WITH HEAD ADD - CS1 INCORRECT
465          001350          053662          : EM107
466          001352          060307          : EM2003
467          001354          047730          : DT006
468          001356          050412          : DF006
469          ;
470          ; ERROR 7: ATTEMPTING A SELECT WITH HEAD ADD - HEAD INCORRECT
471          001360          053662          : EM107
472          001362          060423          : EM2005
473          001364          047730          : DT006
474          001366          050412          : DF006
475          ;
476          ; ERROR 10: ATTEMPTING A SELECT WITH HEAD ADD - MESSAGE A INCORRECT
477          001370          053662          : EM107
478          001372          060231          : EM2001
479          001374          047730          : DT006
480          001376          050412          : DF006
481          ;
482          ; ERROR 11: ATTEMPTING A SELECT WITH HEAD ADD - MESSAGE B INCORRECT

```

476	001400	053662	EM107
477	001402	060260	EM2002
478	001404	047730	DT006
479	001406	050412	DF006
480			ERROR 12: ATTEMPTING A SELECT WITH MESS SELECT BITS - CSI INCORRECT
481	001410	053757	EM108
482	001412	060307	EM2003
483	001414	047752	DT012
484	001416	050446	DF012
485			ERROR 13: ATTEMPTING A SELECT WITH MESS SELECT BITS - MR1 INCORRECT
486	001420	053757	EM108
487	001422	060470	EM2006
488	001424	047752	DT012
489	001426	050446	DF012
490			ERROR 14: ATTEMPTING A SELECT WITH MESS SELECT BITS - MESS SELECT CODE INCORRECT
491	001430	053757	EM108
492	001432	060517	EM2007
493	001434	047752	DT012
494	001436	050446	DF012
495			ERROR 15: ATTEMPTING A SELECT WITH MESS SELECT BITS - MESS A INCORRECT
496	001440	053757	EM108
497	001442	060231	EM2001
498	001444	047752	DT012
499	001446	050446	DF012
500			ERROR 16: ATTEMPTING A SELECT WITH MESS SELECT BITS - MESS B INCORRECT
501	001450	053757	EM108
502	001452	060260	EM2002
503	001454	047752	DT012
504	001456	050446	DF012
505			ERROR 17: ATTEMPTING A SEEK TO AN RK06 - CSI INCORRECT
506	001460	054057	EM109
507	001462	060307	EM2003
508	001464	050000	DT017
509	001466	050502	DF017
510			ERROR 20: ATTEMPTING A SEEK TO AN RK06 - SEEK BIT IN MESS A NOT SET
511	001470	054057	EM109
512	001472	060153	EM2000
513	001474	050000	DT017
514	001476	050502	DF017
515			ERROR 21: ATTEMPTING A SEEK TO AN RK06 - CYLINDER ADD INCORRECT IN MESS B
516	001500	054057	EM109
517	001502	060567	EM2008
518	001504	050000	DT017
519	001506	050502	DF017
520			ERROR 22: ATTEMPTING A SEEK TO AN RK06 - MESSAGE A INCORRECT
521	001510	054057	EM109
522	001512	060231	EM2001
523	001514	050000	DT017
524	001516	050502	DF017
525			ERROR 23: ATTEMPTING A SEEK TO AN RK06 - MESSAGE B INCORRECT
526	001520	054057	EM109
527	001522	060260	EM2002
528	001524	050000	DT017
529	001526	050000	DT017
530			ERROR 24: ATTEMPTING A SEEK WITH CDT SET - CSI INCORRECT
531	001530	054132	EM110

532	001532	060307	EM2003
533	001534	050000	DT017
534	001536	050502	DF017
535			ERROR 25: ATTEMPTING A SEEK TO AN RKK07 - SEEK BIT IN MESS A NOT SET
536	001540	054132	EM110
537	001542	060153	EM2000
538	001544	050000	DT017
539	001546	050502	DF017
540			ERROR 26: ATTEMPTING A SEEK WITH CDT SET
541			CYLINDER ADD INCORRECT IN MESS B
542	001550	054132	EM110
543	001552	060567	EM2008
544	001554	050000	DT017
545	001556	050502	DF017
546			ERROR 27: ATTEMPTING A SEEK WITH CDT SET - MESSAGE A INCORRECT
547	001560	054132	EM110
548	001562	060231	EM2001
549	001564	050000	DT017
550	001566	050502	DF017
551			ERROR 30: ATTEMPTING A SEEK WITH CDT SET - MESSAGE B INCORRECT
552	001570	054132	EM110
553	001572	060260	EM2002
554	001574	050000	DT017
555	001576	050502	DF017
556			ERROR 31: ATTEMPTING OFFSET - CS1 INCORRECT
557	001600	054207	EM111
558	001602	060307	EM2003
559	001604	050022	DT031
560	001606	050536	DF031
561			ERROR 32: ATTEMPTING OFFSET - OFFSET BITS INCORRECT
562	001610	054207	EM111
563	001612	060640	EM2009
564	001614	050022	DT031
565	001616	050536	DF031
566			ERROR 33: ATTEMPTING OFFSET - MESS A INCORRECT
567	001620	054207	EM111
568	001622	060231	EM2001
569	001624	050022	DT031
570	001626	050536	DF031
571			ERROR 34: ATTEMPTING OFFSET - MESS B INCORRECT
572	001630	054207	EM111
573	001632	060231	EM2001
574	001634	050022	DT031
575	001636	050536	DF031
576			ERROR 35: ATTEMPTING COMMAND WITH NON-ZERO CYL ADD AND OFFSET-
577			CS1 INCORRECT
578	001640	054252	EM112
579	001642	060307	EM2003
580	001644	050044	DT035
581	001646	050572	DF035
582			ERROR 36: ATTEMPTING COMMAND WITH NON-ZERO CYL ADD AND OFFSET-
583			DRIVE COMMAND BIT NOT SET IN MESS A
584	001650	054252	EM112
585	001652	060153	EM2000
586	001654	050044	DT035
587	001656	050572	DF035

# H03

588			:	ERROR 37: ATTEMPTING COMMAND WITH NON-ZERO CYL ADD AND OFFSET-
589			:	CYLINDER ADDRESS BITS INCORRECT IN MESS B
590	001660	054252	:	EM112
591	001662	060567	:	EM2008
592	001664	050044	:	DT035
593	001666	050572	:	DF035
594			:	ERROR 40: ATTEMPTING COMMAND WITH NON-ZERO CYL ADD AND OFFSET-
595			:	MESS A INCORRECT
596	001670	054252	:	EM112
597	001672	060231	:	EM2001
598	001674	050044	:	DT035
599	001676	050572	:	DF035
600			:	ERROR 41: ATTEMPTING COMMAND WITH NON-ZERO CYL ADD AND OFFSET-
601			:	MESS B INCORRECT
602	001700	054252	:	EM112
603	001702	060260	:	EM2002
604	001704	050044	:	DT035
605	001706	050572	:	DF035
606			:	ERROR 42: ATTEMPTING COMMAND WITH NON-ZERO MSG SELECT
607			:	CS1 INCORRECT
608	001710	054405	:	EM113
609	001712	060307	:	EM2003
610	001714	047752	:	DT012
611	001716	050446	:	DF012
612			:	ERROR 43: ATTEMPTING COMMAND WITH NON-ZERO MSG SELECT
613			:	MAINT REG 1 INCORRECT
614	001720	054405	:	EM113
615	001722	060470	:	EM2006
616	001724	047752	:	DT012
617	001726	050446	:	DF012
618			:	ERROR 44: ATTEMPTING COMMAND WITH NON-ZERO MSG SELECT
619			:	DRIVE COMMAND BIT INCORRECT
620	001730	054405	:	EM113
621	001732	060153	:	EM2000
622	001734	047752	:	DT012
623	001736	047752	:	DT012
624			:	ERROR 45: ATTEMPTING COMMAND WITH NON-ZERO MSG SELECT
625			:	MESSAGE SELECT SELECT CODE IN MESSAGE B INCORRECT
626	001740	054405	:	EM113
627	001742	060517	:	EM2007
628	001744	047752	:	DT012
629	001746	050446	:	DF012
630			:	ERROR 46: ATTEMPTING COMMAND WITH NON-ZERO MSG SELECT
631			:	MESS A INCORRECT
632	001750	054405	:	EM113
633	001752	060231	:	EM2001
634	001754	047752	:	DT012
635	001756	050446	:	DF012
636			:	ERROR 47: ATTEMPTING COMMAND WITH NON-ZERO MSG SELECT
637			:	MESS B INCORRECT
638	001760	054405	:	EM113
639	001762	060260	:	EM2002
640	001764	047752	:	DT012
641	001766	050446	:	DF012
642			:	ERROR 50:
643			:	ATTEMPTING TO SHIFT DRIVE MESSAGE
			:	SHIFT REG A INCORRECT

644	001770	054517	EM114	
645	001772	060231	EM2001	
646	001774	050070	DT050	
647	001776	050626	DF050	
648			ERROR 51:	ATTEMPTING TO SHIFT DRIVE MESSAGE
649				SHIFT REG B INCORRECT
650	002000	054517	EM114	
651	002002	060260	EM2002	
652	002004	050070	DT050	
653	002006	050626	DF050	
654			ERROR 52:	ATTEMPTING TO GENERATE ODD PARITY ON SELECT DRIVE MESSAGE
655				PARITY ON MESSAGE A INCORRECT
656	002010	054562	EM115	
657	002012	060711	EM2010	
658	002014	050112	DT052	
659	002016	050662	DF052	
660			ERROR 53:	ATTEMPTING TO GENERATE ODD PARITY ON SELECT DRIVE MESSAGE
661				PARITY ON MESSAGE B INCORRECT
662	002020	054562	EM115	
663	002022	060753	EM2011	
664	002024	050112	DT052	
665	002026	050662	DF052	
666			ERROR 54:	ATTEMPTING TO GENERATE ODD PARITY ON SELECT DRIVE MESSAGE
667				MESSAGE A INCORRECT
668	002030	054562	EM115	
669	002032	060231	EM2001	
670	002034	050112	DT052	
671	002036	050662	DF052	
672			ERROR 55:	ATTEMPTING TO GENERATE ODD PARITY ON SELECT DRIVE MESSAGE
673				MESSAGE B INCORRECT
674	002040	054562	EM115	
675	002042	060260	EM2002	
676	002044	050112	DT052	
677	002046	050662	DF052	
678			ERROR 56:	ATTEMPTING TO GENERATE EVEN PARITY ON SELECT DRIVE MESSAGE
679				PARITY ON MESSAGE A INCORRECT
680	002050	054654	EM116	
681	002052	060711	EM2010	
682	002054	050112	DT052	
683	002056	050662	DF052	
684			ERROR 57:	ATTEMPTING TO GENERATE EVEN PARITY ON SELECT DRIVE MESSAGE
685				PARITY ON MESSAGE IS INCORRECT
686	002060	054654	EM116	
687	002062	060753	EM2011	
688	002064	050112	DT052	
689	002066	050662	DF052	
690			ERROR 60:	ATTEMPTING TO GENERATE EVEN PARITY ON SELECT DRIVE MESSAGE
691				MESSAGE A INCORRECT
692	002070	054654	EM116	
693	002072	060231	EM2001	
694	002074	050112	DT052	
695	002076	050662	DF052	
696			ERROR 61:	ATTEMPTING TO GENERATE EVEN PARITY ON SELECT DRIVE MESSAGE
697				MESSAGE B INCORRECT
698	002100	054654	EM116	
699	002102	060260	EM2002	

700	002104	050112	DT052	
701	002106	050662	DF052	
702			ERROR 62:	ATTEMPTING COMPLETE EXECUTION OF DESELECT DRIVE IN MAINT MODE - COMMAND AND STATUS REG 1 INCORRECT.
703				
704	002110	054747	EM117	
705	002112	060307	EM2003	
706	002114	050126	DT062	
707	002116	050706	DF062	
708			ERROR 63:	ATTEMPTING COMPLETE EXECUTION OF DESELECT DRIVE IN MAINT MODE - COMMAND AND STATUS REG 2 INCORRECT.
709				
710	002120	054747	EM117	
711	002122	061015	EM2012	
712	002124	050126	DT062	
713	002126	050706	DF062	
714			ERROR 64:	ATTEMPTING COMPLETE EXECUTION OF DESELECT DRIVE IN MAINT MODE - ERROR REG. INCORRECT.
715				
716	002130	054747	EM117	
717	002132	061060	EM2013	
718	002134	050126	DT062	
719	002136	050706	DF062	
720			ERROR 65:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN MAINT MODE - COMMAND AND STATUS REGISTER 1 INCORRECT AT PHASE ADDRESS 4
721				
722				
723	002140	055064	EM118	
724	002142	061104	EM2014	
725	002144	050152	DT065	
726	002146	050732	DF065	
727			ERROR 66:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN MAINT MODE - COMMAND AND STATUS REG 1 INVALID DURING COMMAND EXECUTION.
728				
729				
730	002150	055064	EM118	
731	002152	061172	EM2015	
732	002154	050152	DT065	
733	002156	050732	DF065	
734			ERROR 67:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN MAINT MODE - MAINTENANCE REG 2 UNEXPECTEDLY CHANGED DURING COMMAND EXECUTION.
735				
736				
737	002160	055064	EM118	
738	002162	061264	EM2016	
739	002164	050162	DT067	
740	002166	050756	DF067	
741			ERROR 70:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN MAINT MODE - MAINTENANCE REG 3 UNEXPECTEDLY CHANGED DURING COMMAND EXECUTION.
742				
743				
744	002170	055064	EM118	
745	002172	061364	EM2017	
746	002174	050162	DT067	
747	002176	050756	DF067	
748			ERROR 71:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN MAINT MODE - COMMAND AND STATUS REG 1 INCORRECT:
749				
750	002200	055064	EM118	
751	002202	060307	EM2003	
752	002204	050126	DT062	
753	002206	050706	DF062	
754			ERROR 72:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN MAINT MODE - COMMAND AND STATUS REG. 2 INCORRECT.
755				

DZR68B.P11

ERROR POINTER TABLE

756	002210	055064	EM118	
757	002212	061015	EM2012	
758	002214	050126	DT062	
759	002216	050706	DF062	
760			ERROR 73:	ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE IN
761				MAINT MODE - ERROR REGISTER INCORRECT.
762	002220	055064	EM118	
763	002222	061060	EM2013	
764	002224	050126	DT062	
765	002226	050706	DF062	
766			ERROR 74:	ATTEMPTING EXECUTION OF DESELECT DRIVE AT NORMAL
767				SPEED - COMMAND AND STATUS REG. 1 INCORRECT.
768	002230	055177	EM119	
769	002232	060231	EM2001	
770	002234	050126	DT062	
771	002236	050706	DF062	
772			ERROR 75:	ATTEMPTING EXECUTION OF DESELECT DRIVE AT NORMAL
773				SPEED - COMMAND AND STATUS REG. 2 INCORRECT.
774	002240	055177	EM119	
775	002242	061015	EM2012	
776	002244	050126	DT062	
777	002246	050706	DF062	
778			ERROR 76:	ATTEMPTING EXECUTION OF DESELECT DRIVE AT NORMAL
779				SPEED - ERROR REG INCORRECT.
780	002250	055177	EM119	
781	002252	061060	EM2013	
782	002254	050126	DT062	
783	002256	050706	DF062	
784			ERROR 77:	ATTEMPTING TO WRITE CS1 IN MAINT MODE - CS1 INCORRECT
785	002260	055266	EM120	
786	002262	060307	EM2003	
787	002264	050152	DT065	
788	002266	050732	DF065	
789			ERROR 100:	ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE SET
790				INTERRUPT DID NOT OCCUR.
791	002270	055362	EM121	
792	002272	061464	EM2018	
793	002274	050176	DT100	
794	002276	051002	DF100	
795			ERROR 101:	ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE SET
796				CS1 INCORRECT AFTER INTERRUPT.
797	002300	055362	EM121	
798	002302	061514	EM2019	
799	002304	050126	DT062	
800	002306	050706	DF062	
801			ERROR 102:	ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE SET
802				CS2 INCORRECT AFTER INTERRUPT.
803	002310	055362	EM121	
804	002312	061577	EM2020	
805	002314	050126	DT062	
806	002316	050706	DF062	
807			ERROR 103:	ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE SET
808				ERROR REGISTER IN CORRECT AFTER INTERRUPT
809	002320	055362	EM121	
810	002322	061662	EM2021	
811	002324	050126	DT062	

812	002326	050706	DF062		
813			ERROR 104:	ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE SET	
814				INTERRUPT DID NOT CLEAR IN RK611	
815	002330	055362	EM121		
816	002332	061733	EM2022		
817	002334	050176	DT100		
818	002336	051002	DF100		
819			ERROR 105:	ATTEMPTING DESELECT COMMAND AFTER WRITING SILO	
820				TO CHECK GO CLEAR-CS2 INCORRECT	
821					
822	002340	055463	EM122		
823	002342	061015	EM2012		
824	002344	050126	DT062		
825	002346	050706	DF062		
826			ERROR 106:	ATTEMPTING DESELECT COMMAND AFTER WRITING SILO	
827				TO CHECK GO CLEAR-DATA LATE DID NOT OCCUR WHEN	
828				READING SILO	
829	002350	055463	EM122		
830	002352	061774	EM2023		
831	002354	050126	DT062		
832	002356	050706	DF062		
833			ERROR 107:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
834				COMMAND AND STATUS REG 1 INCORRECT AT PHASE ADDRESS 4	
835	002360	055564	EM123		
836	002362	061104	EM2014		
837	002364	050152	DT065		
838	002366	050732	DF065		
839			ERROR 110:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
840				COMMAND AND STATUS REG 1 INVALID DURING COMMAND EXECUTION	
841	002370	055564	EM123		
842	002372	061172	EM2015		
843	002374	050152	DT065		
844	002376	050732	DF065		
845					
846			ERROR 111:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
847				MAINTENANCE REG 2 UNEXPECTEDLY CHANGED DURING COMMAND EXECUTION	
848	002400	055564	EM123		
849	002402	061264	EM2016		
850	002404	050162	DT067		
851	002406	050756	DF067		
852			ERROR 112:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
853				MAINTENANCE REG 3 UNEXPECTEDLY CHANGED DURING COMMAND EXECUTION	
854	002410	055564	EM123		
855	002412	061364	EM2017		
856	002414	050162	DT067		
857	002416	050756	DF067		
858			ERROR 113:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
859				COMMAND AND STATUS REG. 1 INCORRECT	
860	002420	055564	EM123		
861	002422	060307	EM2003		
862	002424	050126	DT062		
863	002426	050706	DF062		
864			ERROR 114:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
865				COMMAND AND STATUS REG. 2 INCORRECT	
866	002430	055564	EM123		
867	002432	061015	EM2012		

868	002434	050126	DT062		
869	002436	050706	DF062		
870			ERROR 115:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE	
871				ERROR REGISTER INCORRECT	
872	002440	055564	EM123		
873	002442	061060	EM2013		
874	002444	050126	DT062		
875	002446	050706	DF062		
876			ERROR 116:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
877				COMMAND AND STATUS REG. 1 INCORRECT	
878	002450	055650	EM124		
879	002452	060307	EM2003		
880	002454	047666	DT001		
881	002456	050322	DF001		
882			ERROR 117:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
883				DRIVE SELECT CODE IN MESSAGE INCORRECT	
884	002460	055650	EM124		
885	002462	060352	EM2004		
886	002464	047666	DT001		
887	002466	050322	DF001		
888			ERROR 120:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
889				DRIVE COMMAND BITS IN MESSAGE INCORRECT	
890	002470	055650	EM124		
891	002472	062046	EM2024		
892	002474	047666	DT001		
893	002476	050322	DF001		
894			ERROR 121:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
895				HEAD ADD CODE IN MESSAGE A INCORRECT	
896	002500	055650	EM124		
897	002502	060423	EM2005		
898	002504	047666	DT001		
899	002506	050322	DF001		
900			ERROR 122:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
901				PARITY BIT IN MESSAGE INCORRECT	
902	002510	055650	EM124		
903	002512	060711	EM2010		
904	002514	047666	DT001		
905	002516	050322	DF001		
906			ERROR 123:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
907				MESS SELECT CODE IN MESSAGE IN CORRECT	
908	002520	055650	EM124		
909	002522	060517	EM2007		
910	002524	047666	DT001		
911	002526	050322	DF001		
912			ERROR 124:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
913				CYLINDER AND BITS IN MESSAGE IS INCORRECT	
914	002530	055650	EM124		
915	002532	060567	EM2008		
916	002534	047666	DT001		
917	002536	050322	DF001		
918			ERROR 125:	ATTEMPTING SELECT DRIVE IN MAINT MODE	
919				PARITY BIT IN MESSAGE IS INCORRECT	
920	002540	055650	EM124		
921	002542	060753	EM2011		
922	002544	047666	DT001		
923	002546	050322	DF001		

924			ERROR 126:	ATTEMPTING COMPLETE EXECUTION OF DESELECT DRIVE IN MAINT MODE - DRIVE STATUS REG INCORRECT
925				
926	002550	054747	EM117	
927	002552	062116	EM2025	
928	002554	050126	DT062	
929	002556	050706	DF062	
930			ERROR 127:	ATTEMPTING EXECUTION OF SELECT DRIVE IN MAINT MODE - DRIVE STATUS REG INCORRECT
931				
932	002560	055064	EM118	
933	002562	062116	EM2025	
934	002564	050126	DT062	
935	002566	050706	DF062	
936			ERROR 130:	ATTEMPTING EXECUTION OF DESELECT DRIVE AT NORMAL SPEED - DRIVE STATUS REG INCORRECT
937				
938	002570	055177	EM119	
939	002572	062116	EM2025	
940	002574	050126	DT062	
941	002576	050706	DF062	
942			ERROR 131:	ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE DRIVE STATUS REG INCORRECT
943				
944	002600	055564	EM123	
945	002602	062116	EM2025	
946	002604	050126	DT062	
947	002606	050706	DF062	
948			ERROR 132:	ATTEMPTING TO FORCE DRA,SPDLSS,VV,OFST,DRDY,WRL CONTROLLER READY DID NOT SET
949				
950	002610	055716	EM125	
951	002612	062156	EM2026	
952	002614	050176	DT100	
953	002616	051002	DF100	
954			ERROR 133:	ATTEMPTING TO FORCE DRA,SPDLSS,VV,OFST,DRDY,WRL LOAD STATUS DID NOT LOAD DRIVE STATUS REF
955				
956	002620	055716	EM125	
957	002622	062213	EM2027	
958	002624	050126	DT062	
959	002626	050706	DF062	
960			ERROR 134:	ATTEMPTING TO FORCE DRA,SPDLSS,VV,OFST,DRDY,WRL CS1 INCORRECT
961				
962	002630	055716	EM125	
963	002632	060307	EM2003	
964	002634	050126	DT062	
965	002636	050706	DF062	
966			ERROR 135:	ATTEMPTING TO FORCE DRA,SPDLSS,VV,OFST,DRDY,WRL CS2 INCORRECT
967				
968	002640	055716	EM125	
969	002642	061015	EM2012	
970	002644	050126	DT062	
971	002646	050706	DF062	
972			ERROR 136:	ATTEMPTING TO FORCE DRA,SPDLSS,VV,OFST,DRDY,WL ERROR REG. INCORRECT
973				
974	002650	055716	EM125	
975	002652	061060	EM2013	
976	002654	050126	DT062	
977	002656	050706	DF062	
978			ERROR 137:	ATTEMPTING TO FORCE DRA,SPDLSS,VV,OFST,DRDY,WL DRIVE STATUS REG. INCORRECT
979				

980	002660	055716	EM125	
981	002662	062116	EM2025	
982	002664	050126	DT062	
983	002666	050706	DF062	
984			ERROR 140:	ATTEMPTING TO FORCE DRIVE AVAILIABLE
985				CS1 INCORRECT
986	002670	056135	EM126	
987	002672	060307	EM2003	
988	002674	050126	DT062	
989	002676	050706	DF062	
990			ERROR 141:	ATTEMPTING TO FORCE DRIVE AVAILIABLE
991				CS2 INCORRECT
992	002700	056135	EM126	
993	002702	061015	EM2012	
994	002704	050126	DT062	
995	002706	050706	DF062	
996			ERROR 142:	ATTEMPTING TO FORCE DRIVE AVAILIABLE
997				DRIVE STATUS REC INCORRECT
998	002710	056135	EM126	
999	002712	062116	EM2025	
1000	002714	050126	DT062	
1001	002716	050706	DF062	
1002			ERROR 143:	ATTEMPTING TO FORCE DRIVE AVAIVIAABLE
1003				ERROR REGISTER INCORRECT
1004	002720	056135	EM126	
1005	002722	061060	EM2013	
1006	002724	050126	DT062	
1007	002726	050706	DF062	
1008			ERROR 144:	ATTEMPTING TO FORCE DRIVE BUS PARITY ERROR DETECTED BY RK611
1009				CS1 INCORRECT
1010	002730	056202	EM127	
1011	002732	060307	EM2003	
1012	002734	050126	DT062	
1013	002736	050706	DF062	
1014			ERROR 145:	ATTEMPTING TO FORCE DRIVE BUS PARITY ERROR DETECTED BY RK611
1015				CS2 INCORRECT
1016	002740	056202	EM127	
1017	002742	061015	EM2012	
1018	002744	050126	DT062	
1019	002746	050706	DF062	
1020			ERROR 146:	ATTEMPTING TO FORCE DRIVE BUS PARITY ERROR DETECTED BY RK611
1021				DRIVE STATUS REG INCORRECT
1022	002750	056202	EM127	
1023	002752	062116	EM2025	
1024	002754	050126	DT062	
1025	002756	050706	DF062	
1026			ERROR 147:	ATTEMPTING TO FORCE DRIVE BUS PARITY ERROR DETECTED BY RK611
1027				ERROR REC INCORRECT
1028	002760	056202	EM127	
1029	002762	061060	EM2013	
1030	002764	050126	DT062	
1031	002766	050706	DF062	
1032			ERROR 150:	ATTEMPTING TO FORCE DRIVE AVAILIABLE RESET ERROR
1033				CS1 INCORRECT
1034	002770	056300	EM128	
1035	002772	060307	EM2003	

1036	002774	050126	DT062	
1037	002776	050706	DF062	
1038			ERROR 151:	ATTEMPTING TO FORCE DRIVE AVAILABLE RESET ERROR
1039				CS2 INCORRECT
1040	003000	056300	EM128	
1041	003002	061015	EM2012	
1042	003004	050126	DT062	
1043	003006	050706	DF062	
1044			ERROR 152:	ATTEMPTING TO FORCE DRIVE AVAILABLE RESET ERROR
1045				DRIVE STATUS REG. INCORRECT
1046	003010	056300	EM128	
1047	003012	062116	EM2025	
1048	003014	050126	DT062	
1049	003016	050706	DF062	
1050			ERROR 153:	ATTEMPTING TO FORCE DRIVE AVAILABLE RESET ERROR
1051				ERROR REG. INCORRECT
1052	003020	056300	EM128	
1053	003022	061060	EM2013	
1054	003024	050126	DT062	
1055	003026	050706	DF062	
1056			ERROR 154:	TESTING CDT SET DRIVE TYPE DETECTION
1057				CS1 INCORRECT
1058	003030	056361	EM129	
1059	003032	060307	EM2003	
1060	003034	050126	DT062	
1061	003036	050706	DF062	
1062			ERROR 155:	TESTING CDT SET DRIVE TYPE DETECTION
1063				CS2 INCORRECT
1064	003040	056361	EM129	
1065	003042	061015	EM2012	
1066	003044	050126	DT062	
1067	003046	050706	DF062	
1068			ERROR 156:	TESTING CDT SET DRIVE TYPE DETECTION
1069				DRIVE STATUS REG INCORRECT
1070	003050	056361	EM129	
1071	003052	062116	EM2025	
1072	003054	050126	DT062	
1073	003056	050706	DF062	
1074			ERROR 157:	TESTING CDT SET DRIVE TYPE DETECTION
1075				ERROR REG INCORRECT
1076	003060	056361	EM129	
1077	003062	061060	EM2013	
1078	003064	050126	DT062	
1079	003066	050706	DF062	
1080			ERROR 160:	ATTEMPTING TO FORCE DRIVE TYPE ERROR ADDRESSING RK06
1081				CS1 INCORRECT
1082	003070	056426	EM130	
1083	003072	060307	EM2003	
1084	003074	050126	DT062	
1085	003076	050706	DF062	
1086			ERROR 161:	ATTEMPTING TO FORCE DRIVE TYPE ERROR ADDRESSING RK06
1087				CS2 INCORRECT
1088	003100	056426	EM130	
1089	003102	061015	EM2012	
1090	003104	050126	DT062	
1091	003106	050706	DF062	

1092			:	ERROR 162:	ATTEMPTING TO FORCE DRIVE TYPE ERROR ADDRESSING RK06
1093			:		DRIVE STATUS REG INCORRECT
1094	003110	056426	:	EM130	
1095	003112	062116	:	EM2025	
1096	003114	050126	:	DT062	
1097	003116	050706	:	DF062	
1098			:	ERROR 163:	ATTEMPTING TO FORCE DRIVE TYPE ERROR ADDRESSING RK06
1099			:		ERROR REG INCORRECT
1100	003120	056426	:	EM130	
1101	003122	061060	:	EM2013	
1102	003124	050126	:	DT062	
1103	003126	050706	:	DF062	
1104			:	ERROR 164:	ATTEMPTING TO FORCE DRIVE TYPE ERROR WITH CTD SET
1105			:		CS1 INCORRECT
1106	003130	056510	:	EM131	
1107	003132	060307	:	EM2003	
1108	003134	050126	:	DT062	
1109	003136	050706	:	DF062	
1110			:	ERROR 165:	ATTEMPTING TO FORCE DRIVE TYPE ERROR WITH CDT SET
1111			:		CS2 INCORRECT
1112	003140	056510	:	EM131	
1113	003142	061015	:	EM2012	
1114	003144	050126	:	DT062	
1115	003146	050706	:	DF062	
1116			:	ERROR 166:	ATTEMPTING TO FORCE DRIVE TYPE ERROR WITH CDT SET
1117			:		DRIVE STATUS REG INCORRECT
1118	003150	056510	:	EM131	
1119	003152	062116	:	EM2025	
1120	003154	050126	:	DT062	
1121	003156	050706	:	DF062	
1122			:	ERROR 167:	ATTEMPTING TO FORCE DRIVE TYPE ERROR WITH CDT SET
1123			:		ERROR REG INCORRECT
1124	003160	056510	:	EM131	
1125	003162	061060	:	EM2013	
1126	003164	050126	:	DT062	
1127	003166	050706	:	DF062	
1128			:	ERROR 170:	ATTEMPTING TO FORCE SPEED LOSS
1129			:		CS1 INCORRECT
1130	003170	056575	:	EM132	
1131	003172	060307	:	EM2003	
1132	003174	050126	:	DT062	
1133	003176	050706	:	DF062	
1134			:	ERROR 171:	ATTEMPTING TO FORCE SPEED LOSS
1135			:		CS2 INCORRECT
1136	003200	056575	:	EM132	
1137	003202	061015	:	EM2012	
1138	003204	050126	:	DT062	
1139	003206	050706	:	DF062	
1140			:	ERROR 172:	ATTEMPTING TO FORCE SPEED LOSS
1141			:		DRIVE STATUS REG INCORRECT
1142	003210	056575	:	EM132	
1143	003212	062116	:	EM2025	
1144	003214	050126	:	DT062	
1145	003216	050706	:	DF062	
1146			:	ERROR 173:	ATTEMPTING TO FORCE SPEED LOSS
1147			:		ERROR REG. INCORRECT

1148	003220	056575	EM132	
1149	003222	061060	EM2013	
1150	003224	050126	DT062	
1151	003226	050706	DF062	
1152			ERROR 174:	ATTEMPTING TO FORCE DRIVE OFF TRACK
1153				CS1 INCORRECT
1154	003230	056634	EM133	
1155	003232	060307	EM2003	
1156	003234	050126	DT062	
1157	003236	050706	DF062	
1158			ERROR 175:	ATTEMPTING TO FORCE DRIVE OFF TRACK
1159				CS2 INCORRECT
1160	003240	056634	EM133	
1161	003242	061015	EM2012	
1162	003244	050126	DT062	
1163	003246	050706	DF062	
1164			ERROR 176:	ATTEMPTING TO FORCE DRIVE OFF TRACK
1165				DRIVE STATUS REG INCORRECT
1166	003250	056634	EM133	
1167	003252	062116	EM2025	
1168	003254	050126	DT062	
1169	003256	050706	DF062	
1170			ERROR 177:	ATTEMPTING TO FORCE DRIVE OFF TRACK
1171				ERROR REG INCORRECT
1172	003260	056634	EM133	
1173	003262	061060	EM2013	
1174	003264	050126	DT062	
1175	003266	050706	DF062	
1176			ERROR 200:	ATTEMPTING TO FORCE WRITE LOCK ERROR
1177				CS1 INCORRECT
1178	003270	056700	EM134	
1179	003272	060307	EM2003	
1180	003274	050126	DT062	
1181	003276	050706	DF062	
1182			ERROR 201:	ATTEMPTING TO FORCE WRITE LOCK ERROR
1183				CS2 INCORRECT
1184	003300	056700	EM134	
1185	003302	061015	EM2012	
1186	003304	050126	DT062	
1187	003306	050706	DF062	
1188			ERROR 202:	ATTEMPTING TO FORCE WRITE LOCK ERROR
1189				DRIVE STATUS REG INCORRECT
1190	003310	056700	EM134	
1191	003312	062116	EM2025	
1192	003314	050126	DT062	
1193	003316	050706	DF062	
1194			ERROR 203:	ATTEMPTING TO FORCE WRITE LOCK ERROR
1195				ERROR REG INCORRECT
1196	003320	056700	EM134	
1197	003322	061060	EM2013	
1198	003324	050126	DT062	
1199	003326	050706	DF062	
1200			ERROR 204:	ATTEMPTING TO FORCE SEEK INCOMPLETE
1201				CS1 INCORRECT
1202	003330	056745	EM135	
1203	003332	060307	EM2003	

1204	003334	050126	DT062	
1205	003336	050706	DF062	
1206			ERROR 205:	ATTEMPTING TO FORCE SEEK INCOMPLETE
1207				CS2 INCORRECT
1208	003340	056745	EM135	
1209	003342	061015	EM2012	
1210	003344	050126	DT062	
1211	003346	050706	DF062	
1212			ERROR 206:	ATTEMPTING TO FORCE SEEK INCOMPLETE
1213				DRIVE STATUS REG INCORRECT
1214	003350	056745	EM135	
1215	003352	062116	EM2025	
1216	003354	050126	DT062	
1217	003356	050706	DF062	
1218			ERROR 207:	ATTEMPTING TO FORCE SEEK INCOMPLETE
1219				ERROR REG INCORRECT
1220	003360	056745	EM135	
1221	003362	061060	EM2013	
1222	003364	050126	DT062	
1223	003366	050706	DF062	
1224			ERROR 210:	ATTEMPTING TO FORCE NON-EXECUTABLE FUNCTION
1225				CS1 INCORRECT
1226	003370	057011	EM136	
1227	003372	060307	EM2003	
1228	003374	050126	DT062	
1229	003376	050706	DF062	
1230			ERROR 211:	ATTEMPTING TO FORCE NON-EXECUTABLE FUNCTION
1231				CS2 INCORRECT
1232	003400	057011	EM136	
1233	003402	061015	EM2012	
1234	003404	050126	DT062	
1235	003406	050706	DF062	
1236			ERROR 212:	ATTEMPTING TO FORCE NON-EXECUTABLE FUNCTION
1237				DRIVE STATUS REG INCORRECT
1238	003410	057011	EM136	
1239	003412	062116	EM2025	
1240	003414	050126	DT062	
1241	003416	050706	DF062	
1242			ERROR 213:	ATTEMPTING TO FORCE NON-EXECUTABLE FUNCTION
1243				ERROR REG INCORRECT
1244	003420	057011	EM136	
1245	003422	061060	EM2013	
1246	003424	050126	DT062	
1247	003426	050706	DF062	
1248			ERROR 214:	ATTEMPTING TO FORCE AC LOW AND C-D PARITY ERROR
1249				CS1 INCORRECT
1250	003430	057065	EM137	
1251	003432	060307	EM2003	
1252	003434	050126	DT062	
1253	003436	050706	DF062	
1254			ERROR 215:	ATTEMPTING TO FORCE AC LOW AND C-D PARITY ERROR
1255				CS2 INCORRECT
1256	003440	057065	EM137	
1257	003442	061015	EM2012	
1258	003444	050126	DT062	
1259	003446	050706	DF062	

1260			:	ERROR 216:	ATTEMPTING TO FORCE AC LOW AND C-D PARITY ERROR
1261			:		DRIVE STATUS REG INCORRECT
1262	003450	057065	:	EM137	
1263	003452	062116	:	EM2025	
1264	003454	050126	:	DT062	
1265	003456	050706	:	DF062	
1266			:	ERROR 217:	ATTEMPTING TO FORCE AC LOW AND C-D PARITY ERROR
1267			:		ERROR REG INCORRECT
1268	003460	057065	:	EM137	
1269	003462	061060	:	EM2013	
1270	003464	050126	:	DT062	
1271	003466	050706	:	DF062	
1272			:	ERROR 220:	ATTEMPTING TO FORCE ILLEGAL DISK ADDRESS ERROR (DRIVE)
1273			:		CS1 INCORRECT
1274	003470	057145	:	EM138	
1275	003472	060307	:	EM2003	
1276	003474	050126	:	DT062	
1277	003476	050706	:	DF062	
1278			:	ERROR 221:	ATTEMPTING TO FORCE ILLEGAL DISK ADDRESS ERROR (DRIVE)
1279			:		CS2 INCORRECT
1280	003500	057145	:	EM138	
1281	003502	061015	:	EM2012	
1282	003504	050126	:	DT062	
1283	003506	050706	:	DF062	
1284			:	ERROR 222:	ATTEMPTING TO FORCE ILLEGAL DISK ADDRESS ERROR (DRIVE)
1285			:		DRIVE STATUS REG INCORRECT
1286	003510	057145	:	EM138	
1287	003512	062116	:	EM2025	
1288	003514	050126	:	DT062	
1289	003516	050706	:	DF062	
1290			:	ERROR 223:	ATTEMPTING TO FORCE ILLEGAL DISK ADDRESS ERROR (DRIVE)
1291			:		ERROR REG INCORRECT
1292	003520	057145	:	EM138	
1293	003522	061060	:	EM2013	
1294	003524	050126	:	DT062	
1295	003526	050706	:	DF062	
1296			:	ERROR 224:	ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR
1297			:		CS1 INCORRECT
1298	003530	057255	:	EM139	
1299	003532	060307	:	EM2003	
1300	003534	050212	:	DT224	
1301	003536	051042	:	DF224	
1302			:	ERROR 225:	ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR
1303			:		CS2 INCORRECT
1304	003540	057255	:	EM139	
1305	003542	061015	:	EM2012	
1306	003544	050212	:	DT224	
1307	003546	051042	:	DF224	
1308			:	ERROR 226:	ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR
1309			:		DRIVE STATUS REG INCORRECT
1310	003550	057255	:	EM139	
1311	003552	062116	:	EM2025	
1312	003554	050212	:	DT224	
1313	003556	051042	:	DF224	
1314			:	ERROR 227:	ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR
1315			:		ERROR REG INCORRECT

77

1316	003560	057255	EM139
1317	003562	061060	EM2013
1318	003564	050212	DT224
1319	003566	051042	DF224
1320			ERROR 230: TESTING ILLEGAL DISK ADDRESS ERROR LOGIC IN RK611
1321			CSI INCORRECT
1322	003570	057337	EM140
1323	003572	060307	EM2003
1324	003574	050246	DT230
1325	003576	051076	DF230
1326			ERROR 231: TESTING ILLEGAL DISK ADDRESS ERROR LOGIC IN RK611
1327			CS2 INCORRECT
1328	003600	057337	EM140
1329	003602	061015	EM2012
1330	003604	050246	DT230
1331	003606	051076	DF230
1332			ERROR 232: TESTING ILLEGAL DISK ADDRESS ERROR LOGIC IN RK611
1333			DRIVE STATUS REG INCORRECT
1334	003610	057337	EM140
1335	003612	062116	EM2025
1336	003614	050246	DT230
1337	003616	051076	DF230
1338			ERROR 233: TESTING ILLEGAL DISK ADDRESS ERROR LOGIC IN RK611
1339			ERROR REGISTER
1340	003620	057337	EM140
1341	003622	061060	EM2013
1342	003624	050246	DT230
1343	003626	051076	DF230
1344			ERROR 234: ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES
1345			CSI INCORRECT
1346	003630	057421	EM141
1347	003632	060307	EM2003
1348	003634	050126	DT062
1349	003636	050706	DF062
1350			ERROR 235: ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES
1351			CS2 INCORRECT
1352	003640	057421	EM141
1353	003642	061015	EM2012
1354	003644	050126	DT062
1355	003646	050706	DF062
1356			ERROR 236: ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES
1357			DRIVE STATUS REG. INCORRECT
1358	003650	057421	EM141
1359	003652	062116	EM2025
1360	003654	050126	DT062
1361	003656	050706	DF062
1362			ERROR 237: ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES
1363			ERROR REG. INCORRECT
1364	003660	057421	EM141
1365	003662	061060	EM2013
1366	003664	050126	DT062
1367	003666	050706	DF062
1368			ERROR 240: ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES WITH
1369			BAD PARITY - CSI INCORRECT
1370	003670	057475	EM142
1371	003672	060307	EM2003

1372	003674	050126	DT062	
1373	003676	050706	DF062	
1374			ERROR 241:	ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES WITH
1375				BAD PARITY - CS2 INCORRECT
1376	003700	057475	EM142	
1377	003702	061015	EM2012	
1378	003704	050126	DT062	
1379	003706	050706	DF062	
1380			ERROR 242:	ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES WITH
1381				BAD PARITY - DRIVE STATUS REG. INCORRECT
1382	003710	057475	EM142	
1383	003712	062116	EM2025	
1384	003714	050126	DT062	
1385	003716	050706	DF062	
1386			ERROR 243:	ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES WITH
1387				BAD PARITY - ERROR ERROR INCOMPLETE
1388	003720	057475	EM142	
1389	003722	061060	EM2013	
1390	003724	050126	DT062	
1391	003726	050706	DF062	
1392			ERROR 244:	ATTEMPTING TO FORCE NON-EXISTENT DRIVE (DRIVE BUS TIMEOUT)
1393				CS1 INCORRECT
1394	003730	057573	EM143	
1395	003732	060307	EM2003	
1396	003734	050126	DT062	
1397	003736	050706	DF062	
1398			ERROR 245:	ATTEMPTING TO FORCE NON-EXISTENT DRIVE (DRIVE BUS TIMEOUT)
1399				CS2 INCORRECT
1400	003740	057573	EM143	
1401	003742	061015	EM2012	
1402	003744	050126	DT062	
1403	003746	050706	DF062	
1404			ERROR 246:	ATTEMPTING TO FORCE NON-EXISTENT DRIVE (DRIVE BUS TIMEOUT)
1405				DRIVE STATUS REG INCORRECT
1406	003750	057573	EM143	
1407	003752	062116	EM2025	
1408	003754	050126	DT062	
1409	003756	050706	DF062	
1410			ERROR 247:	ATTEMPTING TO FORCE NON-EXISTENT DRIVE (DRIVE BUS TIMEOUT)
1411				ERROR REG INCORRECT
1412	003760	057573	EM143	
1413	003762	061060	EM2013	
1414	003764	050126	DT062	
1415	003766	050706	DF062	
1416			ERROR 250:	ATTEMPTING TO FORCE NON-EXISTENT DRIVE (NO SACK)
1417				CS1 INCORRECT
1418	003770	057666	EM144	
1419	003772	060307	EM2003	
1420	003774	050126	DT062	
1421	003776	050706	DF062	
1422			ERROR 251:	ATTEMPTING TO FORCE NON-EXISTENT DRIVE (NO SACK)
1423				CS2 INCORRECT
1424	004000	057666	EM144	
1425	004002	061015	EM2012	
1426	004004	050126	DT062	
1427	004006	050706	DF062	

1428			;	ERROR 252: ATTEMPTING TO FORCE NON-EXISTENT DRIVE (NO SACK)
1429			;	DRIVE STATUS REG INCORRECT
1430	004010	057666		EM144
1431	004012	062116		EM2025
1432	004014	050126		DT062
1433	004016	050706		DF062
1434			;	ERROR 253: ATTEMPTING TO FORCE NON-EXISTENT DRIVE (NO SACK)
1435			;	ERROR REG INCORRECT
1436	004020	057666		EM144
1437	004022	061060		EM2013
1438	004024	050126		DT062
1439	004026	050706		DF062
1440			;	ERROR 254: ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE RESET
1441			;	UNEXPECTED INTERRUPT OCCURRED
1442	004030	057747		EM145
1443	004032	062266		EM2028
1444	004034	050176		DT100
1445	004036	051002		DF100
1446			;	ERROR 255: ATTEMPTING EXECUTION FO DESELECT DRIVE WITH IE RESET
1447			;	INTERRUPT OCCURRED WHEN INTERRUPT ENABLE SET
1448	004040	057747		EM145
1449	004042	062324		EM2029
1450	004044	050176		DT100
1451	004046	051002		DF100
1452			;	ERROR 256: ATTEMPTING TO EXECUTE AN ILLEGAL FUNCTION
1453			;	CSI INCORRECT
1454	004050	060034		EM146
1455	004052	060307		EM2003
1456	004054	050300		DT256
1457	004056	051132		DF256
1458			;	ERROR 257: ATTEMPTING TO EXECUTE AN ILLEGAL FUNCTION
1459			;	ERROR REG INCORRECT
1460	004060	060034		EM146
1461	004062	061060		EM2013
1462	004064	050300		DT256
1463	004066	051132		DF256
1464			;	ERROR 260: ATTEMPTING TO CLEAR ILLEGAL FUNCTION - CSI INCORRECT
1465	004070	060106		EM147
1466	004072	060307		EM2003
1467	004074	050300		DT256
1468	004076	051132		DF256
1469			;	ERROR 261: ATTEMPTING TO CLEAR ILLEGAL FUNCTION - ERROR REG INCORRECT
1470	004100	060106		EM147
1471	004102	061060		EM2013
1472	004104	050300		DT256
1473	004106	051132		DF256
1474			;	ERROR 262: UNEXPECTED MEMORY PARITY ERROR TRAP
1475	004110	053134		EM000
1476	004112	051406		DH000C
1477	004114	047662		DT000
1478	004116	050316		DF000

```

1479          .SBTTL  TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
1480
1481 004120 000000 T.CS1: .WORD 0 ;CONTROL AND STATUS REGISTER 1
1482 004122 000000 T.WC: .WORD 0 ;WORD COUNT REGISTER
1483 004124 000000 T.BA: .WORD 0 ;BUS ADDRESS REGISTER
1484 004126 000000 T.DA: .WORD 0 ;DESIRED TRACK SECTOR REGISTER
1485 004130 000000 T.CS2: .WORD 0 ;CONTROL AND STATUS REGISTER 2
1486 004132 000000 T.DS: .WORD 0 ;DRIVE STATUS REGISTER
1487 004134 000000 T.ER: .WORD 0 ;ERROR REGISTER
1488 004136 000000 T.ASOF: .WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
1489 004140 000000 T.DCYL: .WORD 0 ;DESIRED CYLINDER REGISTER
1490 004142 000000 T.DB: .WORD 0 ;DATA BUFFER
1491 004144 000000 T.MR1: .WORD 0 ;MAINTENANCE REGISTER 1
1492 004146 000000 T.MR2: .WORD 0 ;MAINTENANCE REGISTER 2
1493 004150 000000 T.MR3: .WORD 0 ;MAINTENANCE REGISTER 3
1494 004152 000000 T.ECPS: .WORD 0 ;ECC POSITION INFORMATION
1495 004154 000000 T.ECPT: .WORD 0 ;ECC PATTERN INFORMATION
1496 004156 000000 T.SPARE: .WORD 0 ;SPARE REGISTER
1497
1498          .SBTTL  EXPECTED RK611 CONTROLLER REGISTERS
1499
1500 004160 000000 E.CS1: .WORD 0 ;CONTROL AND STATUS REGISTER 1
1501 004162 000000 E.WC: .WORD 0 ;WORD COUNT REGISTER
1502 004164 000000 E.BA: .WORD 0 ;BUS ADDRESS REGISTER
1503 004166 000000 E.DA: .WORD 0 ;DESIRED TRACK SECTOR REGISTER
1504 004170 000000 E.CS2: .WORD 0 ;CONTROL AND STATUS REGISTER 2
1505 004172 000000 E.DS: .WORD 0 ;DRIVE STATUS REGISTER
1506 004174 000000 E.ER: .WORD 0 ;ERROR REGISTER
1507 004176 000000 E.ASOF: .WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
1508 004200 000000 E.DCYL: .WORD 0 ;DESIRED CYLINDER REGISTER
1509 004202 000000 E.DB: .WORD 0 ;DATA BUFFER
1510 004204 000000 E.MR1: .WORD 0 ;MAINTENANCE REGISTER 1
1511 004206 000000 E.MR2: .WORD 0 ;MAINTENANCE REGISTER 2
1512 004210 000000 E.MR3: .WORD 0 ;MAINTENANCE REGISTER 3
1513 004212 000000 E.ECPS: .WORD 0 ;ECC POSITION INFORMATION
1514 004214 000000 E.ECPT: .WORD 0 ;ECC PATTERN INFORMATION
1515 004216 000000 E.SPARE: .WORD 0 ;SPARE REGISTER
1516
1517          .SBTTL  PREVIOUS RK611 CONTROLLER REGISTERS
1518
1519 004220 000000 P.CS1: .WORD 0 ;PREVIOUS COMMAND AND STATUS REG 1
1520 004222 000000 P.CS2: .WORD 0 ;PREVIOUS COMMAND AND STATUS REG 2
1521 004224 000000 P.DS: .WORD 0 ;PREVIOUS DRIVE STATUS REG
1522 004226 000000 P.ER: .WORD 0 ;PREVIOUS ERROR REG
1523 004230 000000 U.MR2: .WORD 0 ;UNSHIFTED MAINTENANCE REG 2
1524 004232 000000 U.MR3: .WORD 0 ;UNSHIFTED MAINTENANCE REG 3

```

			.SBTTL PROGRAM DEFINED VARIABLES		
1525					
1526					
1527	004234	000210	RKVEC: .WORD	210	:RK611 VECTOR
1528	004236	000240	RKPRI: .WORD	PR5	:RK611 PRIORITY
1529	004240	000000	SRTFLG: .WORD	0	:START FLAG
1530					: 0 = 200
1531					: 1 = 214
1532					: -1 = 204
1533	004242	000000	ERRCNT: .WORD	0	:ERROR COUNT FOR SWITCH 12 ABORT
1534	004244	000000	DRVCOD: .WORD	0	:DRIVE SELECT CODE
1535	004246	000000	MSGCOD: .WORD	0	:MESSAGE SELECT CODE
1536	004250	000000	HDCODE: .WORD	0	:HEAD SELECT CODE
1537	004252	000000	CYLIN: .WORD	0	:CYLINDER ADD VALUE
1538	004254	000000	OFFVAL: .WORD	0	:OFFSET VALUE
1539	004256	000000	SFTCNT: .WORD	0	:SHIFT COUNT FOR DRIVE MESSAGE SHIFTING
1540	004260	000000	PARBIT: .WORD	0	:PARITY BIT FOR SHIFT
1541	004262	000015	WAITIM: .WORD	15	:WAITING FOR DESELECT COMMAND
1542	004264	000144	STALL: .WORD	100.	:STALL TIME FOR MESSAGE TIME OUT (NED)
1543	004266	000000	DRVTYP: .WORD	0	:DRIVE TYPE INDICATOR
1544	004270	000000	ILLFUN: .WORD	0	:ILLEGAL FUNCTION CODE
1545	004272	000000	TRAPPFC: .WORD	0	:ADDRESS OF TRAP FROM MEMORY CHECK
1546	004274	000000	SAVSWR: .WORD	0	:SAVED SWITCH REG FOR POWER FAIL

```

1547          .SBTTL  PROGRAM SETUP
1548
1549 004276 012737 000001 004240 PARM:  MOV   #1,SRTFLG      ;LOAD START FLAG FOR PARMETER START
1550 004304 000406                BR     START1
1551
1552 004306 012737 177777 004240 RESTRT: MOV   #-1,SRTFLG     ;LOAD START FLAG FOR RESTART
1553 004314 000402                BR     START1
1554
1555 004316 005037 004240          START: CLR   SRTFLG      ;CLEAR START FLAG
1556 004322 000005                START1: RESET  ;RESET THE WHOLE SYSTEM
1557 004324 012706 001100          MOV   #STACK,SP  ;INITIALIZE STACK POINTER
1558 004330 012746 000340          MOV   #PR7,-(SP) ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
1559 004334 012746 004342          MOV   #1$,-(SP) ;LOAD START OF PROGRAM
1560 004340 000002                RTI     ;LOAD PSW
1561
1562 004342          1$:
1563          .SETTL  INITIALIZE THE COMMON TAGS
1564          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1565 004342 012706 001100          MOV   #CMTAG,R6  ;;FIRST LOCATION TO BE CLEARED
1566 004346 005026                CLR   (R6)+      ;;CLEAR MEMORY LOCATION
1567 004350 022706 001140          CMP   #SWR,R6   ;;DONE?
1568 004354 001374                BNE   -6         ;;LOOP BACK IF NO
1569 004356 012706 001100          MOV   #STACK,SP ;SETUP THE STACK POINTER
1570          ;;INITIALIZE A FEW VECTORS
1571 004362 012737 043654 000020          MOV   #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1572 004370 012737 000340 000022          MOV   #340,@IOTVEC+2 ;;LEVEL 7
1573 004376 012737 044664 000030          MOV   #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1574 004404 012737 000340 000032          MOV   #340,@EMTVEC+2 ;;LEVEL 7
1575 004412 012737 047572 000034          MOV   #TRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1576 004420 012737 000340 000036          MOV   #340,@TRAPVEC+2;LEVEL 7
1577 004426 012737 047440 000024          MOV   #PWRDN,@PWRVEC ;;POWER FAILURE VECTOR
1578 004434 012737 000340 000026          MOV   #340,@PWRVEC+2 ;;LEVEL 7
1579 004442 013737 043346 043340          MOV   SENDCT,$EOPCT ;SETUP END-OF-PROGRAM COUNTER
1580 004450 005037 001200          CLR   $TIMES    ;INITIALIZE NUMBER OF ITERATIONS
1581 004454 005037 001202          CLR   $ESCAPE   ;CLEAR THE ESCAPE ON ERROR ADDRESS
1582 004460 112737 000001 001115          MOVB  #1,$ERMAX  ;ALLOW ONE ERROR PER TEST
1583 004466 012737 004466 001106          MOV   #,$SLPADR  ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1584 004474 012737 004474 001110          MOV   #,$SLPERR  ;SETUP THE ERROR LOOP ADDRESS
1585          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1586          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1587 004502 013746 000004          MOV   @ERRVEC,-(SP) ;SAVE ERROR VECTOR
1588 004506 012737 004542 000004          MOV   #64$,@ERRVEC ;SET UP ERROR VECTOR
1589 004514 012737 177570 001140          MOV   #DSWR,$SWR ;SETUP FOR A HARDWARE SWICH REGISTER
1590 004522 012737 177570 001142          MOV   #DDISP,$DISPLAY ;AND A HARDWARE DISPLAY REGISTER
1591 004530 022777 177777 174402          CMP   #-1,$SWR  ;TRY TO REFERENCE HARDWARE SWR
1592 004536 001012                BNE   66$       ;BRANCH IF NO TIMEOUT TRAP OCCURRED
1593          ;;AND THE HARDWARE SWR IS NOT = -1
1594 004540 000403                BR    65$       ;BRANCH IF NO TIMEOUT
1595 004542 012716 004550          64$:  MOV   #65$, (SP) ;SET UP FOR TRAP RETURN
1596 004546 000002                RTI
1597 004550 012737 000176 001140          65$:  MOV   #SWREG,$SWR ;POINT TO SOFTWARE SWR
1598 004556 012737 000174 001142          MOV   #DISPREG,$DISPLAY
1599 004564 012637 000004          66$:  MOV   (SP)+,@ERRVEC ;RESTORE ERROR VECTOR
1600
1601 004570 005037 001222          CLR   $PASS     ;CLEAR PASS COUNT
1602 004574 132737 000200 001235          BITB  #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT

```

```

1603 004602 001403      BEQ      67$      ;;YES,USE NON-APT SWITCH
1604 004604 012737 001236 001140      MOV      #$$SWREG,SWR      ;;NO,USE APT SWITCH REGISTER
1605 004612      67$:
1606 004612 005037 004242      CLR      ERRCNT      ;CLEAR ERROR COUNT FOR SWITCH 12 ABORT
1607      .SBTTL      TYPE PROGRAM NAME
1608      ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1609 004616 005227 177777      INC      #-1      ;;FIRST TIME?
1610 004622 001063      BNE      68$      ;;BRANCH IF NO
1611 004624 022737 043502 000042      CMP      #ENDAD,#42      ;;ACT-11?
1612 004632 001457      BEQ      68$      ;;BRANCH IF YES
1613 004634 104401 004702      TYPE      69$      ;;TYPE ASCIZ STRING
1614      .SBTTL      GET VALUE FOR SOFTWARE SWITCH REGISTER
1615 004640 005737 000042      TST      #42      ;;ARE WE RUNNING UNDER XXDP/ACT?
1616 004644 001012      BNE      70$      ;;BRANCH IF YES
1617 004646 123727 001234 000001      CMPB     $ENV,#1      ;;ARE WE RUNNING UNDER APT?
1618 004654 001406      BEQ      70$      ;;BRANCH IF YES
1619 004656 023727 001140 000176      CMP      SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1620 004664 001005      BNE      71$      ;;BRANCH IF NO
1621 004666 104406      GTSWR
1622 004670 000403      BR      71$      ;;GET SOFT-SWR SETTINGS
1623 004672 112737 000001 001134 70$:      MOVB     #1,$AUTOB      ;;SET AUTO-MODE INDICATOR
1624 004700      71$:
1625 004700 000434      BR      68$      ;;GET OVER THE ASCIZ
1626      ;;69$:
1627      .ASCIZ <CRLF>/RK611 DISKLESS DIAGNOSTIC: PART 2 MAINDEC-11-DZR6B-B/<CRLF>
1628 004772 022737 000001 004240 68$:
1629 005000 001122      CMP      #1,SRTFLG      ;CHECK IF PARAMETER START
1630 005002 104401 051156      BNE      15$      ;NO, CONTINUE SETUP
1631 005006 013746 001270 5$:      TYPE     ,OPR001      ;TYPE "RK611 BUS ADDRESS ( ) ="
1632 005012 104402      MOV      $BASE,-(SP)      ;;SAVE $BASE FOR TYPEOUT
1633 005014 104401 051205      TYPOC
1634 005020 104412      TYPE     ,OPR002      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1635 005022 012637 001160      RDOCT
1636 005026 001407      MOV      (SP)+,$TMP0      ;GET VALUE
1637 005030 022737 160000 001160      BEQ      7$      ;CHECK IF <CR>
1638 005036 101361      CMP      #160000,$TMP0      ;CHECK IF IN I/O PAGE
1639 005040 013737 001160 001270      BHI      5$
1640 005046 104401 051213      MOV      $TMP0,$BASE      ;LOAD NEW BUS ADDRESS
1641 005052 013746 001264 7$:      TYPE     ,OPR003      ;TYPE "RK611 VECTOR ADDRESS ( ) ="
1642 005056 042716 160000      MOV      $VECT1,-(SP)
1643 005062 104402      BIC      #160000,(SP)
1644 005064 104401 051205      TYPOC
1645 005070 104412      TYPE     ,OPR002
1646 005072 012637 001160      RDOCT
1647 005076 001412      MOV      (SP)+,$TMP0      ;GET VALUE
1648 005100 022737 001000 001160      BEQ      10$      ;CHECK IF <CR>
1649 005106 101757      CMP      #1000,$TMP0      ;CHECK IF LEGAL
1650 005110 042737 017777 001264      BLOS     7$
1651 005116 053737 001160 001264      BIC      #17777,$VECT1      ;LOAD NEW VECTOR ADDRESS
1652 005124 104401 051243 10$:      BIS      $TMP0,$VECT1
1653 005130 005046      TYPE     ,OPR004      ;TYPE "RK611 PRIORITY ( ) ="
1654 005132 113716 001265      CLR      -(SP)      ;MAKE ROOM ON THE STACK
1655 005136 006216      MOVB     $VECT1+1,(SP)      ;SHIFT 5 BITS RIGHT
1656 005140 006216      ASR      (SP)
1657 005142 006216      ASR      (SP)
1658 005144 006216      ASR      (SP)

```

1659	005146	006216			ASR	(SP)	
1660	005150	104402			TYP0C		
1661	005152	104401	051205		TYPE	.OPR002	
1662	005156	104412			RDOCT		;GET VALUE
1663	005160	012637	001160		MOV	(SP)+,STMPO	
1664	005164	001430			BEQ	15\$	;CHECK FOR DEFAULT
1665	005166	022737	000007	001160	CMP	#7,STMPO	;CHECK IF LEGAL
1666	005174	103753			BLO	10\$	
1667	005176	022737	000004	001160	CMP	#4,STMPO	
1668	005204	101347			BHI	10\$	
1669	005206	006337	001160		ASL	STMPO	;SHIFT 5 BITS LEFT
1670	005212	006337	001160		ASL	STMPO	
1671	005216	006337	001160		ASL	STMPO	
1672	005222	006337	001160		ASL	STMPO	
1673	005226	006337	001160		ASL	STMPO	
1674	005232	042737	160000	001264	BIC	#160000,SVECT1	;STORE NEW PRIORITY
1675	005240	153737	001160	001265	BISB	STMPO,SVECT1+1	
1676	005246	013737	001264	004234	15\$: MOV	SVECT1,RKVEC	;STORE RK611 VECTOR
1677	005254	042737	160000	004234	BIC	#160000,RKVEC	
1678	005262	113737	001265	004236	MOVB	SVECT1+1,RKPRI	;STORE RK611 PRIORITY
1679							
1680	005270	004737	043522		NEWPAS: JSR	PC,CHKPAR	;CHECK FOR MEMORY CHECK ENABLE
1681	005274	012746	000340		MOV	#PR7,-(SP)	;LOCK OUT INTERRUPTS
1682	005300	012746	005306		MOV	#TST1,-(SP)	
1683	005304	000002			RTI		

1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739

.SBTTL \*\*DRIVE MESSAGE LOADING

\*\*\*\*\*  
\*TEST 1 FIRST COMMAND IN MAINT MODE  
\*  
\* INITIALIZE RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN  
\* MODE. ISSUE SELECT DRIVE. WAIT AND MAKE SURE CS1 REMAINS  
\* THE SAME. CLOCK IN MESSAGES A AND B. MAKE SURE  
\* CORRECT MSG ARE LOADED. CHECKING IS DONE A FIELD AT A  
\* TIME.  
\*\*\*\*\*

TST1: SCOPE  
MOV #100,STIMES ;;DO 100. ITERATIONS  
MOV \$BASE,R2 ;;LOAD RK611 BASE  
MOV #CCLR,RKCS1(R2) ;;CLEAR RK611  
MOV #DMD,RKMR1(R2) ;;PUT RK611 IN DIAGNOSTIC MODE  
MOV #SELDRV,RKCS1(R2) ;;LOAD CS1 WITH SELECT DRIVE  
MOV #15,R0 ;;WAIT FOR READY TO SET  
1\$: DEC R0  
BNE 1\$  
MOV RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG. 1  
MOV #SELDRV,E.CS1 ;;LOAD EXPECT CS1  
CMP E.CS1,T.CS1 ;;CHECK IF CS1 CHANGED  
BEQ 2\$ ;;NO CONTINUE  
ERROR 77 ;;CS1 INCORRECT  
BR TST2 ;;GO ON TO NEXT TEST  
2\$: MOV #3\*4+2,R0 ;;CLOCK IN DRIVE MESSAGE  
3\$: MOV #DMD:MCLK,RKMR1(R2)  
MOV #DMD,RKMR1(R2)  
DEC R0  
BNE 3\$  
MOV RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG. 1  
MOV RKMR2(R2),T.MR2 ;;STORE MAINT REG. 2  
MOV RKMR3(R2),T.MR3 ;;STORE MAINT REG. 3  
MOV #SELDRV,E.CS1 ;;LOAD EXPECTED CS1  
CLR E.MR2 ;;LOAD EXPECTED MAINT REG. 2  
CLR E.MR3 ;;LOAD EXPECTED MAINT REG. 3  
CMP E.CS1,T.CS1 ;;CHECK COMMAND AND STATUS REG. 1 CORRECT  
BEQ 4\$ ;;YES, CHECK MESSAGES A & B  
ERROR 116 ;;CS1 INCORRECT  
BR TST2 ;;GO ON TO NEXT TEST  
4\$: BIT #17,T.MR2 ;;CHECK IF DRIVE SELECT BITS ZERO  
BEQ 5\$ ;;YES, CONTINUE  
ERROR 117 ;;MESSAGE SELECT BITS NOT ZERO  
5\$: BIT #7760,T.MR2 ;;CHECK IF COMMAND BITS ZERO  
BEQ 6\$ ;;YES, CONTINUE  
ERROR 120 ;;COMMAND BITS NOT ZERO  
6\$: BIT #70000,T.MR2 ;;CHECK IF HEAD SELECT BITS ZERO  
BEQ 7\$ ;;YES, CONTINUE  
ERROR 121 ;;HEAD SELECT NOT ZERO  
7\$: BIT #BIT15,T.MR2 ;;CHECK PARITY BIT ON MESS A ZERO  
BEQ 8\$ ;;YES, CONTINUE

```

1740 005552 104122          ERROR 122          ;PARITY ON MESS A NOT ZERO
1741 005554 032737 000017 004150 8$: BIT #17,T.MR3      ;CHECK MESS SELECT BITS ZERO
1742 005562 001401          BEQ 9$           ;YES, CONTINUE
1743 005564 104123          ERROR 123          ;MESSAGE SELECT BITS NOT ZERO
1744 005566 032737 077760 004150 9$: BIT #77760,T.MR3  ;CHECK CYLINDER ADDRESS BUFFER
1745 005574 001401          BEQ 10$          ;YES, CONTINUE
1746 005576 104124          ERROR 124          ;CYLINDER ADD BITS NOT ZERO
1747 005600 032737 100000 004150 10$: BIT #BIT15,T.MR3 ;CHECK PARITY BIT ON MESSAGE B
1748 005606 001401          BEQ TST2         ;YES, GO ON TO NEXT TEST
1749 005610 104125          ERROR 125          ;PARITY ON MESS. B NOT ZERO
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762 005612 000004          *TEST 2          DRIVE SELECT BITS LOADING FOR DRIVE MESS.
1763 005614 012737 000144 001200          *
1764 005622 013702 001270          * INITIALIZE RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
1765 005626 005037 004244          * DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 2 WITH
1766 005632 012737 000001 004160          * ZERO. LOAD COMMAND AND STATUS REGISTER WITH A SELECT
1767 005640 012737 005646 001110          * COMMAND. CLOCK IN MESSAGES A AND B INTO SHIFT REGISTER.
1768
1769          * MAKE SURE CORRECT MESSAGES ARE LOADED. REPEAT FOR DRIVE
1770          * SELECT = 1-17.
1771 005646 012762 100000 000000          *****
1772 005654 012762 000040 000026          TST2: SCOPE
1773 005662 013762 004244 000010          MOV #100.,$TIMES ;DO 100. ITERATIONS
1774 005670 012762 000001 000000          MOV $BASE,R2    ;LOAD RK611 BASE
1775 005676 012700 000016          MOV DRVCOD      ;INITIALIZE DRIVE SELECT CODE
1776 005702 012762 000440 000026          CLR #SELDIV,E.CS1 ;LOAD EXPECTED CS1
1777 005710 012762 000040 000026          MOV #1$,SLPERR  ;LOAD LOOP ON ERROR LOCATION FOR
1778 005716 005300          ; SUBTEST LOOP
1779 005720 001370          1$:
1780 005722 016237 000000 004120          MOV #CCLR,RKCS1(R2) ;CLEAR RK611
1781 005730 016237 000034 004146          MOV #DMD,RKMR1(R2) ;PUT RK611 IN DIAGNOSTIC MODE
1782 005736 016237 000036 004150          MOV DRVCOD,RKCS2(R2) ;LOAD DRIVE NUMBER
1783 005744 013737 004244 004206          MOV #SELDIV,RKCS1(R2) ;LOAD SELECT COMMAND
1784 005752 005037 004210          MOV #3*4+2,R0    ;CLOCK IN DRIVE MESSAGE
1785 005756 023737 004160 004120          2$: MOV #DMD!MCLK,RKMR1(R2)
1786 005764 001405          MOV #DMD,RKMR1(R2)
1787 005766 104002          DEC R0
1788 005770 012762 100000 000000          BNE 2$
1789 005776 000426          MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
1790
1791 006000 013737 004146 001160          3$: MOV RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
1792 006006 042737 177760 001160          MOV RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
1793 006014 023737 004244 001160          MOV DRVCOD,E.MR2  ;LOAD EXPECTED MAINT REG. 2
1794 006022 001402          CLR E.MR3        ;LOAD EXPECTED MAINT REG. 3
1795 006024 104003          CMP E.CS1,T.CS1  ;CHECK IF CS1 CORRECT
          BEQ 3$      ;YES, CHECK MESSAGE A&B
          ERROR 2
          MOV #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
          BR 25$     ;CHECK IF LOOP ON ERROR
          MOV T.MR2,$TMPD ;MASK BITS NOT UNDER TEST
          BIC #177760,$TMPD
          CMP DRVCOD,$TMPD ;CHECK IF DRIVE SELECT BITS CORRECT
          BEQ 4$      ;YES, CHECK MESSAGES A&B
          ERROR 3     ;DRIVE SELECT BITS INCORRECT

```

```

1796 006026 000412 BR 25$ ;CHECK IF LOOP ON ERROR
1797
1798 006030 023737 004206 004146 4$: CMP E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
1799 006036 001401 BEQ 5$ ;YES, CHECK MESSAGE B
1800 006040 104004 ERROR 4 ;MESSAGE A INCORRECT
1801 006042 023737 004210 004150 5$: CMP E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
1802 006050 001401 BEQ 25$ ;YES, CHECK IF LOOP ON ERROR
1803 006052 104005 ERROR 5 ;MESSAGE B INCORRECT
1804 006054 104415 25$: SCOP1 ;CHECK IF LOOP ON ERROR
1805 006056 005237 004244 INC DRVCOD ;INCREMENT DRIVE SELECT CODE
1806 006062 022737 000017 004244 CMP #17,DRVCOD ;CHECK IF FINISHED
1807 006070 103266 BHIS 1$ ;NO, TRY NEXT CONFIGURATION
1808
1809

```

\*\*\*\*\*  
\*TEST 3 FORMAT BIT LOADING TO FOR DRIVE MESS.  
\*\*\*\*\*

\* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN  
\* DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH  
\* A SELECT COMMAND AND 24 SECTOR MODE FORMAT. MAKE SURE  
\* CORRECT MESSAGE IS LOADED.  
\*\*\*\*\*

```

1818 006072 000004 TST3: SCOPE
1819 006074 012737 000144 001200 MOV #100,$TIMES ;;DO 100. ITERATIONS
1820 006102 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE
1821 006106 012737 053201 001300 MOV #EM100,EM1N ;LOAD ERROR MESSAGE
1822 006114 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611
1823 006122 012762 000040 000026 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
1824 006130 012762 010001 000000 MOV #CFMT!SELDRV,RKCS1(R2) ;LOAD CFMT!SELDRV INTO COMMAND AND STATUS REG.
1825 006136 012737 010001 004160 MOV #CFMT!SELDRV,E.CS1 ;LOAD EXPECT CS1
1826 006144 012700 000016 MOV #3*4+2,R0 ;CLOCK IN DRIVE MESSAGES
1827 006150 012762 000440 000026 1$: MOV #DMD!MCLK,RKMR1(R2)
1828 006156 012762 000040 000026 MOV #DMD,RKMR1(R2)
1829 006164 005300 DEC R0
1830 006166 001370 BNE 1$
1831 006170 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
1832 006176 016237 000034 004146 MOV RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
1833 006204 016237 000036 004150 MOV RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
1834 006212 012737 001000 004206 MOV #S.FMT,E.MR2 ;LOAD EXPECTED MAINT REG. 2
1835 006220 005037 004210 CLR E.MR3 ;LOAD EXPECTED MAINT REG. 3
1836 006224 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
1837 006232 001410 BEQ 2$ ;YES, CHECK MESSAGE A&B
1838 006234 012737 060307 001302 MOV #EM2003,EM1N+2 ;LOAD ERROR MESSAGE
1839 006242 104001 ERROR 1
1840 006244 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT TEST
1841 006252 000431 BR TST4 ;;GO ON TO NEXT TEST
1842
1843 006254 032737 001000 004146 2$: BIT #S.FMT,T.MR2 ;CHECK IF S.FMT SET IN MESSAGE A
1844 006262 001005 BNE 3$ ;YES, CHECK MESSAGES A&B
1845 006264 012737 060153 001302 MOV #EM2000,EM1N+2 ;LOAD ERROR MESSAGE
1846 006272 104001 ERROR 1
1847 006274 000420 BR TST4 ;;GO ON TO NEXT TEST
1848
1849 006276 023737 004206 004146 3$: CMP E.MR2,T.MR2 ;CHECK IF DRIVE MESSAGE A CORRECT
1850 006304 001404 BEQ 4$ ;YES, CHECK MESSAGE B
1851 006306 012737 060231 001302 MOV #EM2001,EM1N+2 ;LOAD ERROR MESSAGE

```

# F05

```

1852 006314 104001          ERROR 1
1853 006316 023737 004210 004150 4$: CMP E.MR3,T.MR3 ;CHECK IF DRIVE MESSAGE B CORRECT
1854 006324 001404          BEQ TST4 ;:YES, GO ON TO NEXT TEST
1855 006326 012737 060260 001302 MOV #EM2002,EM1N+2 ;LOAD ERROR MESSAGE
1856 006334 104001          ERROR 1
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869 006336 000004          *****
1870 006340 012737 000144 001200 TST4: SCOPE
1871 006346 013702 001270          MOV #100,$TIMES ;:DO 100. ITERATIONS
1872 006352 005037 004250          MOV $BASE,R2 ;LOAD RK611 BASE
1873 006356 012737 000001 004160 CLR HDCODE ;CLEAR HEAD SELECT CODE
1874 006364 012737 006372 001110 MOV #SELDV E.CS1 ;LOAD EXPECTED CS1
1875
1876
1877 006372          1$: MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
1878 006372 012762 100000 000000          SUBTEST LOOP
1879 006400 012762 000040 000026          MOV #CCLR,RKCS1(R2) ;CLEAR RK611
1880 006406 005046          MOV #DMD,RKMR1(R2) ;PUT RK611 IN DIAGNOSTIC MODE
1881 006410 113766 004250 000001          CLR -(SP) ;MAKE ROOM ON STACK
1882 006416 012662 000006          MOVB HDCODE,1(SP) ;LOAD HEAD ADDRESS
1883 006422 012762 000001 000000          MOV (SP)+,RKDA(R2)
1884 006430 012700 000016          MOV #SELDV,RKCS1(R2) ;LOAD SELECT COMMAND
1885 006434 012762 000440 000026 2$: MOV #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
1886 006442 012762 000040 000026          MOV #DMD!MCLK,RKMR1(R2)
1887 006450 005300          DEC R0
1888 006452 001370          BNE 2$
1889 006454 016237 000000 004120          MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
1890 006462 016237 000034 004146          MOV RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
1891 006470 016237 000036 004150          MOV RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
1892 006476 005037 004206          CLR E.MR2
1893 006502 113737 004250 004207          MOVB HDCODE,E.MR2+1 ;GENERATE EXPECTED MAINT REG. 2
1894 006510 006337 004206          ASL E.MR2
1895 006514 006337 004206          ASL E.MR2
1896 006520 006337 004206          ASL E.MR2
1897 006524 006337 004206          ASL E.MR2
1898 006530 005037 004210          CLR E.MR3 ;LOAD EXPECTED MAINT REG. 3
1899 006534 023737 004160 004120          CMP E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
1900 006542 001405          BEQ 3$ ;YES, CHECK MESSAGE A&B
1901 006544 104006          ERROR 6
1902 006546 012762 100000 000000          MOV #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
1903 006554 000426          BR 25$ ;CHECK IF LOOP ON ERROR
1904
1905 006556 013737 004146 001160 3$: MOV T.MR2,$TMP0 ;MASK BITS NOT UNDER TEST
1906 006564 042737 103777 001160          BIC #103777,$TMP0
1907 006572 023737 004206 001160          CMP E.MR2,$TMP0 ;CHECK IF HEAD SELECT BITS CORRECT
    
```

```

1908 006600 001402             BEQ      4$      ;YES, CHECK MESSAGES A&B
1909 006602 104007             ERROR    7       ;HEAD SELECT BITS INCORRECT
1910 006604 000412             BR       25$     ;CHECK IF LOOP ON ERROR
1911
1912 006606 023737 004206 004146 4$:  CMP      E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
1913 006614 001401             BEQ      5$      ;YES, CHECK MESSAGE B
1914 006616 104010             ERROR    10      ;MESSAGE A INCORRECT
1915 006620 023737 004210 004150 5$:  CMP      E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
1916 006626 001401             BEQ      25$     ;YES, CHECK IF LOOP ON ERROR
1917 006630 104011             ERROR    11      ;MESSAGE B INCORRECT
1918 006632 104415             25$:  SCOP1      ;CHECK IF LOOP ON ERROR
1919 006634 005237 004250             INC      HDCODE   ;INCREMENT HEAD SELECT CODE F
1920 006640 022737 000007 004250             CMP      #7,HDCODE ;CHECK IF FINISHED
1921 006646 103251             BHIS     1$      ;NO, TRY NEXT CONFIGURATION
    
```

```

*****
;TEST 5        MESSAGE SELECT BITS LOADING FOR DRIVE MESS.
;
    
```

```

;
;   CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
;   DIAGNOSTIC MODE AND ZERO IN MESSAGE SELECT BITS.  LOAD
;   COMMAND AND STATUS REGISTER 1 WITH A SELECT COMMAND.  CLOCK
;   IN MESSAGE A AND B INTO SHIFT REGISTER.  MAKE SURE
;   CORRECT MESSAGE IS LOADED.  REPEAT FOR MESSAGE SELECT = 1-17.
;
    
```

```

*****
;
    
```

```

1933 006650 000004             TS:  SCOPE
1934 006652 012737 000144 001200             MOV      #100,$TIMES  ;:DO 100. ITERATIONS
1935 006660 013702 001270             MOV      $BASE,R2    ;:LOAD RK611 BASE
1936 006664 005037 004246             CLR      MSGCOD     ;:INITIALIZE MESSAGE SELECT
1937 006670 012737 000001 004160             MOV      #SELDRV,E.CS1 ;:LOAD EXPECTED CSI
1938 006676 012737 006704 001110             MOV      #1$,SLPERR  ;:LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP
1939
1940
1941 006704             1$:  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
1942 006704 012762 100000 000000             MOV      MSGCOD,RKMR1(R2) ;LOAD MESSAGE SELECT BITS
1943 006712 013762 004246 000026             BIS      #DMD,RKMR1(R2) ;PUT RK611 IN DIAGNOSTIC MODE
1944 006720 052762 000040 000026             MOV      #SELDRV,RKCS1(R2) ;LOAD SELECT COMMAND
1945 006726 012762 000001 000000             MOV      #3*4+2,R0   ;CLOCK IF DRIVE MESSAGE
1946 006734 012700 000016             2$:  BIS      #MCLK,RKMR1(R2)
1947 006740 052762 000400 000026             BIC      #MCLK,RKMR1(R2)
1948 006746 042762 000400 000026             DEC      R0
1949 006754 005300             BNE     2$
1950 006756 001370             MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
1951 006760 016237 000000 004120             MOV      RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
1952 006766 016237 000026 004144             MOV      RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG. 2
1953 006774 016237 000034 004146             MOV      RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
1954 007002 016237 000036 004150             MOV      MSGCOD,E.MR1  ;LOAD EXPECTED MAINT REG. 1
1955 007010 013737 004246 004204             BIS      #MWD!DMD,E.MR1
1956 007016 052737 002040 004204             BIT      #ECCW,T.MR1
1957 007024 032737 020000 004144             BEQ      10$
1958 007032 001403             BIS      #ECCW,E.MR1
1959 007034 052737 020000 004204             10$:  CLR      E.MR2
1960 007042 005037 004206             MOV      MSGCOD,E.MR3  ;LOAD EXPECTED MAINT REG. 2
1961 007046 013737 004246 004210             MOV      E.CS1,T.CS1  ;LOAD EXPECTED MAINT REG. 3
1962 007054 023737 004160 004120             CMP      E.CS1,T.CS1  ;CHECK IF CSI CORRECT
1963 007062 001405             BEQ      3$          ;YES, CHECK MAINT REG. 1
    
```

```

1964 007064 104012          ERROR 12          ;CS1 INCORRECT
1965 007066 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
1966 007074 000437          BR       25$        ;CHECK IF LOOP ON ERROR
1967
1968 007076 023737 004204 004144 3$:  CMP      E.MR1,T.MR1  ;CHECK IF MAINT REG. 1 CORRECT
1969 007104 001405          BEQ      4$         ;YES, CHECK MESSAGE A&B
1970 007106 104013          ERROR 13          ;MR1 INCORRECT
1971 007110 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
1972 007116 000426          BR       25$        ;CHECK IF LOOP ON ERROR
1973
1974 007120 013737 004150 001160 4$:  MOV      T.MR3,$TMPD  ;MASK BITS NOT UNDER TEST
1975 007126 042737 177760 001160  BIC      #177760,$TMPD
1976 007134 023737 004246 001160  CMP      MSGCOD,$TMPD ;CHECK IF MESSAGE SELECT CODE CORRECT
1977 007142 001402          BEQ      5$         ;YES, CHECK MESSAGES A&B
1978 007144 104014          ERROR 14          ;MESSAGE SELECT CODE INCORRECT
1979 007146 000412          BR       25$
1980
1981 007150 023737 004206 004146 5$:  CMP      E.MR2,T.MR2  ;CHECK IF MESSAGE A CORRECT
1982 007156 001401          BEQ      6$         ;YES, CHECK MESSAGE B
1983 007160 104015          ERROR 15          ;MESSAGE A INCORRECT
1984 007162 023737 004210 004150 6$:  CMP      E.MR3,T.MR3  ;CHECK IF MESSAGE B CORRECT
1985 007170 001401          BEQ      25$        ;YES, CHECK IF LOOP ON ERROR
1986 007172 104016          ERROR 16          ;MESSAGE B INCORRECT
1987 007174 104415          SCOPE 25$:        ;CHECK IF LOOP ON ERROR
1988 007176 005237 004246          INC      MSGCOD      ;INCREMENT MESSAGE SELECT CODE
1989 007202 022737 000017 004246  CMP      #17,MSGCOD  ;CHECK IF FINISHED
1990 007210 103235          BHIS   1$          ;NO, TRY NEXT CONFIGURATION
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002 007212 000004          *TEST 6 CLEAR DRIVE COMMAND LOADING FOR DRIVE MESS
2003 007214 012737 000144 001200  *
2004 007222 013702 001270          * CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
2005 007226 012737 053267 001300  * DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH
2006 007234 012737 000005 004160  * A DRIVE CLEAR. CLOCK MESSAGE A AND B INTO SHIFT REGISTERS.
2007 007242 012737 000400 004206  * MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY. REPEAT
2008 007250 012737 007256 001110  * FOR 24 SECTOR FORMAT.
2009
2010
2011 007256          *
2012 007256 012762 100000 000000 1$:  MOV      #100,$TIMES ;DO 100. ITERATIONS
2013 007264 012762 000040 000026  MOV      $BASE,R2    ;LOAD RK611 BASE
2014 007272 013762 004160 000000  MOV      #EM101,EMIN ;LOAD ERROR MESSAGE
2015 007300 012700 000016  MOV      #CLEAR,E.CS1 ;LOAD EXPECTED COMMAND AND STATUS REG. 1
2016 007304 012762 000440 000026 2$:  MOV      #S.CLR,E.MR2 ;LOAD EXPECTED MAINT. REG. 2
2017 007312 012762 000040 000026  MOV      #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
2018 007320 005300          ; SUBTEST LOOP
2019 007322 001370          ;

```

```

2020 007324 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2021 007332 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG. 2
2022 007340 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
2023 007346 005037 004210      CLR      E.MR3           ;STORE EXPECTED MAINT REG. 3
2024 007352 023737 004160 004120      CMP      E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
2025 007360 001410      BEQ      3$              ;YES, CHECK MESSAGE A&B
2026 007362 012737 060307 001302      MOV      #EM2003,EM1N+2 ;LOAD ERROR MESSAGE
2027 007370 104001      ERROR   1
2028 007372 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
2029 007400 000437      BR       25$            ;CHECK IF LOOP ON ERROR
2030
2031 007402 013737 004146 001160 3$:      MOV      T.MR2,$TMPD     ;MASK BITS NOT UNDER TEST
2032 007410 042737 176377 001160      BIC      #C<S.FMT!S.CLR>,$TMPD
2033 007416 023737 004206 001160      CMP      E.MR2,$TMPD     ;CHECK IF S.CLR AND FORMAT
2034                                ;BITS IN MESSAGE CORRECT
2035 007424 001405      BEQ      4$              ;YES, CHECK MESSAGE A&B
2036 007426 012737 060153 001302      MOV      #EM2000,EM1N+2 ;LOAD ERROR MESSAGE
2037 007434 104001      ERROR   1
2038 007436 000420      BR       25$            ;CHECK IF LOOP ON ERROR
2039
2040 007440 023737 004206 004146 4$:      CMP      E.MR2,T.MR2     ;CHECK IF DRIVE MESSAGE A CORRECT
2041 007446 001404      BEQ      5$              ;YES, CHECK MESSAGE B
2042 007450 012737 060231 001302      MOV      #EM2001,EM1N+2 ;LOAD ERROR MESSAGE
2043 007456 104001      ERROR   1
2044 007460 023737 004210 004150 5$:      CMP      E.MR3,T.MR3     ;CHECK IF DRIVE MESSAGE B CORRECT
2045 007466 001404      BEQ      25$            ;YES, CHECK IF LOOP ON EROR
2046 007470 012737 060260 001302      MOV      #EM2002,EM1N+2 ;LOAD ERROR MESSAGE
2047 007476 104001      ERROR   1
2048 007500 104415      SCOPI   1
2049 007502 032737 010000 004160 25$:      ;CHECK IF LOOP ON ERROR
2050 007510 001007      BIT      #CFMT,E.CS1     ;CHECK IF ISSUED IN 24 SECTOR FORMAT
2051 007512 052737 010000 004160      BNE      TST7            ;YES, GO ON TO NEXT TEST
2052 007520 052737 001000 004206      BIS      #CFMT,E.CS1     ;INDICATE COMMAND IN 24 SECTOR FORMAT
2053 007526 000653      BIS      #S.FMT,E.MR2   ;REISSUE IN 24 SECTOR FORMAT
2054      BR       1$

```

```

*****
:TEST 7          UNLOAD COMMAND LOADING FOR DRIVE MESS.
:
:      CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
:      DIAGNOSTIC MODE.  LOAD COMMAND AND STATUS REGISTER 1 WITH
:      AN UNLOAD COMMAND.  CLOCK IN MESSAGES A AND B INTO SHIFT
:      REGISTERS.  MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY.
:      REPEAT FOR 24 SECTOR FORMAT.
:
*****

```

```

2064
2065 007530 000004      TST7:  SCOPE
2066 007532 012737 000144 001200      MOV      #100,$TIMES    ;DO 100. ITERATIONS
2067 007540 013702 001270      MOV      $BASE,R2       ;LOAD RK611 BASE
2068 007544 012737 053336 001300      MOV      #EM102,EM1N    ;LOAD ERROR MESSAGE
2069 007552 012737 000007 004160      MOV      #UNLOAD,E.CS1  ;LOAD EXPECTED COMMAND AND STATUS REG. 1
2070 007560 012737 002000 004206      MOV      #S.UNLD,E.MR2  ;LOAD EXPECTED MAINT. REG. 2
2071 007566 012737 007574 001110      MOV      #1$,SLPERR     ;LOAD LOOP ON ERROR LOCATION FOR
2072                                ;SUBTEST LOOP
2073
2074 007574 012762 100000 000000 1$:      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
2075

```

```

2076 007602 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
2077 007610 013762 004160 000000      MOV      E.CS1,RKCS1(R2) ;LOAD UNLOAD INTO COMMAND AND STATUS REG. 1
2078 007616 012700 000016          MOV      #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
2079 007622 012762 000440 000026 2$:      MOV      #DMD!MCLK,RKMR1(R2)
2080 007630 012762 000040 000026      MOV      #DMD,RKMR1(R2)
2081 007636 005300          DEC      R0
2082 007640 001370          BNE     2$
2083 007642 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2084 007650 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG. 2
2085 007656 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
2086 007664 005037 004210          CLR     E.MR3 ;STORE EXPECTED MAINT REG. 3
2087 007670 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
2088 007676 001410          BEQ     3$ ;YES, CHECK MESSAGE A&B
2089 007700 012737 060307 001302      MOV      #EM2003,EMIN+2 ;LOAD ERROR MESSAGE
2090 007706 104001          ERROR  1
2091 007710 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
2092 007716 000437          BR      25$ ;CHECK IF LOOP ON ERROR
2093
2094 007720 013737 004146 001160 3$:      MOV      T.MR2,$TMPO ;MASK BITS NOT UNDER TEST
2095 007726 042737 174777 001160      BIC     #↑C<S.FMT!S.UNLD>,$TMPO
2096 007734 023737 004206 001160      CMP     E.MR2,$TMPO ;CHECK IF S.UNLD AND FORMAT
2097                                     ;BITS IN MESSAGE CORRECT
2098 007742 001405          BEQ     4$ ;YES, CHECK MESSAGE A&B
2099 007744 012737 060153 001302      MOV      #EM2000,EMIN+2 ;LOAD ERROR MESSAGE
2100 007752 104001          ERROR  1
2101 007754 000420          BR      25$ ;CHECK IF LOOP ON ERROR
2102
2103 007756 023737 004206 004146 4$:      CMP     E.MR2,T.MR2 ;CHECK IF DRIVE MESSAGE A CORRECT
2104 007764 001404          BEQ     5$ ;YES, CHECK MESSAGE B
2105 007766 012737 060231 001302      MOV      #EM2001,EMIN+2 ;LOAD ERROR MESSAGE
2106 007774 104001          ERROR  1
2107 007776 023737 004210 004150 5$:      CMP     E.MR3,T.MR3 ;CHECK IF DRIVE MESSAGE B CORRECT
2108 010004 001404          BEQ     25$ ;YES, CHECK IF LOOP ON EROR
2109 010006 012737 060260 001302      MOV      #EM2002,EMIN+2 ;LOAD ERROR MESSAGE
2110 010014 104001          ERROR  1
2111 010016 104415          SCOP1  ;CHECK IF LOOP ON ERROR
2112 010020 032737 010000 004160 25$:      BIT     #CFMT,E.CS1 ;CHECK IF ISSUED IN 24 SECTOR FORMAT
2113 010026 001007          BNE     TST10 ;YES, GO ON TO NEXT TEST
2114 010030 052737 010000 004160      BIS     #CFMT,E.CS1 ;INDICATE COMMAND IN 24 SECTOR FORMAT
2115 010036 052737 001000 004206      BIS     #S.FMT,E.MR2
2116 010044 000653          BR      1$ ;REISSUE IN 24 SECTOR FORMAT
2117
2118                                     ;*****
2119 *TEST 10      PACK ACKNOWLEDGE COMMAND LOADING FOR DRIVE MESS.
2120 *
2121 *      CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
2122 *      DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH
2123 *      A.PACK ACKNOWLEDGE. CLOCK MESSAGES A AND B INTO SHIFT
2124 *      REGISTERS. MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY.
2125 *      REPEAT FOR 24 SECTOR FORMAT.
2126 *
2127                                     ;*****
2128 010046 000004      TST10: SCOP2
2129 010050 012737 000144 001200      MOV     #100,$TIMES ;DO 100. ITERATIONS
2130 010056 013702 001270          MOV     $BASE,R2 ;LOAD RK611 BASE
2131 010062 012737 053400 001300      MOV     #EM103,EMIN ;LOAD ERROR MESSAGE

```

# K05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P2 MD-11-DZR6B-B MACY11 27(732) 01-OCT-76 10:23 PAGE 42  
DZR6BB.P11 T10 PACK ACKNOWLEDGE COMMAND LOADING FOR DRIVE MESS.

SEQ 0044

2132	010070	012737	000003	004160	MOV	#PACK,E.CS1	;LOAD EXPECTED COMMAND AND STATUS REG. 1	
2133	010076	012737	004000	004206	MOV	#S.PACK,E.MR2	;LOAD EXPECTED MAINT. REG. 2	
2134	010104	012737	010112	001110	MOV	#1\$,SLPERR	;LOAD LOOP ON ERROR LOCATION FOR	
2135							; SUBTEST LOOP	
2136								
2137	010112				1\$:			
2138	010112	012762	100000	000000	MOV	#CLR,RKCS1(R2)	;CLEAR RK611	
2139	010120	012762	000040	000026	MOV	#DMD,RKMR1(R2)	;PUT RK611 IN MAINTENANCE MODE	
2140	010126	013762	004160	000000	MOV	E.CS1,RKCS1(R2)	;LOAD PACK INTO COMMAND AND STATUS REG. 1	
2141	010134	012700	000016		MOV	#3*4+2,RO	;CLOCK IN DRIVE MESSAGE	
2142	010140	012762	000440	000026	2\$:	MOV	#DMD!MCLK,RKMR1(R2)	
2143	010146	012762	000040	000026	MOV	#DMD,RKMR1(R2)		
2144	010154	005300			DEC	RO		
2145	010156	001370			BNE	2\$		
2146	010160	016237	000000	004120	MOV	RKCS1(R2),T.CS1	;STORE COMMAND AND STATUS REG. 1	
2147	010166	016237	000034	004146	MOV	RKMR2(R2),T.MR2	;STORE MAINTENANCE REG. 2	
2148	010174	016237	000036	004150	MOV	RKMR3(R2),T.MR3	;STORE MAINTENANCE REG. 3	
2149	010202	005037	004210		CLR	E.MR3	;STORE EXPECTED MAINT REG. 3	
2150	010206	023737	004160	004120	CMP	E.CS1,T.CS1	;CHECK IF CS1 CORRECT	
2151	010214	001410			BEQ	3\$	;YES, CHECK MESSAGE A&B	
2152	010216	012737	060307	001302	MOV	#EM2003,EM1N+2	;LOAD ERROR MESSAGE	
2153	010224	104001			ERROR	1		
2154	010226	012762	100000	000000	MOV	#CLR,RKCS1(R2)	;CLEAN UP FOR NEXT CONFIGURATION	
2155	010234	000437			BR	25\$	;CHECK IF LOOP ON ERROR	
2156								
2157	010236	013737	004146	001160	3\$:	MOV	T.MR2,\$TMPO	;MASK BITS NOT UNDER TEST
2158	010244	042737	172777	001160	BIC	#T<S.FMT!S.PACK>,\$TMPO		
2159	010252	023737	004206	001160	CMP	E.MR2,\$TMPO	;CHECK IF S.PACK AND FORMAT	
2160							; BITS IN MESSAGE CORRECT	
2161	010260	001405			BEQ	4\$	;YES, CHECK MESSAGE A&B	
2162	010262	012737	060153	001302	MOV	#EM2000,EM1N+2	;LOAD ERROR MESSAGE	
2163	010270	104001			ERROR	1		
2164	010272	000420			BR	25\$	;CHECK IF LOOP ON ERROR	
2165								
2166	010274	023737	004206	004146	4\$:	CMP	E.MR2,T.MR2	;CHECK IF DRIVE MESSAGE A CORRECT
2167	010302	001404			BEQ	5\$	;YES, CHECK MESSAGE B	
2168	010304	012737	060231	001302	MOV	#EM2001,EM1N+2	;LOAD ERROR MESSAGE	
2169	010312	104001			ERROR	1		
2170	010314	023737	004210	004150	5\$:	CMP	E.MR3,T.MR3	;CHECK IF DRIVE MESSAGE B CORRECT
2171	010322	001404			BEQ	25\$	;YES, CHECK IF LOOP ON EROR	
2172	010324	012737	060260	001302	MOV	#EM2002,EM1N+2	;LOAD ERROR MESSAGE	
2173	010332	104001			ERROR	1		
2174	010334	104415			25\$:	SCOP1	;CHECK IF LOOP ON ERROR	
2175	010336	032737	010000	004160	BIT	#CFMT,E.CS1	;CHECK IF ISSUED IN 24 SECTOR FORMAT	
2176	010344	001007			BNE	TST11	;YES, GO ON TO NEXT TEST	
2177	010346	052737	010000	004160	BIS	#CFMT,E.CS1	;INDICATE COMMAND IN 24 SECTOR FORMAT	
2178	010354	052737	001000	004206	BIS	#S.FMT,E.MR2		
2179	010362	000653			BR	1\$	;REISSUE IN 24 SECTOR FORMAT	

2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187

```
*****  
*TEST 11 RECALIBRATE COMMAND LOADING FOR DRIVE MESS.  
*  
* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN  
* DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH  
* A RECALIBRATE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS.  
* MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY.  
*****
```

L05

```

2188
2189
2190 010364 000004
2191 010366 012737 000144 001200
2192 010374 013702 001270
2193 010400 012737 053454 001300
2194 010406 012762 100000 000000
2195 010414 012762 000040 000026
2196 010422 012762 000013 000000
2197 010430 012737 000013 004160
2198 010436 012700 000016
2199 010442 012762 000440 000026 1$:
2200 010450 012762 000040 000026
2201 010456 005300
2202 010460 001370
2203 010462 016237 000000 004120
2204 010470 016237 000034 004146
2205 010476 016237 000036 004150
2206 010504 012737 000040 004206
2207 010512 005037 004210
2208 010516 023737 004160 004120
2209 010524 001410
2210 010526 012737 060307 001302
2211 010534 104001
2212 010536 012762 100000 000000
2213 010544 000431
2214
2215 010546 032737 000040 004146 2$:
2216 010554 001005
2217 010556 012737 060153 001302
2218 010564 104001
2219 010566 000420
2220
2221 010570 023737 004206 004146 3$:
2222 010576 001404
2223 010600 012737 060231 001302
2224 010606 104001
2225 010610 023737 004210 004150 4$:
2226 010616 001404
2227 010620 012737 060260 001302
2228 010626 104001
2229

```

\*\*\*\*\*  
TST11: SCOPE

```

MOV #100.,$TIMES ;:DO 100. ITERATIONS
MOV $BASE,R2 ;:LOAD RK611 BASE
MOV #EM104,EMIN ;:LOAD ERROR MESSAGE
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611
MOV #DMD,RKMR1(R2) ;:PUT RK611 IN MAINTENANCE MODE
MOV #RECAL,RKCS1(R2) ;:LOAD RECAL INTO COMMAND AND STATUS REG. 1
MOV #RECAL,E.CS1 ;:LOAD EXPECT CS1
MOV #3*4+2,R0 ;:CLOCK IN DRIVE MESSAGES
MOV #DMD!MCLK,RKMR1(R2)
MOV #DMD,RKMR1(R2)
DEC R0
BNE 1$
MOV RKCS1(R2),T.CS1 ;:STORE COMMAND AND STATUS REG. 1
MOV RKMR2(R2),T.MR2 ;:STORE MAINT REG. 2
MOV RKMR3(R2),T.MR3 ;:STORE MAINT REG. 3
MOV #S.RECL,E.MR2 ;:LOAD EXPECTED MAINT REG. 2
CLR E.MR3 ;:LOAD EXPECTED MAINT REG. 3
CMP E.CS1,T.CS1 ;:CHECK IF CS1 CORRECT
BEQ 2$ ;:YES, CHECK MESSAGE A&B
MOV #EM2003,EMIN+2 ;:LOAD ERROR MESSAGE
ERROR 1
MOV #CCLR,RKCS1(R2) ;:CLEAN UP FOR NEXT TEST
BR TST12 ;:GO ON TO NEXT TEST

```

```

2$: BIT #S.RECL,T.MR2 ;:CHECK IF S.RECL SET IN MESSAGE A
BNE 3$ ;:YES, CHECK MESSAGES A&B
MOV #EM2000,EMIN+2 ;:LOAD ERROR MESSAGE
ERROR 1
BR TST12 ;:GO ON TO NEXT TEST

```

```

3$: CMP E.MR2,T.MR2 ;:CHECK IF DRIVE MESSAGE A CORRECT
BEQ 4$ ;:YES, CHECK MESSAGE B
MOV #EM2001,EMIN+2 ;:LOAD ERROR MESSAGE
ERROR 1
4$: CMP E.MR3,T.MR3 ;:CHECK IF DRIVE MESSAGE B CORRECT
BEQ TST12 ;:YES, GO ON TO NEXT TEST
MOV #EM2002,EMIN+2 ;:LOAD ERROR MESSAGE
ERROR 1

```

```

*****
:TEST 12 START SPINDLE COMMAND LOADING FOR DRIVE MESS.
:
: CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
: DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH
: A START SPINDLE. CLOCK MESSAGES A AND B INTO SHIFT REGISTERS.
: MAKE SURE SHIFT REGISTERS ARE LOADED CORRECTLY.
:
*****

```

```

2238
2239 010630 000004
2240 010632 012737 000144 001200
2241 010640 013702 001270
2242 010644 012737 053523 001300
2243 010652 012762 100000 000000

```

```

*****
TST12: SCOPE
MOV #100.,$TIMES ;:DO 100. ITERATIONS
MOV $BASE,R2 ;:LOAD RK611 BASE
MOV #EM105,EMIN ;:LOAD ERROR MESSAGE
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611

```

```

2244 010660 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
2245 010666 012762 000011 000000      MOV      #SRTSPL,RKCS1(R2) ;LOAD SRTSPL INTO COMMAND AND STATUS REG. 1
2246 010674 012737 000011 004160      MOV      #SRTSPL,E.CS1 ;LOAD EXPECT CS1
2247 010702 012700 000016 000000      MOV      #3*4+2,R0 ;CLOCK IN DRIVE MESSAGES
2248 010706 012762 000440 000026 1$:      MOV      #DMD!MCLK,RKMR1(R2)
2249 010714 012762 000040 000026      MOV      #DMD,RKMR1(R2)
2250 010722 005300      DEC      R0
2251 010724 001370      BNE     1$
2252 010726 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2253 010734 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
2254 010742 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
2255 010750 012737 000100 004206      MOV      #S.STSP,E.MR2 ;LOAD EXPECTED MAINT REG. 2
2256 010756 005037 004210 004206      CLR      E.MR3 ;LOAD EXPECTED MAINT REG. 3
2257 010762 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
2258 010770 001410      BEQ     2$ ;YES, CHECK MESSAGE A&B
2259 010772 012737 060307 001302      MOV      #EM2003,EM1N+2 ;LOAD ERROR MESSAGE
2260 011000 104001      ERROR   1
2261 011002 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT TEST
2262 011010 000431      BR      TST13 ;GO ON TO NEXT TEST
2263
2264 011012 032737 000100 004146 2$:      BIT      #S.STSP,T.MR2 ;CHECK IF S.STSP SET IN MESSAGE A
2265 011020 001005      BNE     3$ ;YES, CHECK MESSAGES A&B
2266 011022 012737 060153 001302      MOV      #EM2000,EM1N+2 ;LOAD ERROR MESSAGE
2267 011030 104001      ERROR   1
2268 011032 000420      BR      TST13 ;GO ON TO NEXT TEST
2269
2270 011034 023737 004206 004146 3$:      CMP      E.MR2,T.MR2 ;CHECK IF DRIVE MESSAGE A CORRECT
2271 011042 001404      BEQ     4$ ;YES, CHECK MESSAGE B
2272 011044 012737 060231 001302      MOV      #EM2001,EM1N+2 ;LOAD ERROR MESSAGE
2273 011052 104001      ERROR   1
2274 011054 023737 004210 004150 4$:      CMP      E.MR3,T.MR3 ;CHECK IF DRIVE MESSAGE B CORRECT
2275 011062 001404      BEQ     TST13 ;YES, GO ON TO NEXT TEST
2276 011064 012737 060260 001302      MOV      #EM2002,EM1N+2 ;LOAD ERROR MESSAGE
2277 011072 104001      ERROR   1
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289 011074 000004      TST13:  SCOPE
2290 011076 012737 000144 001200      MOV      #100,$TIMES ;DO 100. ITERATIONS
2291 011104 013702 001270 001200      MOV      $BASE,R2 ;LOAD RK611 BASE
2292 011110 005037 004252 001200      CLR      CYLIN ;INITIALIZE CYLINDER
2293 011114 012737 000017 004160      MOV      #SEEK,E.CS1 ;LOAD EXPECTED CS1
2294 011122 012737 011130 001110      MOV      #1$,SLPERR ;LOAD LOOP ON ERROR LOCATION FOR
2295
2296
2297 011130
2298 011130 012762 100000 000000 1$:      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
2299 011136 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE

```

```

*****
*TEST 13      SEEK AND CYLINDER ADD 0-777 LOADING FOR DRIVE MESS
*
*      CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
*      DIAGNOSTIC MODE.  LOAD ZERO IN CYLINDER ADDRESS.  LOAD
*      COMMAND AND STATUS REGISTER 1 WITH A SEEK COMMAND.
*      CLOCK IN MESSAGE A AND B INTO SHIFT REGISTER.  MAKE SURE
*      CORRECT MESSAGE IS LOADED.  REPEAT FOR CYLINDER = 1-777.
*
*****

```

# N05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P2 MD-11-DZR6B-B MACY11 27(732) 01-OCT-76 10:23 PAGE 45  
 DZR6BB.P11 T13 SEEK AND CYLINDER ADD 0-777 LOADING FOR DRIVE MESS

SEQ 0047

2300	011144	013762	004252	000020		MOV	CYLIN,RKDCYL(R2)	:LOAD CYLINDER ADDRESS
2301	011152	012762	000017	000000		MOV	#SEEK,RKCS1(R2)	:ISSUE SEEK
2302	011160	012700	000016			MOV	#3*4+2,RO	:CLOCK IN DRIVE MESSAGE
2303	011164	012762	000440	000026	2\$:	MOV	#DMD!MCLK,RKMR1(R2)	
2304	011172	012762	000040	000026		MOV	#DMD,RKMR1(R2)	
2305	011200	005300				DEC	RO	
2306	011202	001370				BNE	2\$	
2307	011204	016237	000000	004120		MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG. 1
2308	011212	016237	000034	004146		MOV	RKMR2(R2),T.MR2	:STORE MAINT REG. 2
2309	011220	016237	000036	004150		MOV	RKMR3(R2),T.MR3	:STORE MAINT REG. 3
2310	011226	012737	000020	004206		MOV	#S.SEEK,E.MR2	:LOAD EXPECTED MAINT REG. 2
2311	011234	013737	004252	004210		MOV	CYLIN,E.MR3	:GENERATE EXPECTED MAINT REG. 3
2312	011242	006337	004210			ASL	E.MR3	
2313	011246	006337	004210			ASL	E.MR3	
2314	011252	006337	004210			ASL	E.MR3	
2315	011256	006337	004210			ASL	E.MR3	
2316	011262	023737	004160	004120		CMP	E.CS1,T.CS1	:CHECK IF CS1 CORRECT
2317	011270	001405				BEQ	3\$	:YES, CHECK MESSAGE A&B
2318	011272	104017				ERROR	17	
2319	011274	012762	100000	000000		MOV	#CCLR,RKCS1(R2)	:CLEAN UP FOR NEXT CONFIGURATION
2320	011302	000434				BR	25\$	:CHECK IF LOOP ON ERROR
2321								
2322	011304	032737	000020	004146	3\$:	BIT	#S.SEEK,T.MR2	:CHECK IF SEEK COMMAND BIT SET
2323	011312	001002				BNE	4\$	:YES, CHECK CYLINDER ADDRESS BITS
2324	011314	104020				ERROR	20	:SEEK BIT NOT SET
2325	011316	000426				BR	25\$	:CHECK IF LOOP ON ERROR
2326								
2327	011320	013737	004150	001160	4\$:	MOV	T.MR3,\$TMPD	:MASK BITS NOT UNDER TEST
2328	011326	042737	140017	001160		BIC	#140017,\$TMPD	
2329	011334	023737	004210	001160		CMP	E.MR3,\$TMPD	:CHECK IF CYLINDER ADDRESS BITS CORRECT
2330	011342	001402				BEQ	5\$	:YES, CHECK MESSAGES A&B
2331	011344	104021				ERROR	21	:CYLINDER ADDRESS BITS INCORRECT
2332	011346	000412				BR	25\$	:CHECK IF LOOP ON ERROR
2333								
2334	011350	023737	004206	004146	5\$:	CMP	E.MR2,T.MR2	:CHECK IF MESSAGE A CORRECT
2335	011356	001401				BEQ	6\$	:YES, CHECK MESSAGE B
2336	011360	104022				ERROR	22	:MESSAGE A INCORRECT
2337	011362	023737	004210	004150	6\$:	CMP	E.MR3,T.MR3	:CHECK IF MESSAGE B CORRECT
2338	011370	001401				BEQ	25\$	:YES, CHECK IF LOOP ON ERROR
2339	011372	104023				ERROR	23	:MESSAGE B INCORRECT
2340	011374	104415			25\$:	SCOP1		:CHECK IF LOOP ON ERROR
2341	011376	005237	004252			INC	CYLIN	:INCREMENT CYLINDER NUMBER
2342	011402	022737	000777	004252		CMP	#777,CYLIN	:CHECK IF FINISHED
2343	011410	103247				BHIS	1\$	:NO, TRY NEXT CONFIGURATION

```

*****
*TEST 14      SEEK AND CYLINDER BIT 9 AND RK06 FOR DRIVE MESS.
*
* CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
* DIAGNOSTIC MODE.  LOAD 1000 IN CYLINDER ADDRESS.  LOAD
* COMMAND AND STATUS REGISTER 1 WITH A SEEK COMMAND.
* CLOCK IN MESSAGE A AND B INTO SHIFT REGISTERS.  MAKE
* SURE CYLINDER BIT 9 IN MESSAGE IN RESET.  REPEAT FOR
* CYLINDER = 1400.
*****

```

76

```

2356 011412 000004          TST14: SCOPE
2357 011414 012737 000144 001200      MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
2358 011422 013702 001270          MOV      $BASE,R2        ;;LOAD RK611 BASE
2359 011426 012737 001000 004252      MOV      #1000,CYLIN     ;;INITIALIZE CYLINDER
2360 011434 005037 004210          CLR      E.MR3           ;;LOAD EXPECTED
2361 011440 012737 000017 004160      MOV      #SEEK,E.CS1     ;;LOAD EXPECTED CS1
2362 011446 012737 011454 001110      MOV      #1$, $LPERR     ;;LOAD LOOP ON ERROR LOCATION FOR
2363                                     ;; SUBTEST LOOP
2364
2365 011454          1$:
2366 011454 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
2367 011462 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
2368 011470 013762 004252 000020      MOV      CYLIN,RKDCYL(R2) ;LOAD CYLINDER ADDRESS
2369 011476 012762 000017 000000      MOV      #SEEK,RKCS1(R2) ;ISSUE SEEK
2370 011504 012700 000016          MOV      #3*4+2,R0       ;;CLOCK IN DRIVE MESSAGE
2371 011510 012762 000440 000026      MOV      #DMD!MCLK,RKMR1(R2)
2372 011516 012762 000040 000026      MOV      #DMD,RKMR1(R2)
2373 011524 005300          DEC      R0
2374 011526 001370          BNE     2$
2375 011530 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2376 011536 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINT REG.2
2377 011544 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINT REG.3
2378 011552 012737 000020 004206      MOV      #S.SEEK,E.MR2   ;;LOAD EXPECTED MAINT REG. 2
2379 011560 023737 004160 004120      CMP      E.CS1,T.CS1    ;;CHECK IF CS1 CORRECT
2380 011566 001405          BEQ     3$               ;;YES, CHECK MESSAGE A&B
2381 011570 104017          ERROR  17
2382 011572 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
2383 011600 000434          BR     25$              ;CHECK IF LOOP ON ERROR
2384
2385 011602 032737 000020 004146      3$:  BIT      #S.SEEK,T.MR2 ;CHECK IF SEEK COMMAND BIT SEEK
2386 011610 001002          BNE     4$               ;;YES, CHECK CYLINDER ADDRESS BITS
2387 011612 104020          ERROR  20               ;;SEEK BIT NOT SET
2388 011614 000426          BR     25$              ;CHECK IF LOOP ON ERROR
2389
2390 011616 013737 004150 001160      4$:  MOV      T.MR3,$TMP0    ;;MASK BITS NOT UNDER TEST
2391 011624 042737 140017 001160      BIC     #140017,$TMP0
2392 011632 023737 004210 001160      CMP      E.MR3,$TMP0    ;;CHECK IF CYLINDER ADDRESS BITS CORRECT
2393 011640 001402          BEQ     5$               ;;YES, CHECK MESSAGES A&B
2394 011642 104021          ERROR  21               ;;CYLINDER ADDRESS BITS INCORRECT
2395 011644 000412          BR     25$              ;CHECK IF LOOP ON ERROR
2396
2397 011646 023737 004206 004146      5$:  CMP      E.MR2,T.MR2    ;;CHECK IF MESSAGE A CORRECT
2398 011654 001401          BEQ     6$               ;;YES, CHECK MESSAGE B
2399 011656 104022          ERROR  22               ;;MESSAGE B INCORRECT
2400 011660 023737 004210 004150      6$:  CMP      E.MR3,T.MR3    ;;CHECK IF MESSAGE IS CORRECT
2401 011666 001401          BEQ     25$              ;;YES, CHECK IF LOOP ON ERROR
2402 011670 104023          ERROR  23               ;;MESSAGE INCORRECT
2403 011672 104415          25$: SCOPE1            ;CHECK IF LOOP ON ERROR
2404 011674 022737 001400 004252      CMP      #1400,CYLIN    ;CHECK IF CYLINDER 1400
2405 011702 001007          BNE     TST15           ;;YES, GO ON TO NEXT TEST
2406 011704 012737 001400 004252      MOV      #1400,CYLIN    ;SET CYLINDER=1400
2407 011712 012737 010000 004210      MOV      #10000,E.MR3   ;;LOAD EXPECTED CONFIGUR
2408 011720 000655          BR     1$               ;TRY NEXT CONFIGURATION
2409
2410                                     ;*****
2411                                     ;*TEST 15      SEEK AND CYLINDER ADD 0,777-1777 LOADING FOR DRIVE MESS

```

```

2412
2413
2414
2415
2416
2417
2418
2419
2420 011722 000004
2421 011724 012737 000144 001200
2422 011732 013702 001270
2423 011736 005037 004252
2424 011742 012737 002017 004160
2425 011750 012737 011756 001110
2426
2427
2428 011756
2429 011756 012762 100000 000000
2430 011764 012762 000040 000026
2431 011772 013762 004252 000020
2432 012000 012762 002017 000000
2433 012006 012700 000016
2434 012012 012762 000440 000026
2435 012020 012762 000040 000026
2436 012026 005300
2437 012030 001370
2438 012032 016237 000000 004120
2439 012040 016237 000034 004146
2440 012046 016237 000036 004150
2441 012054 012737 000020 004206
2442 012062 013737 004252 004210
2443 012070 006337 004210
2444 012074 006337 004210
2445 012100 006337 004210
2446 012104 006337 004210
2447 012110 023737 004160 004120
2448 012116 001405
2449 012120 104024
2450 012122 012762 100000 000000
2451 012130 000434
2452
2453 012132 032737 000020 004146
2454 012140 001002
2455 012142 104025
2456 012144 000426
2457
2458 012146 013737 004150 001160
2459 012154 042737 140017 001160
2460 012162 023737 004210 001160
2461 012170 001402
2462 012172 104026
2463 012174 000412
2464
2465 012176 023737 004206 004146
2466 012204 001401
2467 012206 104027

```

```

*****
* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
* DIAGNOSTIC MODE. LOAD 0 IN CYLINDER ADDRESS. LOAD
* COMMAND AND STATUS REGISTER 1 WITH SEEK COMMAND AND
* CDT SET. CLOCK IN MESSAGE A AND B INTO SHIFT REGISTER.
* MAKE SURE CYLINDER CORRECT. REPEAT FOR CYLINDER = 777-1777.
*****
T15: SCOPE
MOV #100.,$TIMES ;;DO 100. ITERATIONS
MOV $BASE,R2 ;;LOAD RK611 BASE
CLR CYLIN ;;INITIALIZE CYLINDER
MOV #CDT!SEEK,E.CS1 ;;LOAD EXPECTED CS1
MOV #15,$LPERA ;;LOAD LOOP ON ERROR LOCATION FOR
;; SUBTEST LOOP

15:
MOV #CLR,RKCS1(R2) ;CLEAR RK611
MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
MOV CYLIN,RKDCYL(R2) ;LOAD CYLINDER ADDRESS
MOV #CDT!SEEK,RKCS1(R2) ;ISSUE SEEK WITH CDT SET
MOV #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
25:
MOV #DMD!MCLK,RKMR1(R2)
MOV #DMD,RKMR1(R2)
DEC R0
BNE 25
MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
MOV RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
MOV RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
MOV #5,SEEK,E.MR2 ;LOAD EXPECTED MAINT REG. 2
MOV CYLIN,E.MR3 ;GENERATE EXPECTED MAINT REG. 3
ASL E.MR3
ASL E.MR3
ASL E.MR3
ASL E.MR3
CMP E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
BEQ 35 ;YES, CHECK MESSAGE A&B
ERROR 24
MOV #CLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
BR 255 ;CHECK IF LOOP ON ERROR

35:
BIT #5,SEEK,T.MR2 ;CHECK IF SEEK COMMAND BIT SET
BNE 45 ;YES, CHECK CYLINDER ADDRESS BITS
ERROR 25 ;SEEK BIT NOT SEEK
BR 255 ;CHECK IF LOOP ON ERROR

45:
MOV T.MR3,$TMP0 ;MASK BITS NOT UNDER TEST
BIC #140017,$TMP0
CMP E.MR3,$TMP0 ;CHECK IF CYLINDER ADDRESS BITS CORRECT
BEQ 55 ;YES, CHECK MESSAGES A&B
ERROR 26 ;CYLINDER ADDRESS BIT INCORRECT
BR 255 ;CHECK IF LOOP ON ERROR

55:
CMP E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
BEQ 65 ;YES, CHECK M MESSAGE B
ERROR 27

```

2468	012210	023737	004210	004150	6\$:	CMP	E.MR3,T.MR3	:CHECK IF MESSAGE B
2469	012216	001401				BEG	25\$	:YES, CHECK IF LOOP ON ERROR
2470	012220	104030				ERROR	30	:MESSAGE B INCORRECT
2471	012222	104415			25\$:	SCOPI		:CHECK IF LOOP ON ERROR
2472	012224	005737	004252			TST	CYLIN	:CHECK IF ZERO
2473	012230	001003				BNE	26\$	:NO, INCREMENT CYLINDER
2474	012232	012737	000776	004252		MOV	#776,CYLIN	:NEXT CYLINDER=777
2475	012240	005237	004252		26\$:	INC	CYLIN	:INCREMENT CYLINDER NUMBER
2476	012244	022737	001777	004252		CMP	#1777,CYLIN	:CHECK IF FINISHED
2477	012252	103241				BHIS	1\$	:NO, TRY NEXT CONFIGURATION

\*\*\*\*\*  
:TEST 16           OFFSET COMMAND LOADING FOR DRIVE MESS.  
\*\*\*\*\*

:  
:           CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN  
:           DIAGNOSTIC MODE. LOAD OFFSET REGISTER WITH 0. LOAD  
:           COMMAND AND STATUS REGISTER 1 WITH AN OFFSET. CLOCK  
:           MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE SHIFT  
:           REGISTERS ARE LOADED CORRECTLY. REPEAT FOR OFFSET  
:           REGISTER = 1-377.  
\*\*\*\*\*

2490	012254	000004			†ST16:	SCOPE		
2491	012256	012737	000144	001200		MOV	#100,\$TIMES	:DO 100. ITERATIONS
2492	012264	013702	001270			MOV	\$BASE,R2	:LOAD RK611 BASE
2493	012270	005037	004254			CLR	OFFVAL	:INITIALIZE OFFSET VALUE
2494	012274	012737	000015	004160		MOV	#OFFSET,E.CS1	:LOAD EXPECTED CS1
2495	012302	005037	004206			CLR	E.MR2	:LOAD EXPECT MAINT REG 2
2496	012306	012737	012314	001110		MOV	#1\$, \$LPERR	:LOAD LOOP ON ERROR LOCATION FOR :           SUBTEST LOOP
2497								
2498								
2499	012314				1\$:			
2500	012314	012762	100000	000000		MOV	#CLR,RKCS1(R2)	:CLEAR RK611
2501	012322	012762	000040	000026		MOV	#DMD,RKMR1(R2)	:PUT RK611 IN MAINT MODE
2502	012330	013762	004254	000016		MOV	OFFVAL,RKASOF(R2)	:LOAD OFFSET VALUE
2503	012336	012762	000015	000000		MOV	#OFFSET,RKCS1(R2)	:ISSUE OFFSET
2504	012344	012700	000016			MOV	#3*4+2,R0	:CLOCK IN DRIVE MESSAGE
2505	012350	012762	000440	000026	2\$:	MOV	#DMD!MCLK,RKMR1(R2)	
2506	012356	012762	000040	000026		MOV	#DMD,RKMR1(R2)	
2507	012364	005300				DEC	R0	
2508	012366	001370				BNE	2\$	
2509	012370	016237	000000	004120		MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG. 1
2510	012376	016237	000034	004146		MOV	RKMR2(R2),T.MR2	:STORE MAINT REG. 2
2511	012404	016237	000036	004150		MOV	RKMR3(R2),T.MR3	:STORE MAINT REG. 3
2512	012412	005037	004210			CLR	E.MR3	:LOAD EXPECTED MAINT REG. 2
2513	012416	013737	004254	004210		MOV	OFFVAL,E.MR3	:GENERATE EXPECTED MR3
2514	012424	005137	004210			COM	E.MR3	
2515	012430	042737	177700	004210		BIC	#177700,E.MR3	
2516	012436	006337	004210			ASL	E.MR3	
2517	012442	006337	004210			ASL	E.MR3	
2518	012446	006337	004210			ASL	E.MR3	
2519	012452	006337	004210			ASL	E.MR3	
2520	012456	052737	014000	004210		BIS	#14000,E.MR3	
2521	012464	032737	000200	004254		BIT	#BIT7,OFFVAL	:DETERMINE SIGN
2522	012472	001003				BNE	10\$	
2523	012474	052737	002000	004210		BIS	#BIT10,E.MR3	

```

2524 012502 023737 004160 004120 10$: CMP E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
2525 012510 001405 BEQ 4$ ;YES, CHECK MESSAGE A&B
2526 012512 104031 ERROR 31
2527 012514 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAN UP FOR NEXT CONFIGURATION
2528 012522 000426 BR 25$ ;CHECK IF LOOP ON ERROR
2529
2530 012524 013737 004150 001160 4$: MOV T.MR3,$TMP0 ;MASK BITS NOT UNDER TEST
2531 012532 042737 140017 001160 BIC #140017,$TMP0
2532 012540 023737 004210 001160 CMP E.MR3,$TMP0 ;CHECK IF OFFSET VALUE CORRECT
2533 012546 001402 BEQ 5$ ;YES, CHECK MESSAGES A&B
2534 012550 104032 ERROR 32 ;OFFSET VALUE INCORRECT
2535 012552 000412 BR 25$ ;CHECK IF LOOP ON ERROR
2536
2537 012554 023737 004206 004146 5$: CMP E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
2538 012562 001401 BEQ 6$ ;YES, CHECK MESSAGE B
2539 012564 104033 ERROR 33 ;MESSAGE A INCORRECT
2540 012566 023737 004210 004150 6$: CMP E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
2541 012574 001401 BEQ 25$ ;YES, CHECK IF LOOP ON ERROR
2542 012576 104034 ERROR 34 ;MESSAGE B INCORRECT
2543 012600 104415 25$: SCOP1 ;CHECK IF LOOP ON ERROR
2544 012602 005237 004254 INC OFFVAL ;INCREMENT OFFSET VALUE
2545 012606 022737 000377 004254 CMP #377,OFFVAL ;CHECK IF FINISHED
2546 012614 103237 BHIS 1$ ;NO, TRY NEXT CONFIGURATION
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559

```

```

*****
*TEST 17 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 1)
*

```

```

* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
* DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER
* WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND
* AND STATUS REGISTER 1 WITH A SELECT. CLOCK
* MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE
* SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER
* ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.
*

```

```

*****
*ST17: SCOPE
2560 012616 000004 MOV #100,$TIMES ;DO 100. ITERATIONS
2561 012620 012737 MOV $BASE,R2 ;LOAD RK611 BASE
2562 012626 013702 MOV #1777,CYLIN ;LOAD CYLINDER VALUE
2563 012632 012737 MOV #52,OFFVAL ;LOAD OFFSET VALUE
2564 012640 012737 MOV #SELDRV,E.CS1 ;LOAD EXPECTED CS1
2565 012646 012737 MOV #CCLR,RKCS1(R2) ;CLEAR RK611
2566 012654 012762 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
2567 012662 012762 MOV #1777,RKDCYL(R2) ;LOAD CYLINDER VALUE
2568 012670 012762 MOV #52,RKASOF(R2) ;LOAD OFFSET VALUE
2569 012676 012762 MOV #SELDRV,RKCS1(R2) ;ISSUE SELDRV
2570 012704 012762 MOV #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
2571 012712 012700 MOV #DMD!MCLK,RKMR1(R2)
2572 012716 012762 1$: MOV #DMD,RKMR1(R2)
2573 012724 012762 MOV R0
2574 012732 005300 BNE 1$
2575 012734 001370 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2576 012736 016237 000000 004120 MOV RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
2577 012744 016237 000034 004146 MOV RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
2578 012752 016237 000036 004150 MOV #0,E.MR2 ;LOAD EXPECTED MAINT REG. 2
2579 012760 012737 000000 004206

```

2580	012766	005037	004210		CLR	E.MR3	:LOAD EXPECTED MAINTENANCE REG. 3
2581	012772	023737	004160	004120	CMP	E.CS1,T.CS1	:CHECK IF CS1 CORRECT
2582	013000	001405			BEQ	2\$	:YES, CHECK MESSAGES A&B
2583	013002	104035			ERROR	35	
2584	013004	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	:CLEAR CONTROLLER FOR NEXT TEST
2585	013012	000423			BR	TST20	::GO ON TO NEXT TEST
2586							
2587	013014						
2588	013014	013737	004150	001160	MOV	T.MR3,\$TMP0	:MASK OUT BITS NOT UNDER TEST
2589	013022	042737	140017	001160	BIC	#140017,\$TMP0	
2590	013030	001402			BEQ	4\$	:CHECK IF CYLINDER ADDRESS ZERO
2591	013032	104037			ERROR	37	:CYLINDER ADDRESS BITS INCORRECT
2592	013034	000412			BR	TST20	::GO ON TO NEXT TEST
2593							
2594	013036	023737	004206	004146	4\$: CMP	E.MR2,T.MR2	:CHECK IF MESSAGE A CORRECT
2595	013044	001401			BEQ	5\$	:YES, CHECK MESSAGE B
2596	013046	104040			ERROR	40	:MESS A INCORRECT
2597	013050	023737	004210	004150	5\$: CMP	E.MR3,T.MR3	:CHECK IF MESSAGE B CORRECT
2598	013056	001401			BEQ	TST20	:YES, GO ON TO NEXT TEST
2599	013060	104041			ERROR	41	:MESS B INCORRECT

```

*****
:TEST 20      CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 2)
:
: CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
: DIAGNOSTIC MODE.  LOAD CYLINDER ADDRESS REGISTER
: WITH 777.  LOAD THE OFFSET REG TO 52.  LOAD COMMAND
: AND STATUS REGISTER 1 WITH A PACK ACKNOWLEDGE.  CLOCK
: MESSAGES A AND B INTO SHIFT REGISTERS.  MAKE SURE
: SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER
: ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.
:
*****

```

2600							
2601							
2602							
2603							
2604							
2605							
2606							
2607							
2608							
2609							
2610							
2611							
2612							
2613	013062	000004			TST20: SCOPE		
2614	013064	012737	000144	001200	MOV	#100,\$TIMES	:DO 100. ITERATIONS
2615	013072	013702	001270		MOV	\$BASE,R2	:LOAD RK611 BASE
2616	013076	012737	001777	004252	MOV	#1777,CYLIN	:LOAD CYLINDER VALUE
2617	013104	012737	000052	004254	MOV	#52,OFFVAL	:LOAD OFFSET VALUE
2618	013112	012737	000003	004160	MOV	#PACK,E.CS1	:LOAD EXPECTED CS1
2619	013120	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	:CLEAR RK611
2620	013126	012762	000040	000026	MOV	#DMD,RKMR1(R2)	:PUT RK611 IN MAINTENANCE MODE
2621	013134	012762	001777	000020	MOV	#1777,RKDCYL(R2)	:LOAD CYLINDER VALUE
2622	013142	012762	000052	000016	MOV	#52,RKASOF(R2)	:LOAD OFFSET VALUE
2623	013150	012762	000003	000000	MOV	#PACK,RKCS1(R2)	:ISSUE PACK
2624	013156	012700	000016		MOV	#3*4+2,R0	:CLOCK IN DRIVE MESSAGE
2625	013162	012762	000440	000026	1\$: MOV	#DMD!MCLK,RKMR1(R2)	
2626	013170	012762	000040	000026	MOV	#DMD,RKMR1(R2)	
2627	013176	005300			DEC	R0	
2628	013200	001370			BNE	1\$	
2629	013202	016237	000000	004120	MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG. 1
2630	013210	016237	000034	004146	MOV	RKMR2(R2),T.MR2	:STORE MAINT REG. 2
2631	013216	016237	000036	004150	MOV	RKMR3(R2),T.MR3	:STORE MAINT REG. 3
2632	013224	012737	004000	004206	MOV	#S.PACK,E.MR2	:LOAD EXPECTED MAINT REG. 2
2633	013232	005037	004210		CLR	E.MR3	:LOAD EXPECTED MAINTENANCE REG. 3
2634	013236	023737	004160	004120	CMP	E.CS1,T.CS1	:CHECK IF CS1 CORRECT
2635	013244	001405			BEQ	2\$	:YES, CHECK MESSAGES A&B

```

2636 013246 104035          ERROR 35
2637 013250 012762 100000 000000  MOV  #CCLR,RKCS1(R2) ;CLEAR CONTROLLER FOR NEXT TEST
2638 013256 000431          BR    TST21           ;;GO ON TO NEXT TEST
2639
2640 013260          2$:
2641 013260 032737 004000 004146  BIT  #S.PACK,T.MR2   ;CHECK IF PACK COMMAND
2642          BIT SET
2643 013266 001002          BNE  3$             ;YES, CHECK CYLINDER ADDRESS BITS
2644 013270 104036          ERROR 36           ;S.PACK BIT NOT SET
2645 013272 000423          BR    TST21           ;;GO ON TO NEXT TEST
2646
2647 013274          3$:
2648 013274 013737 004150 001160  MOV  T.MR3,$TMP0    ;MASK OUT BITS NOT UNDER TEST
2649 013302 042737 140017 001160  BIC  #140017,$TMP0
2650 013310 001402          BEQ  4$             ;CHECK IF CYLINDER ADDRESS ZERO
2651 013312 104037          ERROR 37           ;CYLINDER ADDRESS BITS INCORRECT
2652 013314 000412          BR    TST21           ;;GO ON TO NEXT TEST
2653
2654 013316 023737 004206 004146  4$:  CMP  E.MR2,T.MR2    ;CHECK IF MESSAGE A CORRECT
2655 013324 001401          BEQ  5$             ;YES, CHECK MESSAGE B
2656 013326 104040          ERROR 40           ;MESS A INCORRECT
2657 013330 023737 004210 004150  5$:  CMP  E.MR3,T.MR3    ;CHECK IF MESSAGE B CORRECT
2658 013336 001401          BEQ  TST21         ;YES, GO ON TO NEXT TEST
2659 013340 104041          ERROR 41           ;MESS B INCORRECT
2660

```

```

*****
;TEST 21      CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 3)
*****

```

```

;
; CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
; DIAGNOSTIC MODE.  LOAD CYLINDER ADDRESS REGISTER
; WITH 777.  LOAD THE OFFSET REG TO 52.  LOAD COMMAND
; AND STATUS REGISTER 1 WITH A CLEAR DRIVE.  CLOCK
; MESSAGES A AND B INTO SHIFT REGISTERS.  MAKE SURE
; SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER
; ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.
;

```

```

*****
TST21:  SCOPE
2673 013342 000004          MOV  #100,$TIMES    ;;DO 100. ITERATIONS
2674 013344 012737 000144 001200  MOV  $BASE,R2      ;LOAD RK611 BASE
2675 013352 013702 001270          MOV  #1777,CYLIN   ;LOAD CYLINDER VALUE
2676 013356 012737 001777 004252  MOV  #52,OFFVAL    ;LOAD OFFSET VALUE
2677 013364 012737 000052 004254  MOV  #CLEAR,E.CS1  ;LOAD EXPECTED CS1
2678 013372 012737 000005 004160  MOV  #CCLR,RKCS1(R2);CLEAR RK611
2679 013400 012762 100000 000000  MOV  #DMD,RKMR1(R2);PUT RK611 IN MAINTENANCE MODE
2680 013406 012762 000040 000026  MOV  #1777,RKDCYL(R2);LOAD CYLINDER VALUE
2681 013414 012762 001777 000020  MOV  #52,RKASOF(R2);LOAD OFFSET VALUE
2682 013422 012762 000052 000016  MOV  #CLEAR,RKCS1(R2);ISSUE CLEAR
2683 013430 012762 000005 000000  MOV  #3*4+2,R0     ;CLOCK IN DRIVE MESSAGE
2684 013436 012700 000016          MOV  #DMD:MCLK,RKMR1(R2)
2685 013442 012762 000440 000026  1$:  MOV  #DMD,RKMR1(R2)
2686 013450 012762 000040 000026  MOV  R0
2687 013456 005300          DEC  R0
2688 013460 001370          BNE  1$
2689 013462 016237 000000 004120  MOV  RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2690 013470 016237 000034 004146  MOV  RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
2691 013476 016237 000036 004150  MOV  RKMR3(R2),T.MR3 ;STORE MAINT REG. 3

```

H06

```

2692 013504 012737 000400 004206 MOV #S.CLR,E.MR2 ;LOAD EXPECTED MAINT REG. 2
2693 013512 005037 004210 CLR E.MR3 ;LOAD EXPECTED MAINTENANCE REG. 3
2694 013516 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
2695 013524 001405 BEQ 2$ ;YES, CHECK MESSAGES A&B
2696 013526 104035 ERROR 35
2697 013530 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR CONTROLLER FOR NEXT TEST
2698 013536 000431 BR TST22 ;GO ON TO NEXT TEST
2699
2700 013540
2701 013540 032737 000400 004146 2$: BIT #S.CLR,T.MR2 ;CHECK IF CLEAR COMMAND
2702 ; BIT SET
2703 013546 001002 BNE 3$ ;YES, CHECK CYLINDER ADDRESS BITS
2704 013550 104036 ERROR 36 ;S.CLR BIT NOT SET
2705 013552 000423 BR TST22 ;GO ON TO NEXT TEST
2706
2707 013554
2708 013554 013737 004150 001160 3$: MOV T.MR3,$TMPD ;MASK OUT BITS NOT UNDER TEST
2709 013562 042737 140017 001160 BIC #140017,$TMPD
2710 013570 001402 BEQ 4$ ;CHECK IF CYLINDER ADDRESS ZERO
2711 013572 104037 ERROR 37 ;CYLINDER ADDRESS BITS INCORRECT
2712 013574 000412 BR TST22 ;GO ON TO NEXT TEST
2713
2714 013576 023737 004206 004146 4$: CMP E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
2715 013604 001401 BEQ 5$ ;YES, CHECK MESSAGE B
2716 013606 104040 ERROR 40 ;MESS A INCORRECT
2717 013610 023737 004210 004150 5$: CMP E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
2718 013616 001401 BEQ TST22 ;YES, GO ON TO NEXT TEST
2719 013620 104041 ERROR 41 ;MESS B INCORRECT

```

```

*****
*TEST 22 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 4)
*
* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
* DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER
* WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND
* AND STATUS REGISTER 1 WITH AN UNLOAD. CLOCK
* MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE
* SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER
* ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.
*****

```

```

2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733 013622 000004 TST22: SCOPE
2734 013624 012737 000144 001200 MOV #100,$TIMES ;DO 100. ITERATIONS
2735 013632 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE
2736 013636 012737 001777 004252 MOV #1777,CYLIN ;LOAD CYLINDER VALUE
2737 013644 012737 000052 004254 MOV #52,OFFVAL ;LOAD OFFSET VALUE
2738 013652 012737 000007 004160 MOV #UNLOAD,E.CS1 ;LOAD EXPECTED CS1
2739 013660 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611
2740 013666 012762 000040 000026 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
2741 013674 012762 001777 000020 MOV #1777,RKDCYL(R2) ;LOAD CYLINDER VALUE
2742 013702 012762 000052 000016 MOV #52,RKASOF(R2) ;LOAD OFFSET VALUE
2743 013710 012762 000007 000000 MOV #UNLOAD,RKCS1(R2) ;ISSUE UNLOAD
2744 013716 012700 000016 MOV #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
2745 013722 012762 000440 000026 1$: MOV #DMD!MCLK,RKMR1(R2)
2746 013730 012762 000040 000026 MOV #DMD,RKMR1(R2)
2747 013736 005300 DEC R0

```

```

2748 013740 001370          BNE      1$
2749 013742 016237 000000 004120  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2750 013750 016237 000034 004146  MOV     RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
2751 013756 016237 000036 004150  MOV     RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
2752 013764 012737 002000 004206  MOV     #S.UNLD,E.MR2 ;LOAD EXPECTED MAINT REG. 2
2753 013772 005037 004210          CLR     E.MR3 ;LOAD EXPECTED MAINTENANCE REG. 3
2754 013776 023737 004160 004120  CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
2755 014004 001405          BEQ     2$ ;YES, CHECK MESSAGES A&B
2756 014006 104035          ERROR   35
2757 014010 012762 100000 000000  MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER FOR NEXT TEST
2758 014016 000431          BR      TST23 ;GO ON TO NEXT TEST
2759
2760 014020          2$:
2761 014020 032737 002000 004146  BIT     #S.UNLD,T.MR2 ;CHECK IF UNLOAD COMMAND
2762          ; BIT SET
2763 014026 001002          BNE     3$ ;YES, CHECK CYLINDER ADDRESS BITS
2764 014030 104036          ERROR   36 ;S.UNLD BIT NOT SET
2765 014032 000423          BR      TST23 ;GO ON TO NEXT TEST
2766
2767 014034          3$:
2768 014034 013737 004150 001160  MOV     T.MR3,$TMPD ;MASK OUT BITS NOT UNDER TEST
2769 014042 042737 140017 001160  BIC     #140017,$TMPD
2770 014050 001402          BEQ     4$ ;CHECK IF CYLINDER ADDRESS ZERO
2771 014052 104037          ERROR   37 ;CYLINDER ADDRESS BITS INCORRECT
2772 014054 000412          BR      TST23 ;GO ON TO NEXT TEST
2773
2774 014056 023737 004206 004146  4$:    CMP     E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
2775 014064 001401          BEQ     5$ ;YES, CHECK MESSAGE B
2776 014066 104040          ERROR   40 ;MESS A INCORRECT
2777 014070 023737 004210 004150  5$:    CMP     E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
2778 014076 001401          BEQ     TST23 ;YES, GO ON TO NEXT TEST
2779 014100 104041          ERROR   41 ;MESS B INCORRECT

```

```

*****
*TEST 23      CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 5)
*
* CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
* DIAGNOSTIC MODE.  LOAD CYLINDER ADDRESS REGISTER
* WITH 777.  LOAD THE OFFSET REG TO 52.  LOAD COMMAND
* AND STATUS REGISTER 1 WITH A START SPINDLE.  CLOCK
* MESSAGES A AND B INTO SHIFT REGISTERS.  MAKE SURE
* SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER
* ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.
*****

```

```

2793 014102 000004          TST23: SCOPE
2794 014104 012737 000144 001200  MOV     #100,$TIMES ;DO 100. ITERATIONS
2795 014112 013702 001270          MOV     $BASE,R2 ;LOAD RK611 BASE
2796 014116 012737 001777 004252  MOV     #1777,CYLIN ;LOAD CYLINDER VALUE
2797 014124 012737 000052 004254  MOV     #52,OFFVAL ;LOAD OFFSET VALUE
2798 014132 012737 000011 004160  MOV     #SRTSPL,E.CS1 ;LOAD EXPECTED CS1
2799 014140 012762 100000 000000  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2800 014146 012762 000040 000026  MOV     #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
2801 014154 012762 001777 000020  MOV     #1777,RKDCYL(R2) ;LOAD CYLINDER VALUE
2802 014162 012762 000052 000016  MOV     #52,RKASOF(R2) ;LOAD OFFSET VALUE
2803 014170 012762 000011 000000  MOV     #SRTSPL,RKCS1(R2) ;ISSUE SRTSPL

```

```

2804 014176 012700 000016      MOV      #3*4+2,R0      ;CLOCK IN DRIVE MESSAGE
2805 014202 012762 000440 000026 1$:      MOV      #DMD!MCLK,RKMR1(R2)
2806 014210 012762 000040 000026      MOV      #DMD,RKMR1(R2)
2807 014216 005300      DEC      R0
2808 014220 001370      BNE     1$
2809 014222 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2810 014230 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINT REG. 2
2811 014236 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINT REG. 3
2812 014244 012737 000100 004206      MOV      #S.STSP,E.MR2   ;LOAD EXPECTED MAINT REG. 2
2813 014252 005037 004210      CLR      E.MR3           ;LOAD EXPECTED MAINTENANCE REG. 3
2814 014256 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK IF CS1 CORRECT
2815 014264 001405      BEQ     2$              ;YES, CHECK MESSAGES A&B
2816 014266 104035      ERROR   35
2817 014270 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER FOR NEXT TEST
2818 014276 000431      BR      TST24           ;;GO ON TO NEXT TEST
2819
2820 014300      2$:
2821 014300 032737 000100 004146      BIT      #S.STSP,T.MR2   ;CHECK IF SRTSPL COMMAND
2822                                ; BIT SET
2823 014306 001002      BNE     3$              ;YES, CHECK CYLINDER ADDRESS BITS
2824 014310 104036      ERROR   36
2825 014312 000423      BR      TST24           ;S.STSP BIT NOT SET
2826                                ;;GO ON TO NEXT TEST
2827
2828 014314 013737 004150 001160 3$:      MOV      T.MR3,$TMP0     ;MASK OUT BITS NOT UNDER TEST
2829 014322 042737 140017 001160      BIC     #140017,$TMP0
2830 014330 001402      BEQ     4$              ;CHECK IF CYLINDER ADDRESS ZERO
2831 014332 104037      ERROR   37
2832 014334 000412      BR      TST24           ;CYLINDER ADDRESS BITS INCORRECT
2833                                ;;GO ON TO NEXT TEST
2834 014336 023737 004206 004146 4$:      CMP      E.MR2,T.MR2     ;CHECK IF MESSAGE A CORRECT
2835 014344 001401      BEQ     5$              ;YES, CHECK MESSAGE B
2836 014346 104040      ERROR   40
2837 014350 023737 004210 004150 5$:      CMP      E.MR3,T.MR3     ;CHECK IF MESSAGE B CORRECT
2838 014356 001401      BEQ     TST24           ;YES, GO ON TO NEXT TEST
2839 014360 104041      ERROR   41
2840                                ;MESS B INCORRECT
    
```

\*\*\*\*\*  
 ;TEST 24 CYLINDER ADDRESS LOADING OF DRIVE MESS (PART 6)  
 ;

CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN  
 DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS REGISTER  
 WITH 777. LOAD THE OFFSET REG TO 52. LOAD COMMAND  
 AND STATUS REGISTER 1 WITH A RECALIBRATE. CLOCK  
 MESSAGES A AND B INTO SHIFT REGISTERS. MAKE SURE  
 SHIFT REGISTERS ARE LOADED CORRECTLY AND THE CYLINDER  
 ADDRESS FIELD IS ZERO IN DRIVE MESSAGE.

\*\*\*\*\*

```

2853 014362 000004      TST24: SCOPE
2854 014364 012737 000144 001200      MOV      #100,$TIMES    ;;DO 100. ITERATIONS
2855 014372 013702 001270      MOV      $BASE,R2       ;LOAD RK611 BASE
2856 014376 012737 001777 004252      MOV      #1777,CYLIN    ;LOAD CYLINDER VALUE
2857 014404 012737 000052 004254      MOV      #52,OFFVAL     ;LOAD OFFSET VALUE
2858 014412 012737 000013 004160      MOV      #RECAL,E.CS1   ;LOAD EXPECTED CS1
2859 014420 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
    
```

# K06

2860	014426	012762	000040	000026		MOV	#DMD,RKMR1(R2)	;PUT RK611 IN MAINTENANCE MODE
2861	014434	012762	001777	000020		MOV	#1777,RKDCYL(R2)	;LOAD CYLINDER VALUE
2862	014442	012762	000052	000016		MOV	#52,RKASOF(R2)	;LOAD OFFSET VALUE
2863	014450	012762	000013	000000		MOV	#RECAL,RKCS1(R2)	;ISSUE RECAL
2864	014456	012700	000016			MOV	#3*4+2,RO	;CLOCK IN DRIVE MESSAGE
2865	014462	012762	000440	000026	1\$:	MOV	#DMD!MCLK,RKMR1(R2)	
2866	014470	012762	000040	000026		MOV	#DMD,RKMR1(R2)	
2867	014476	005300				DEC	RO	
2868	014500	001370				BNE	1\$	
2869	014502	016237	000000	004120		MOV	RKCS1(R2),T.CS1	;STORE COMMAND AND STATUS REG. 1
2870	014510	016237	000034	004146		MOV	RKMR2(R2),T.MR2	;STORE MAINT REG. 2
2871	014516	016237	000036	004150		MOV	RKMR3(R2),T.MR3	;STORE MAINT REG. 3
2872	014524	012737	000040	004206		MOV	#S.RECL,E.MR2	;LOAD EXPECTED MAINT REG. 2
2873	014532	005037	004210			CLR	E.MR3	;LOAD EXPECTED MAINTENANCE REG. 3
2874	014536	023737	004160	004120		CMP	E.CS1,T.CS1	;CHECK IF CS1 CORRECT
2875	014544	001405				BEQ	2\$	;YES, CHECK MESSAGES A&B
2876	014546	104035				ERROR	3\$	
2877	014550	012762	100000	000000		MOV	#CLR,RKCS1(R2)	;CLEAR CONTROLLER FOR NEXT TEST
2878	014556	000431				BR	TST25	;GO ON TO NEXT TEST
2879								
2880	014560				2\$:			
2881	014560	032737	000040	004146		BIT	#S.RECL,T.MR2	;CHECK IF RECAL COMMAND
2882								;BIT SET
2883	014566	001002				BNE	3\$	;YES, CHECK CYLINDER ADDRESS BITS
2884	014570	104036				ERROR	36	
2885	014572	000423				BR	TST25	;S.RECL BIT NOT SET
2886								;GO ON TO NEXT TEST
2887	014574				3\$:			
2888	014574	013737	004150	001160		MOV	T.MR3,\$TMPD	;MASK OUT BITS NOT UNDER TEST
2889	014602	042737	140017	001160		BIC	#140017,\$TMPD	
2890	014610	001402				BEQ	4\$	;CHECK IF CYLINDER ADDRESS ZERO
2891	014612	104037				ERROR	37	;CYLINDER ADDRESS BITS INCORRECT
2892	014614	000412				BR	TST25	;GO ON TO NEXT TEST
2893								
2894	014616	023737	004206	004146	4\$:	CMP	E.MR2,T.MR2	;CHECK IF MESSAGE A CORRECT
2895	014624	001401				BEQ	5\$	;YES, CHECK MESSAGE B
2896	014626	104040				ERROR	40	;MESS A INCORRECT
2897	014630	023737	004210	004150	5\$:	CMP	E.MR3,T.MR3	;CHECK IF MESSAGE B CORRECT
2898	014636	001401				BEQ	TST25	;YES, GO ON TO NEXT TEST
2899	014640	104041				ERROR	41	;MESS B INCORRECT
2900								
2901								
2902								
2903								
2904								
2905								
2906								
2907								
2908								
2909								
2910								
2911	014642	000004				TST25:	SCOPE	
2912	014644	012737	000144	001200		MOV	#100,\$TIMES	;DO 100. ITERATIONS
2913	014652	013702	001270			MOV	\$BASE,R2	;LOAD RK611 BASE
2914	014656	012737	000017	004246		MOV	#17,MSGCOD	;LOAD MESSAGE CODE FOR PRINT OUT
2915	014664	012737	000003	004160		MOV	#PACK,E.CS1	;LOAD EXPECTED CS1

```

*****
*TEST 25      MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 1)
*
*      CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER IN
*      DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17.  LOAD
*      COMMAND AND STATUS REGISTER 1 WITH A PACK ACKNOWLEDGE.
*      CLOCK MESSAGE TO LOAD B SHIFT REG.  TIME.  MAKE SURE
*      MESSAGE SELECT BITS ARE CLEARED.
*****
  
```

L06

```

2916 014672 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
2917 014700 012762 000057 000026      MOV      #DMD!17,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
2918                                     ; SELECT MESSAGE 17
2919 014706 012762 000003 000000      MOV      #PACK,RKCS1(R2) ;ISSUE PACK
2920 014714 012700 000016 000000      MOV      #3*4+2,RO ;CLOCK IN DRIVE MESSAGE
2921 014720 052762 000400 000026 1$:    BIS      #MCLK,RKMR1(R2)
2922 014726 042762 000400 000026      BIC      #MCLK,RKMR1(R2)
2923 014734 005300      DEC      RO
2924 014736 001370      BNE      1$
2925 014740 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2926 014746 016237 000026 004144      MOV      RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2927 014754 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG.2
2928 014762 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
2929 014770 012737 002040 004204      MOV      #MEWD!DMD,E.MR1 ;LOAD EXPECTED MAINT REG. 1
2930 014776 032737 020000 004144      BIT      #ECCW,T.MR1
2931 015004 001403      BEQ      10$
2932 015006 052737 020000 004204      BIS      #ECCW,E.MR1
2933 015014 012737 004000 004206 10$:    MOV      #S.PACK,E.MR2 ;LOAD EXPECTED MAINT REG. 2
2934 015022 005037 004210 004120      CLR      E.MR3 ;LOAD EXPECTED MAINT REG. 3
2935 015026 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
2936 015034 001405      BEQ      2$ ;YES, CHECK MAINT REG. 1
2937 015036 104042      ERROR   42
2938 015040 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
2939 015046 000442      BR      TST26 ;GO ON TO NEXT TEST
2940
2941 015050 023737 004204 004144 2$:    CMP      E.MR1,T.MR1 ;CHECK IF MAINT REG. 1 CORRECT
2942 015056 001405      BEQ      3$ ;YES, CHECK MESSAGES A&B
2943 015060 104043      ERROR   43 ;MAINT REG. 1 INCORRECT
2944 015062 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
2945 015070 000431      BR      TST26 ;GO ON TO NEXT TEST
2946
2947 015072                                     3$:
2948 015072 032737 004000 004146      BIT      #S.PACK,T.MR2 ;CHECK IF PACK COMMAND
2949                                     ; BIT SET
2950 015100 001002      BNE      4$ ;YES, CHECK MESSAGE SELECT BITS
2951 015102 104044      ERROR   44 ;S.PACK BIT NOT SET
2952 015104 000423      BR      TST26 ;GO ON TO NEXT TEST
2953
2954 015106                                     4$:
2955 015106 013737 004150 001160      MOV      T.MR3,$TMP0 ;MASK OUT BITS NOT UNDER TEST
2956 015114 042737 177760 001160      BIC      #177760,$TMP0
2957 015122 001402      BEQ      5$ ;CHECK IF MESSAGE SELECT ZERO
2958 015124 104045      ERROR   45 ;MESSAGE SELECT BITS NOT ZERO
2959 015126 000412      BR      TST26 ;GO ON TO NEXT TEST
2960
2961 015130 023737 004206 004146 5$:    CMP      E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
2962 015136 001401      BEQ      6$ ;YES, CHECK MESSAGE B
2963 015140 104046      ERROR   46 ;MESSAGE A INCORRECT
2964 015142 023737 004210 004150 6$:    CMP      E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
2965 015150 001401      BEQ      TST26 ;YES, GO ON TO NEXT TEST
2966 015152 104047      ERROR   47 ;MESS B INCORRECT

```

```

2967
2968 ;*****
2969 ;*TEST 26 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 2)
2970 ;*
2971 ;* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN

```



```

3028 015442 023737 004206 004146 5$:    CMP    E.MR2,T.MR2    ;CHECK IF MESSAGE A CORRECT
3029 015450 001401                BEQ    5$            ;YES, CHECK MESSAGE B
3030 015452 104046                ERROR  46           ;MESSAGE A INCORRECT
3031 015454 023737 004210 004150 6$:    CMP    E.MR3,T.MR3    ;CHECK IF MESSAGE B CORRECT
3032 015462 001401                BEQ    TST27         ;YES, GO ON TO NEXT TEST
3033 015464 104047                ERROR  47           ;MESS B INCORRECT
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045 015466 000004                TST27: SCOPE
3046 015470 012737 000144 001200    MOV    #100,$TIMES    ;;DO 100. ITERATIONS
3047 015476 013702 001270                MOV    $BASE,R2      ;LOAD RK611 BASE
3048 015502 012737 000017 004246    MOV    #17,MSGCOD    ;LOAD MESSAGE CODE FOR PRINT OUT
3049 015510 012737 000007 004160    MOV    #UNLOAD,E.CS1 ;LOAD EXPECTED CS1
3050 015516 012762 100000 000000    MOV    #CCLR,RKCS1(R2);CLEAR RK611
3051 015524 012762 000057 000026    MOV    #DMD!17,RKMR1(R2);PUT RK611 IN MAINTENANCE MODE
3052
3053 015532 012762 000007 000000    MOV    #UNLOAD,RKCS1(R2);ISSUE UNLOAD
3054 015540 012700 000016                MOV    #3*4+2,R0     ;CLOCK IN DRIVE MESSAGE
3055 015544 052762 000400 000026 1$:    BIS    #MCLK,RKMR1(R2)
3056 015552 042762 000400 000026    BIC    #MCLK,RKMR1(R2)
3057 015560 005300                DEC    R0
3058 015562 001370                BNE    1$
3059 015564 016237 000000 004120    MOV    RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3060 015572 016237 000026 004144    MOV    RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
3061 015600 016237 000034 004146    MOV    RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG.2
3062 015606 016237 000036 004150    MOV    RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
3063 015614 012737 002040 004204    MOV    #MEWD!DMD,E.MR1 ;LOAD EXPECTED MAINT REG. 1
3064 015622 032737 020000 004144    BIT    #ECCW,T.MR1
3065 015630 001403                BEQ    10$
3066 015632 052737 020000 004204    BIS    #ECCW,E.MR1
3067 015640 012737 002000 004206 10$:    MOV    #S.UNLD,E.MR2  ;LOAD EXPECTED MAINT REG. 2
3068 015646 005037 004210                CLR    E.MR3         ;LOAD EXPECTED MAINT REG. 3
3069 015652 023737 004160 004120    CMP    E.CS1,T.CS1   ;CHECK IF CS1 CORRECT
3070 015660 001405                BEQ    2$            ;YES, CHECK MAINT REG. 1
3071 015662 104042                ERROR  42
3072 015664 012762 100000 000000    MOV    #CCLR,RKCS1(R2);CLEAR RK611 CONTROLLER FOR NEXT TEST
3073 015672 000442                BR     TST30         ;;GO ON TO NEXT TEST
3074
3075 015674 023737 004204 004144 2$:    CMP    E.MR1,T.MR1   ;CHECK IF MAINT REG. 1 CORRECT
3076 015702 001405                BEQ    3$            ;YES, CHECK MESSAGES A&B
3077 015704 104043                ERROR  43           ;MAINT REG. 1 INCORRECT
3078 015706 012762 100000 000000    MOV    #CCLR,RKCS1(R2);CLEAR RK611 CONTROLLER FOR NEXT TEST
3079 015714 000431                BR     TST30         ;;GO ON TO NEXT TEST
3080
3081 015716                3$:
3082 015716 032737 002000 004146    BIT    #S.UNLD,T.MR2 ;CHECK IF UNLOAD COMMAND
3083

```

; BIT SET

```

3084 015724 001002          BNE      45          ;YES, CHECK MESSAGE SELECT BITS
3085 015726 104044          ERROR    44          ;S.UNLD BIT NOT SET
3086 015730 000423          BR       TST30      ;GO ON TO NEXT TEST
3087
3088 015732                    45:
3089 015732 013737 004150 001160  MOV     T.MR3,$TMPD ;MASK OUT BITS NOT UNDER TEST
3090 015740 042737 177760 001160  BIC     #177760,$TMPD
3091 015746 001402          BEQ     55          ;CHECK IF MESSAGE SELECT ZERO
3092 015750 104045          ERROR    45          ;MESSAGE SELECT BITS NOT ZERO
3093 015752 000412          BR       TST30      ;GO ON TO NEXT TEST
3094
3095 015754 023737 004206 004146  55:    CMP     E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
3096 015762 001401          BEQ     65          ;YES, CHECK MESSAGE B
3097 015764 104046          ERROR    46          ;MESSAGE A INCORRECT
3098 015766 023737 004210 004150  65:    CMP     E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
3099 015774 001401          BEQ     TST30      ;YES, GO ON TO NEXT TEST
3100 015776 104047          ERROR    47          ;MESS B INCORRECT

```

```

*****
*TEST 30 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 4)
*
* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
* DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD
* COMMAND AND STATUS REGISTER 1 WITH A START SPINDLE.
* CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE
* MESSAGE SELECT BITS ARE CLEARED.
*****

```

```

3111
3112 016000 000004          TST30: SCOPE
3113 016002 012737 000144 001200  MOV     #100, $TIMES ;DO 100. ITERATIONS
3114 016010 013702 0001270  MOV     $BASE,R2 ;LOAD RK611 BASE
3115 016014 012737 000017 004246  MOV     #17,MSGCOD ;LOAD MESSAGE CODE FOR PRINT OUT
3116 016022 012737 000011 004160  MOV     #SRTSPL,E.CS1 ;LOAD EXPECTED CS1
3117 016030 012762 100000 000000  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3118 016036 012762 000057 000026  MOV     #DMD!17,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
3119
3120 016044 012762 000011 000000  MOV     #SRTSPL,RKCS1(R2) ;ISSUE SRTSPL
3121 016052 012700 000016  MOV     #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
3122 016056 052762 000400 000026  15:    BIS     #MCLK,RKMR1(R2)
3123 016064 042762 000400 000026  BIC     #MCLK,RKMR1(R2)
3124 016072 005300          DEC     R0
3125 016074 001370          BNE     15
3126 016076 016237 000000 004120  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3127 016104 016237 000026 004144  MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
3128 016112 016237 000034 004146  MOV     RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG.2
3129 016120 016237 000036 004150  MOV     RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
3130 016126 012737 002040 004204  MOV     #MEWD!DMD,E.MR1 ;LOAD EXPECTED MAINT REG. 1
3131 016134 032737 020000 004144  BIT     #ECCW,T.MR1
3132 016142 001403          BEQ     105
3133 016144 052737 020000 004204  BIS     #ECCW,E.MR1
3134 016152 012737 000100 004206  105:  MOV     #S.STSP,E.MR2 ;LOAD EXPECTED MAINT REG. 2
3135 016160 005037 004210  CLR     E.MR3 ;LOAD EXPECTED MAINT REG. 3
3136 016164 023737 004160 004120  CMP     E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
3137 016172 001405          BEQ     25          ;YES, CHECK MAINT REG. 1
3138 016174 104042          ERROR    42
3139 016176 012762 100000 000000  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST

```



```

3140 016204 000442 BR TST31 ;;GO ON TO NEXT TEST
3141
3142 016206 023737 004204 004144 2$: CMP E.MR1,T.MR1 ;CHECK IF MAINT REG. 1 CORRECT
3143 016214 001405 BEQ 3$ ;YES, CHECK MESSAGES A&B
3144 016216 104043 ERROR 43 ;MAINT REG. 1 INCORRECT
3145 016220 012762 100000 000000 MOV #CLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
3146 016226 000431 BR TST31 ;;GO ON TO NEXT TEST
3147
3148 016230 BIT #S.STSP,T.MR2 ;CHECK IF SRTSPL COMMAND
3149 016230 032737 000100 004146 3$: ; BIT SET
3150
3151 016236 001002 BNE 4$ ;YES, CHECK MESSAGE SELECT BITS
3152 016240 104044 ERROR 44 ;S.STSP BIT NOT SET
3153 016242 000423 BR TST31 ;;GO ON TO NEXT TEST
3154
3155 016244 MOV T.MR3,$TMPD ;MASK OUT BITS NOT UNDER TEST
3156 016244 013737 004150 001160 4$: BIC #177760,$TMPD
3157 016252 042737 177760 001160 BEQ 5$ ;CHECK IF MESSAGE SELECT ZERO
3158 016260 001402 ERROR 45 ;MESSAGE SELECT BITS NOT ZERO
3159 016262 104045 BR TST31 ;;GO ON TO NEXT TEST
3160 016264 000412
3161
3162 016266 023737 004206 004146 5$: CMP E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
3163 016274 001401 BEQ 6$ ;YES, CHECK MESSAGE B
3164 016276 104046 ERROR 46 ;MESSAGE A INCORRECT
3165 016300 023737 004210 004150 6$: CMP E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
3166 016306 001401 BEQ TST31 ;YES, GO ON TO NEXT TEST
3167 016310 104047 ERROR 47 ;MESS B INCORRECT
3168
3169
3170 *****
3171 *TEST 31 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 5)
3172 *
3173 * CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
3174 * DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD
3175 * COMMAND AND STATUS REGISTER 1 WITH A RECALIBRATE.
3176 * CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE
3177 * MESSAGE SELECT BITS ARE CLEARED.
3178 *****
3179 TST31: SCOPE
3180 016312 000004 MOV #100,$TIMES ;;DO 100. ITERATIONS
3181 016314 012737 000144 001200 MOV $BASE,R2 ;LOAD RK611 BASE
3182 016322 013702 001270 MOV #17,$MSGCOD ;LOAD MESSAGE CODE FOR PRINT OUT
3183 016326 012737 000017 004246 MOV #RECAL,E.CS1 ;LOAD EXPECTED CS1
3184 016334 012737 000013 004160 MOV #CLR,RKCS1(R2) ;CLEAR RK611
3185 016342 012762 100000 000000 MOV #DMD!17,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
3186 016350 012762 000057 000026 ; SELECT MESSAGE 17
3187 016356 012762 000013 000000 MOV #RECAL,RKCS1(R2) ;ISSUE RECAL
3188 016364 012700 000016 MOV #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
3189 016370 052762 000400 000026 1$: BIS #MCLK,RKMR1(R2)
3190 016376 042762 000400 000026 BIC #MCLK,RKMR1(R2)
3191 016404 005300 DEC R0
3192 016406 001370 BNE 1$
3193 016410 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3194 016416 016237 000026 004144 MOV RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
3195 016424 016237 000034 004146 MOV RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG.2

```

```

3196 016432 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
3197 016440 012737 002040 004204      MOV      #MEWD!DMD,E.MR1 ;LOAD EXPECTED MAINT REG. 1
3198 016446 032737 020000 004144      BIT      #ECCW,T.MR1
3199 016454 001403      BEQ      10$
3200 016456 052737 020000 004204      BIS      #ECCW,E.MR1
3201 016464 012737 000040 004206 10$:      MOV      #S.RECL,E.MR2 ;LOAD EXPECTED MAINT REG. 2
3202 016472 005037 004210      CLR      E.MR3 ;LOAD EXPECTED MAINT REG. 3
3203 016476 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
3204 016504 001405      BEQ      2$ ;YES, CHECK MAINT REG. 1
3205 016506 104042      ERROR   42
3206 016510 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
3207 016516 000442      BR      TST32 ;GO ON TO NEXT TEST
3208
3209 016520 023737 004204 004144 2$:      CMP      E.MR1,T.MR1 ;CHECK IF MAINT REG. 1 CORRECT
3210 016526 001405      BEQ      3$ ;YES, CHECK MESSAGES A&B
3211 016530 104043      ERROR   43 ;MAINT REG. 1 INCORRECT
3212 016532 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
3213 016540 000431      BR      TST32 ;GO ON TO NEXT TEST
3214
3215 016542      3$:
3216 016542 032737 000040 004146      BIT      #S.RECL,T.MR2 ;CHECK IF RECAL COMMAND
3217      ; BIT SET
3218 016550 001002      BNE     4$ ;YES, CHECK MESSAGE SELECT BITS
3219 016552 104044      ERROR   44 ;S.RECL BIT NOT SET
3220 016554 000423      BR      TST32 ;GO ON TO NEXT TEST
3221
3222 016556      4$:
3223 016556 013737 004150 001160      MOV      T.MR3,$TMPD ;MASK OUT BITS NOT UNDER TEST
3224 016564 042737 177760 001160      BIC      #177760,$TMPD
3225 016572 001402      BEQ      5$ ;CHECK IF MESSAGE SELECT ZERO
3226 016574 104045      ERROR   45 ;MESSAGE SELECT BITS NOT ZERO
3227 016576 000412      BR      TST32 ;GO ON TO NEXT TEST
3228
3229 016600 023737 004206 004146 5$:      CMP      E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
3230 016606 001401      BEQ      6$ ;YES, CHECK MESSAGE B
3231 016610 104046      ERROR   46 ;MESSAGE A INCORRECT
3232 016612 023737 004210 004150 6$:      CMP      E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
3233 016620 001401      BEQ      TST32 ;YES, GO ON TO NEXT TEST
3234 016622 104047      ERROR   47 ;MESS B INCORRECT
3235

```

```

*****
:TEST 32 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 6)
:
: CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
: DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD
: COMMAND AND STATUS REGISTER 1 WITH A OFFSET.
: CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE
: MESSAGE SELECT BITS ARE CLEARED.
*****

```

```

3245
3246 016624 000004      TST32: SCOPE
3247 016626 012737 000144 001200      MOV      #100,$TIMES ;DO 100. ITERATIONS
3248 016634 013702 001270      MOV      $BASE,R2 ;LOAD RK611 BASE
3249 016640 012737 000017 004246      MOV      #17,$MSGCOD ;LOAD MESSAGE CODE FOR PRINT OUT
3250 016646 012737 000015 004160      MOV      #OFFSET,E.CS1 ;LOAD EXPECTED CS1
3251 016654 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611

```

E07

```

3252 016662 012762 000057 000026      MOV      #DMD!17,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
3253                                     ;      SELECT MESSAGE 17
3254 016670 012762 000015 000000      MOV      #OFFSET,RKCS1(R2) ;ISSUE OFFSET
3255 016676 012700 000016 000000      MOV      #3*4+2,R0 ;CLOCK IN DRIVE MESSAGE
3256 016702 052762 000400 000026 1$:    BIS      #MCLK,RKMR1(R2)
3257 016710 042762 000400 000026      BIC      #MCLK,RKMR1(R2)
3258 016716 005300      DEC      R0
3259 016720 001370      BNE      1$
3260 016722 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3261 016730 016237 000026 004144      MOV      RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
3262 016736 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINTENANCE REG.2
3263 016744 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINTENANCE REG. 3
3264 016752 012737 002040 004204      MOV      #MEWD!DMD,E.MR1 ;LOAD EXPECTED MAINT REG. 1
3265 016760 032737 020000 004144      BIT      #ECCW,T.MR1
3266 016766 001403      BEQ      10$
3267 016770 052737 020000 004204      BIS      #ECCW,E.MR1
3268 016776 005037 004206 10$:    CLR      E.MR2 ;LOAD EXPECTED MAINT REG 2
3269 017002 012737 017760 004210      MOV      #17760,E.MR3 ;LOAD EXPECTED MAINT REG 3
3270 017010 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK IF CS1 CORRECT
3271 017016 001405      BEQ      2$ ;YES, CHECK MAINT REG. 1
3272 017020 104042      ERROR    42
3273 017022 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
3274 017030 000434      BR       TST33 ;GO ON TO NEXT TEST
3275
3276 017032 023737 004204 004144 2$:    CMP      E.MR1,T.MR1 ;CHECK IF MAINT REG. 1 CORRECT
3277 017040 001405      BEQ      3$ ;YES, CHECK MESSAGES A&B
3278 017042 104043      ERROR    43 ;MAINT REG. 1 INCORRECT
3279 017044 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER FOR NEXT TEST
3280 017052 000423      BR       TST33 ;GO ON TO NEXT TEST
3281
3282 017054 3$:
3283 017054 013737 004150 001160      MOV      T.MR3,$TMPD ;MASK OUT BITS NOT UNDER TEST
3284 017062 042737 177760 001160      BIC      #177760,$TMPD
3285 017070 001402      BEQ      5$ ;CHECK IF MESSAGE SELECT ZERO
3286 017072 104045      ERROR    45 ;MESSAGE SELECT BITS NOT ZERO
3287 017074 000412      BR       TST33 ;GO ON TO NEXT TEST
3288
3289 017076 023737 004206 004146 5$:    CMP      E.MR2,T.MR2 ;CHECK IF MESSAGE A CORRECT
3290 017104 001401      BEQ      6$ ;YES, CHECK MESSAGE B
3291 017106 104046      ERROR    46 ;MESSAGE A INCORRECT
3292 017110 023737 004210 004150 6$:    CMP      E.MR3,T.MR3 ;CHECK IF MESSAGE B CORRECT
3293 017116 001401      BEQ      TST33 ;YES, GO ON TO NEXT TEST
3294 017120 104047      ERROR    47 ;MESS B INCORRECT
3295

```

```

*****
;TEST 33 MESSAGE SELECT BIT CLEARING FOR CLASS A (PART 7)
;
; CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER IN
; DIAGNOSTIC MODE WITH MESSAGE SELECT BITS = 17. LOAD
; COMMAND AND STATUS REGISTER 1 WITH A SEEK.
; CLOCK MESSAGE TO LOAD B SHIFT REG. TIME. MAKE SURE
; MESSAGE SELECT BITS ARE CLEARED.
*****

```

```

3305
3306 017122 000004 3TST33: SCOPE
3307 017124 012737 000144 001200      MOV      #100.,$TIMES ;;DO 100. ITERATIONS

```

3308	017132	013702	001270		MOV	\$BASE,R2	;LOAD RK611 BASE
3309	017136	012737	000017	004246	MOV	#17,MSGCOD	;LOAD MESSAGE CODE FOR PRINT OUT
3310	017144	012737	000017	004160	MOV	#SEEK,E.CS1	;LOAD EXPECTED CS1
3311	017152	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR RK611
3312	017160	012762	000057	000026	MOV	#DMD!17,RKMR1(R2)	;PUT RK611 IN MAINTENANCE MODE
3313							;SELECT MESSAGE 17
3314	017166	012762	000017	000000	MOV	#SEEK,RKCS1(R2)	;ISSUE SEEK
3315	017174	012700	000016		MOV	#3*4+2,R0	;CLOCK IN DRIVE MESSAGE
3316	017200	052762	000400	000026	1\$: BIS	#MCLK,RKMR1(R2)	
3317	017206	042762	000400	000026	BIC	#MCLK,RKMR1(R2)	
3318	017214	005300			DEC	R0	
3319	017216	001370			BNE	1\$	
3320	017220	016237	000000	004120	MOV	RKCS1(R2),T.CS1	;STORE COMMAND AND STATUS REG. 1
3321	017226	016237	000026	004144	MOV	RKMR1(R2),T.MR1	;STORE MAINTENANCE REG. 1
3322	017234	016237	000034	004146	MOV	RKMR2(R2),T.MR2	;STORE MAINTENANCE REG.2
3323	017242	016237	000036	004150	MOV	RKMR3(R2),T.MR3	;STORE MAINTENANCE REG. 3
3324	017250	012737	002040	004204	MOV	#MEWD!DMD,E.MR1	;LOAD EXPECTED MAINT REG. 1
3325	017256	032737	020000	004144	BIT	#ECCW,T.MR1	
3326	017264	001403			BEQ	10\$	
3327	017266	052737	020000	004204	BIS	#ECCW,E.MR1	
3328	017274	012737	000020	004206	10\$: MOV	#S.SEEK,E.MR2	;LOAD EXPECTED MAINT REG. 2
3329	017302	005037	004210		CLR	E.MR3	;LOAD EXPECTED MAINT REG. 3
3330	017306	023737	004160	004120	CMF	E.CS1,T.CS1	;CHECK IF CS1 CORRECT
3331	017314	001405			BEQ	2\$	;YES, CHECK MAINT REG. 1
3332	017316	104042			ERROR	42	
3333	017320	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR RK611 CONTROLLER FOR NEXT TEST
3334	017326	000442			BR	TST34	;GO ON TO NEXT TEST
3335							
3336	017330	023737	004204	004144	2\$: CMP	E.MR1,T.MR1	;CHECK IF MAINT REG. 1 CORRECT
3337	017336	001405			BEQ	3\$	;YES, CHECK MESSAGES A&B
3338	017340	104043			ERROR	43	;MAINT REG. 1 INCORRECT
3339	017342	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR RK611 CONTROLLER FOR NEXT TEST
3340	017350	000431			BR	TST34	;GO ON TO NEXT TEST
3341							
3342	017352				3\$: BIT	#S.SEEK,T.MR2	;CHECK IF SEEK COMMAND
3343	017352	032737	000020	004146			;BIT SET
3344							
3345	017360	001002			BNE	4\$	;YES, CHECK MESSAGE SELECT BITS
3346	017362	104044			ERROR	44	;S.SEEK BIT NOT SET
3347	017364	000423			BR	TST34	;GO ON TO NEXT TEST
3348							
3349	017366				4\$: MOV	T.MR3,\$TMPD	;MASK OUT BITS NOT UNDER TEST
3350	017366	013737	004150	001160	BIC	#177760,\$TMPD	
3351	017374	042737	177760	001160			
3352	017402	001402			BEQ	5\$	;CHECK IF MESSAGE SELECT ZERO
3353	017404	104045			ERROR	45	;MESSAGE SELECT BITS NOT ZERO
3354	017406	000412			BR	TST34	;GO ON TO NEXT TEST
3355							
3356	017410	023737	004206	004146	5\$: CMP	E.MR2,T.MR2	;CHECK IF MESSAGE A CORRECT
3357	017416	001401			BEQ	6\$	;YES, CHECK MESSAGE B
3358	017420	104046			ERROR	46	;MESSAGE A INCORRECT
3359	017422	023737	004210	004150	6\$: CMP	E.MR3,T.MR3	;CHECK IF MESSAGE B CORRECT
3360	017430	001401			BEQ	TST34	;YES, GO ON TO NEXT TEST
3361	017432	104047			ERROR	47	;MESS B INCORRECT
3362							
3363							

.SBTTL \*\*DRIVE MESSAGE LOOPBACK AND PARITY GENERATION TESTS

3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411  
3412  
3413  
3414  
3415  
3416  
3417  
3418  
3419

017434 000004  
017436 012737 000144 001200  
017444 013702 001270  
017450 012762 100000 000000  
017456 005037 004256  
017462 012737 000005 004206  
017470 012737 000005 004230  
017476 012737 000003 004210  
017504 012737 000003 004232  
017512 012762 000043 000026  
017520 012762 000005 000010  
017526 012762 000001 000000  
017534 012700 000016  
017540 052762 000400 000026 1\$:  
017546 042762 000400 000026  
017554 005300  
017556 001370  
017560 016237 000034 004146  
017566 016237 000036 004150  
017574 023737 004206 004146  
017602 001402  
017604 104050  
017606 000431  
017610 023737 004210 004150 2\$:  
017616 001402  
017620 104051  
017622 000423  
017624 032737 000001 004210 3\$:  
017632 001402  
017634 000261  
017636 000401  
017640 000241 4\$:  
017642 006037 004206 5\$:  
017646 006037 004210  
017652 012700 000004  
017656 005237 004256  
017662 022737 000004 004256  
017670 103323

```
*****
*TEST 34 DRIVE MESSAGE LOOPBACK
*
* CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
* IN DIAGNOSTIC MODE INDICATING MESSAGE 3. LOAD COMMAND
* STATUS REGISTER FOR DRIVE 5. LOAD COMMAND AND STATUS
* REGISTER 1 WITH A SELECT COMMAND. CLOCK 4 BITS
* THROUGH THE DRIVE MESSAGE LOOPBACK. VERIFY THAT BITS
* ARE INDEED LOOPED BACK.
*****
TST34: SCOPE
MOV #100, $TIMES ;:DO 100. ITERATIONS
MOV $BASE, R2 ;:LOAD RK611 BASE
MOV #CLR, RKCS1(R2) ;:CLEAR RK611
CLR SFTCNT ;:INITIALIZE SHIFT COUNT
MOV #5, E.MR2 ;:LOAD EXPECTED SHIFT REG. A
MOV #5, U.MR2 ;:LOAD UNSHIFTED SHIFT REG. A
MOV #3, E.MR3 ;:LOAD EXPECTED SHIFT REG. B
MOV #3, U.MR3 ;:LOAD UNSHIFTED SHIFT REG. B
MOV #DMD!3, RKMR1(R2) ;:PUT RK611 IN MAINT. MODE
;: MESSAGE SELECT = 3
MOV #5, RKCS2(R2) ;:LOAD DRIVE NUMBER = 5
MOV #SELDRV, RKCS1(R2) ;:ISSUE SELECT DRIVE
MOV #3*4+2, R0 ;:CLOCK IN MESSAGE
1$: BIS #MCLK, RKMR1(R2) ;:ISSUE CLOCKS
BIC #MCLK, RKMR1(R2)
DEC R0
BNE 1$
MOV RKMR2(R2), T.MR2 ;:STORE SHIFT REG. A
MOV RKMR3(R2), T.MR3 ;:STORE SHIFT REG. B
CMP E.MR2, T.MR2 ;:CHECK SHIFT REG A CORRECT
BEQ 2$ ;:YES, CHECK SHIFT REG. B
ERROR 50 ;:SHIFT REG A INCORRECT
BR TST35 ;:GO ON TO NEXT TEST
3402: CMP E.MR3, T.MR3 ;:CHECK SHIFT REG B CORRECT
BEQ 3$ ;:YES, SHIFT A BIT
ERROR 51 ;:SHIFT REG B INCORRECT
BR TST35 ;:GO ON TO NEXT TEST
3407: BIT #BIT0, E.MR3 ;:CHECK IF SHIFT BIT = 1
BEQ 4$ ;:NO, CLEAR SHIFT BIT
SEC ;:SET SHIFT BIT
BR 5$ ;:GENERATE EXPECTED SHIFT
;: REGISTERS A & B
4$: CLC ;:CLEAR SHIFT BIT
5$: ROR E.MR2 ;:GENERATE EXPECTED SHIFT REG A
ROR E.MR3 ;:GENERATE EXPECTED SHIFT REG B
MOV #4, R0 ;:LOAD COUNT FOR 1 BIT SHIFT
INC SFTCNT ;:INCREMENT SHIFT BIT COUNT
CMP #4, SFTCNT ;:CHECK IF FINISHED
BHS 1$ ;:NO, SHIFT IN NEXT BIT
```

3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466  
3467  
3468  
3469  
3470  
3471  
3472  
3473  
3474  
3475

017672 000004  
017674 012737 000144 001200  
017702 013702 001270  
017706 012762 100000 000000  
017714 005037 004256  
017720 012737 011020 004206  
017726 012737 011020 004230  
017734 012737 011020 004210  
017742 012737 011020 004232  
017750 012762 000040 000026  
017756 012762 000441 000020  
017764 012762 000400 000006  
017772 012762 010017 000000  
  
020000 012700 000016  
020004 052762 000400 000026 1\$:  
020012 042762 000400 000026  
020020 005300  
020022 001370  
020024 016237 000034 004146  
020032 016237 000036 004150  
020040 023737 004206 004146  
020046 001402  
020050 104050  
020052 000431  
  
020054 023737 004210 004150 2\$:  
020062 001402  
020064 104051  
020066 000423  
  
020070 032737 000001 004210 3\$:  
020076 001402  
020100 000261  
020102 000401  
  
020104 000241 4\$:  
020106 006037 004206 5\$:  
020112 006037 004210  
020116 012700 000004  
020122 005237 004256  
020126 022737 000010 004256  
020134 103323

```
*****
*TEST 35 DRIVE MESSAGE SHIFT
*
* CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
* IN DIAGNOSTIC MODE. LOAD CYLINDER ADDRESS WITH 441.
* LOAD HEAD ADDRESS WITH 1. LOAD COMMAND AND STATUS
* REGISTER 1 WITH A SEEK IN 24 SECTOR MODE. CLOCK 8 BITS
* THROUGH THE DRIVE MESSAGE LOOPBACK. VERIFY THAT BITS ARE
* SHIFTED PROPERLY.
*****
TST35: SCOPE
MOV #100, $TIMES ;:DO 100. ITERATIONS
MOV $BASE, R2 ;:LOAD RK611 BASE
MOV #CCLR, RKCS1(R2) ;:CLEAR RK611
CLR SFTCNT ;:INITIALIZE SHIFT COUNT
MOV #S.FMT!S.SEEK!BIT12, E.MR2 ;:LOAD EXPECTED SHIFT REG. A
MOV #S.FMT!S.SEEK!BIT12, U.MR2 ;:LOAD UNSHIFTED SHIFT REG. A
MOV #11020, E.MR3 ;:LOAD EXPECTED SHIFT REG. B
MOV #11020, U.MR3 ;:LOAD UNSHIFTED SHIFT REG. B
MOV #DMD, RKMR1(R2) ;:PUT RK611 IN MAINT. MODE
MOV #441, RKDCYL(R2) ;:LOAD CYLINDER ADD. REG.
MOV #400, RKDA(R2) ;:LOAD DISK ADDRESS REG.
MOV #SEEK!CFMT, RKCS1(R2) ;:ISSUE SEEK

MOV #3*4+2, R0 ;:CLOCK IN MESSAGE
1$: BIS #MCLK, RKMR1(R2) ;:ISSUE CLOCKS
BIC #MCLK, RKMR1(R2)
R0
DEC R0
BNE 1$
MOV RKMR2(R2), T.MR2 ;:STORE SHIFT REG. A
MOV RKMR3(R2), T.MR3 ;:STORE SHIFT REG. B
CMP E.MR2, T.MR2 ;:CHECK SHIFT REG A CORRECT
BEQ 2$ ;:YES, CHECK SHIFT REG. B
ERROR 50 ;:SHIFT REG A INCORRECT
BR TST36 ;:GO ON TO NEXT TEST

2$: CMP E.MR3, T.MR3 ;:CHECK SHIFT REG B CORRECT
BEQ 3$ ;:YES, SHIFT A BIT
ERROR 51 ;:SHIFT REG B INCORRECT
BR TST36 ;:GO ON TO NEXT TEST

3$: BIT #BIT0, E.MR3 ;:CHECK IF SHIFT BIT = 1
BEQ 4$ ;:NO, CLEAR SHIFT BIT
SEC ;:SET SHIFT BIT
BR 5$ ;:GENERATE EXPECTED SHIFT
; REGISTERS A & B

4$: CLC ;:CLEAR SHIFT BIT
5$: ROR E.MR2 ;:GENERATE EXPECTED SHIFT REG A
ROR E.MR3 ;:GENERATE EXPECTED SHIFT REG B
MOV #4, R0 ;:LOAD COUNT FOR 1 BIT SHIFT
INC SFTCNT ;:INCREMENT SHIFT BIT COUNT
CMP #8, SFTCNT ;:CHECK IF FINISHED
BHS 1$ ;:NO, SHIFT IN NEXT BIT
```

3476  
3477  
3478  
3479  
3480  
3481  
3482  
3483  
3484  
3485  
3486  
3487  
3488  
3489  
3490  
3491  
3492  
3493  
3494  
3495  
3496  
3497  
3498  
3499  
3500  
3501  
3502  
3503  
3504  
3505  
3506  
3507  
3508  
3509  
3510  
3511  
3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531

020136 000004  
020140 012737 000144 001200  
020146 013702 001270  
020152 012762 100000 000000  
020160 012762 000040 000026  
020166 012762 000001 000000  
020174 012700 000116  
020200 012762 000440 000026  
020206 012762 000040 000026  
020214 005300  
020216 001370  
020220 016237 000034 004146  
020226 016237 000036 004150  
020234 012737 100000 004206  
020242 012737 100000 004210  
020250 032737 100000 004150  
  
020256 001002  
020260 104052  
020262 000420  
  
020264 032737 100000 004146  
020272 001002  
020274 104053  
020276 000412  
  
020300 023737 004210 004150  
020306 001401  
020310 104054  
020312 023737 004206 004146  
020320 001401  
020322 104055  
020324 012762 100000 000000  
020332 012762 000060 000026  
  
020340 012762 000001 000000  
020346 012700 000116  
020352 012762 000460 000026  
020360 012762 000060 000026  
020366 005300  
  
020370 001370  
020372 016237 000034 004146  
020400 016237 000036 004150  
020406 005037 004206

```
*****  
*TEST 36 DRIVE MESSAGE PARITY PRECONDITIONING  
*  
* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER  
* IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 1 WITH  
* A SELECT MESSAGE. CLOCK ALL 16 BITS THROUGH THE  
* DRIVE MESSAGE LOOPBACK. VERIFY PARITY HAS BEEN PRECONDITIONED  
* PROPERLY. REPEAT FOR BAD PARITY GENERATION.  
*  
*****  
TST36: SCOPE  
MOV #100,STIMES ;DO 100. ITERATIONS  
MOV $BASE,R2 ;LOAD RK611 BASE  
MOV #CCLR,RKCS1(R2) ;CLEAR RK611  
MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE  
MOV #SELDRV,RKCS1(R2) ;ISSUE SELECT DRIVE  
MOV #19.*4+2,RO ;LOAD DRIVE MESSAGE AND SHIFT  
1$: MOV #DMD!MCLK,RKMR1(R2) ; ALL 16 BITS  
MOV #DMD,RKMR1(R2)  
DEC RO  
BNE 1$  
MOV RKMR2(R2),T.MR2 ;STORE SHIFTED MESSAGE B  
MOV RKMR3(R2),T.MR3 ;STORE SHIFTED MESSAGE A  
MOV #100000,E.MR2 ;LOAD EXPECTED MESSAGE B  
MOV #100000,E.MR3 ;LOAD EXPECTED MESSAGE A  
BIT #BIT15,T.MR3 ;CHECK IF PARITY ON MESSAGE A CORRECT  
  
BNE 2$ ;YES, CHECK PARITY ON MESSAGE B  
ERROR 52 ;PARITY ON MESSAGE A INCORRECT  
BR 5$ ;TRY EVEN PARITY  
  
2$: BIT #BIT15,T.MR2 ;CHECK IF PARITY ON MESS B CORRECT  
BNE 3$ ;YES, CHECK MESSAGE A AND B  
ERROR 53 ;PARITY ON MESSAGE B INCORRECT  
BR 5$ ;TRY EVEN PARITY  
  
3$: CMP E.MR3,T.MR3 ;CHECK IF MESSAGE A CORRECT  
BEQ 4$ ;YES, CHECK MESSAGE B  
ERROR 54 ;MESSAGE A INCORRECT  
4$: CMP E.MR2,T.MR2 ;CHECK IF MESSAGE B CORRECT  
BEQ 5$ ;YES, TRY EVEN PARITY  
ERROR 55 ;MESSAGE B INCORRECT  
5$: MOV #CCLR,RKCS1(R2) ;CLEAR RK611  
MOV #DMD!PAT,RKMR1(R2) ;PUT RK611 MAINTENANCE MODE  
; AND EVEN PARITY  
MOV #SELDRV,RKCS1(R2) ;ISSUE SELECT DRIVE  
MOV #19.*4+2,RO ;LOAD DRIVE MESSAGE AND SHIFT  
6$: MOV #DMD!PAT!MCLK,RKMR1(R2) ; ALL 16 BITS  
MOV #DMD!PAT,RKMR1(R2)  
DEC RO  
  
BNE 6$  
MOV RKMR2(R2),T.MR2 ;STORE SHIFTED MESSAGE B  
MOV RKMR3(R2),T.MR3 ;STORE SHIFTED MESSAGE A  
CLR E.MR2 ;LOAD EXPECTED MESSAGE B
```

```

3532 020412 005037 004210          CLR      E.MR3      ;LOAD EXPECTED MESSAGE A
3533 020416 032737 100000 004150      BIT      #BIT15,T.MR3 ;CHECK IF PARITY ON MESSAGE A CORRECT
3534 020424 001402          BEQ      7$        ;YES, CHECK PARITY ON MESSAGE B
3535 020426 104056          ERROR   56        ;PARITY ON MESSAGE A INCORRECT
3536 020430 000420          BR      TST37     ;GO ON TO NEXT TEST
3537
3538 020432 032737 100000 004146 7$:      BIT      #BIT15,T.MR2 ;CHECK IF PARITY ON MESS B CORRECT
3539 020440 001402          BEQ      8$        ;YES, CHECK MESSAGE A AND B
3540 020442 104057          ERROR   57        ;PARITY ON MESSAGE B INCORRECT
3541 020444 000412          BR      TST37     ;GO ON TO NEXT TEST
3542
3543 020446 023737 004210 004150 8$:      CMP      E.MR3,T.MR3 ;CHECK IF MESSAGE A CORRECT
3544 020454 001401          BEQ      9$        ;YES, CHECK MESSAGE B
3545 020456 104060          ERROR   60        ;MESSAGE A INCORRECT
3546 020460 023737 004206 004146 9$:      CMP      E.MR2,T.MR2 ;CHECK IF MESSAGE B CORRECT
3547 020466 001401          BEQ      TST37    ;YES, GO ON TO NEXT TEST
3548 020470 104061          ERROR   61        ;MESSAGE B INCORRECT
3549

```

```

*****
*TEST 37          ODD DRIVE MESSAGE PARITY GENERATION
*

```

```

* CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER
* IN DIAGNOSTIC MODE AND MESSAGE SELECT = 1.
* LOAD COMMAND AND STATUS REGISTER 2 WITH DRIVE
* SELECT = 1. LOAD COMMAND AND STATUS REGISTER 1 WITH
* A SELECT COMMAND. VERIFY THAT PARITY HAS BEEN
* GENERATED CORRECTLY. REPEAT FOR MESSAGE SELECT =
* DRIVE SELECT = 2-17.
*

```

```

*****

```

```

3562 020472 000004          TST37:  SCOPE
3563 020474 012737 000144 001200      MOV      #100, $TIMES ;DO 100. ITERATIONS
3564 020502 013702 001270          MOV      $BASE,R2    ;LOAD RK611 BASE
3565 020506 012737 000001 004244      MOV      #1, DRVCOD  ;LOAD DRIVE CODE
3566 020514 012737 020522 001110      MOV      #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
3567                                     ; SUBTEST LOOP
3568
3569 020522          1$:
3570 020522 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
3571 020530 013762 004244 000026      MOV      DRVCOD,RKMR1(R2) ;LOAD MESSAGE SELECT CODE
3572 020536 052762 000040 000026      BIS      #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
3573 020544 013762 004244 000010      MOV      DRVCOD,RKCS2(R2) ;LOAD DRIVE SELECT CODE
3574 020552 012762 000001 000000      MOV      #SELDV,RKCS1(R2) ;ISSUE SELECT DRIVE
3575 020560 012700 000116          MOV      #19,*4+2,R0 ;LOAD DRIVE MESSAGE AND SHIFT
3576 020564 052762 000400 000026 2$:      BIS      #MCLK,RKMR1(R2) ; ALL 16 BITS
3577 020572 042762 000400 000026      BIC      #MCLK,RKMR1(R2)
3578 020600          DEC      R0
3579 020602 001370          BNE     2$
3580 020604 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE SHIFTED MESSAGE B
3581 020612 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE SHIFTED MESSAGE A
3582 020620 013701 004244          MOV      DRVCOD,R1    ;DETERMINE PARITY
3583 020624 012703 000004          MOV      #4,R3
3584 020630 005004          CLR     R4
3585 020632 006001          3$:      ROR     R1
3586 020634 103001          BCC     4$
3587 020636 005204          INC     R4

```



```

3588 020640 005303          4$: DEC      R3
3589 020642 001373          BNE      3$
3590 020644 013737 004244 004206  MOV     DRVCOD,E.MR2 ;LOAD EXPECTED SHIFTED REG. B
3591 020652 013737 004244 004210  MOV     DRVCOD,E.MR3 ;LOAD EXPECTED SHIFTED REG. A
3592 020660 005037 004260          CLR     PARBIT
3593 020664 032704 000001          BIT     #BIT0,R4 ;CHECK FOR PARITY ON WORD
3594 020670 001011          BNE     5$ ;PARITY ALREADY ODD
3595 020672 012737 100000 004260  MOV     #BIT15,PARBIT ;SET PARITY BIT
3596 020700 052737 100000 004206  BIS     #BIT15,E.MR2
3597 020706 052737 100000 004210  BIS     #BIT15,E.MR3
3598 020714 013737 004150 001160  5$: MOV     T.MR3,$TMP0 ;MASK ALL BITS EXCEPT PARITY
3599 020722 042737 077777 001160  BIC     #77777,$TMP0
3600 020730 023737 004260 001160  CMP     PARBIT,$TMP0 ;CHECK IF PARITY CORRECT
3601 020736 001402          BEQ     6$ ;ON MESSAGE A
3602 020740 104052          ERROR  52 ;PARITY ON MESSAGE A INCORRECT
3603 020742 000426          BR      25$ ;CHECK IF LOOP ON ERROR
3604
3605 020744 013737 004146 001160  6$: MOV     T.MR2,$TMP0 ;MASK ALL BITS EXCEPT PARITY
3606 020752 042737 077777 001160  BIC     #77777,$TMP0
3607 020760 023737 004260 001160  CMP     PARBIT,$TMP0 ;CHECK IF PARITY CORRECT
3608 020766 001402          BEQ     7$ ;ON MESSAGE B
3609 020770 104053          ERROR  53 ;PARITY ON MESSAGE B INCORRECT
3610 020772 000412          BR      25$ ;CHECK IF LOOP ON ERROR
3611
3612 020774 023737 004210 004150  7$: CMP     E.MR3,T.MR3 ;CHECK IF MESSAGE A CORRECT
3613 021002 001401          BEQ     8$ ;YES, CHECK MESSAGE B
3614 021004 104054          ERROR  54 ;MESSAGE A INCORRECT
3615 021006 023737 004206 004146  8$: CMP     E.MR2,T.MR2 ;CHECK IF MESSAGE B CORRECT
3616 021014 001401          BEQ     25$ ;YES, CHECK IF LOOP ON ERROR
3617 021016 104055          ERROR  55 ;MESSAGE B INCORRECT
3618 021020 104415          25$: SCOP1 ;CHECK IF LOOP ON ERROR
3619 021022 005237 004244          INC     DRVCOD ;INCREMENT DRIVE SELECT CODE
3620 021026 022737 000017 004244  CMP     #17,DRVCOD ;CHECK IF FINISHED
3621 021034 103232          BHIS   1$ ;NO, TRY NEXT CONFIGURATION
3622
3623 ;*****
3624 ;*TEST 40 DRIVE MESSAGE PARITY INTERACTION
3625 ;*
3626 ;* CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
3627 ;* IN DIAGNOSTIC MODE. LOAD COMMAND AND STATUS REGISTER 2
3628 ;* WITH DRIVE SELECT = 1. LOAD COMMAND AND STATUS REGISTER 1
3629 ;* WITH A SELECT COMMAND. VERIFY THAT THE CORRECT PARITY
3630 ;* IS GENERATED FOR BOTH MESSAGES. REPEAT FOR MESSAGE
3631 ;* SELECT = 1 AND DRIVE SELECT = 0.
3632 ;*
3633 ;*****
3634 ;*ST40: SCOPE
3635 021036 000004          MOV     #100,$TIMES ;DO 100. ITERATIONS
3636 021040 012737 000144 001200  MOV     $BASE,R2 ;LOAD RK611 BASE
3637 021052 012737 000001 004244  MOV     #1,DRVCOD ;SET INITIAL DRIVE SELECT CODE
3638 021060 005037 004246          CLR     MSGCOD ;SET INITIAL MESSAGE SELECT CODE
3639 021064 012737 100000 004206  MOV     #BIT15,E.MR2 ;LOAD EXPECTED MAINT. REG. 2 (MESS B)
3640 021072 012737 000001 004210  MOV     #BIT0,E.MR3 ;LOAD EXPECTED MAINT. REG. 3 (MESS A)
3641 021100 012737 100000 004260  MOV     #BIT15,PARBIT ;LOAD PARITY FOR MESSAGE B
3642 021106 012737 021114 001110  MOV     #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
3643 ; SUBTEST LOOP

```

```

3644
3645 021114
3646 021114 012762 100000 000000 1$: MOV #CCLR,RKCS1(R2) ;CLEAR RK611
3647 021122 013762 004246 000026 MOV MSGCOD,RKMR1(R2) ;LOAD MESSAGE SELECT CODE
3648 021130 052762 000040 000026 BIS #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
3649 021136 013762 004244 000010 MOV DRVCOD,RKCS2(R2) ;LOAD DRIVE SELECT CODE
3650 021144 012762 000001 000000 MOV #SELDRV,RKCS1(R2) ;ISSUE DRIVE SELECT
3651 021152 012700 000116 MOV #19.*4+2,R0 ;LOAD DRIVE MESSAGE AND SHIFT
3652 021156 052762 000400 000026 2$: BIS #MCLK,RKMR1(R2) ; ALL 16 BITS
3653 021164 042762 000400 000026 BIC #MCLK,RKMR1(R2)
3654 021172 005300 DEC R0
3655 021174 001370 BNE 2$
3656 021176 016237 000034 004146 MOV RKMR2(R2),T.MR2 ;STORE SHIFTED MESSAGE B
3657 021204 016237 000036 004150 MOV RKMR3(R2),T.MR3 ;STORE SHIFTED MESSAGE A
3658 021212 013737 004150 001160 MOV T.MR3,$TMP0 ;MASK ALL BITS EXCEPT PARITY
3659 021220 042737 077777 001160 BIC #77777,$TMP0
3660 021226 023737 004260 001160 CMP PARBIT,$TMP0 ;CHECK IF PARITY BIT CORRECT
3661 021234 001002 BNE 3$ ; ON MESSAGE A
3662 021236 104052 ERROR 52 ;NO, PARITY ON MESSAGE INCORRECT
3663 021240 000426 BR 25$ ;CHECK IF LOOP ON ERROR
3664
3665 021242 013737 004146 001160 3$: MOV T.MR2,$TMP0 ;MASK ALL BITSEXCEPT PARITY
3666 021250 042737 077777 001160 BIC #77777,$TMP0
3667 021256 023737 004260 001160 CMP PARBIT,$TMP0 ;CHECK IF PARITY CORRECT
3668 021264 001402 BEQ 4$ ; MESSAGE B
3669 021266 104053 ERROR 53 ;PARITY ON MESSAGE B INCORRECT
3670 021270 000412 BR 25$ ;CHECK IF LOOP ON ERROR
3671
3672 021272 023737 004210 004150 4$: CMP E.MR3,T.MR3 ;CHECK IF MESSAGE A CORRECT
3673 021300 001401 BEQ 5$ ;YES, CHECK IN MESSAGE B CORRECT
3674 021302 104054 ERROR 54 ;MESSAGE A INCORRECT
3675 021304 023737 004206 004146 5$: CMP E.MR2,T.MR2 ;CHECK IF MESSAGE B CORRECT
3676 021312 001401 BEQ 25$ ;YES, CHECK IF LOOP ON ERROR
3677 021314 104055 ERROR 55 ;MESSAGE B INCORRECT
3678 021316 104415 25$: SCOP1 ;CHECK IF LOOP ON ERROR
3679 021320 005737 004244 TST DRVCOD ;CHECK IF DRIVE SELECT = 0 (FINISHED)
3680 021324 001416 BEQ TST41 ;:YES,GO ON TO NEXT TEST
3681 021326 005037 004244 CLR DRVCOD ;SET DRIVE SELECT CODE = 0
3682 021332 012737 000001 004246 MOV #1,MSGCOD ;SET MESSAGE SELECT CODE
3683 021340 012737 000001 004206 MOV #BIT0,E.MR2 ;LOAD EXPECTED MAINT REG 2 (MESS B)
3684 021346 012737 100000 004210 MOV #BIT15,E.MR3 ;LOAD EXPECTED MAINT REG 3 (MESS A)
3685 021354 005037 004260 CLR PARBIT ;LOAD PARITY FOR MESSAGE B
3686 021360 000655 BR 1$ ;TRY SECOND CONFIGURATION

```

```

3687
3688
3689 *****
3690 *TEST 41 EVEN DRIVE MESSAGE PARITY GENERATION
3691 *
3692 * CLEAR RK611 WITH CONTROLLER CLEAR. PUT CONTROLLER
3693 * IN DIAGNOSTIC MODE AND MESSAGE SELECT = 1
3694 * AND BAD PARITY SET. LOAD COMMAND AND STATUS
3695 * REGISTER 2 WITH DRIVE SELECT = 1. LOAD COMMAND
3696 * AND STATUS REGISTER SELECT COMMAND. VERIFY THAT
3697 * EVEN PARITY IS GENERATED. REPEAT FOR MESSAGE SELECT =
3698 * DRIVE SELECT = 2-17.
3699 *****

```

```

3700 021362 000004          TST41: SCOPE
3701 021364 012737 000144 001200  MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
3702 021372 013702 001270          MOV      $BASE,R2      ;;LOAD RK611 BASE
3703 021376 012737 000001 004244  MOV      #1,DRVCOD     ;;LOAD DRIVE CODE
3704 021404 012737 021412 001110  MOV      #1$, $LPERR   ;;LOAD LOOP ON ERROR LOCATION FOR
3705                                     ;; SUBTEST LOOP
3706
3707 021412          1$:
3708 021412 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
3709 021420 013762 004244 000026  MOV      DRVCOD,RKMR1(R2) ;LOAD MESSAGE SELECT CODE
3710 021426 052762 000060 000026  BIS      #DMD!PAT,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
3711                                     ; AND SET BAD PARITY
3712 021434 013762 004244 000010  MOV      DRVCOD,RKCS2(R2) ;LOAD DRIVE SELECT CODE
3713 021442 012762 000001 000000  MOV      #SELDRV,RKCS1(R2) ;ISSUE SELECT DRIVE
3714 021450 012700 000116          MOV      #19.*4+2,R0    ;LOAD DRIVE MESSAGE AND SHIFT
3715 021454 052762 000400 000026  2$:      BIS      #MCLK,RKMR1(R2) ; ALL 16 BITS
3716 021462 042762 000400 000026  BIC      #MCLK,RKMR1(R2)
3717 021470          DEC      R0
3718 021472          BNE     2$
3719 021474 016237 000034 004146  MOV      RKMR2(R2),T.MR2 ;STORE SHIFTED MESSAGE B
3720 021502 016237 000036 004150  MOV      RKMR3(R2),T.MR3 ;STORE SHIFTED MESSAGE A
3721 021510 013701 004244          MOV      DRVCOD,R1      ;DETERMINE PARITY
3722 021514 012703 000004          MOV      #4,R3
3723 021520          CLR     R4
3724 021522          3$:      ROR     R1
3725 021524 103001          BCC     4$
3726 021526          INC     R4
3727 021530          4$:      DEC     R3
3728 021532          BNE     3$
3729 021534 013737 004244 004206  MOV      DRVCOD,E.MR2   ;LOAD EXPECTED SHIFTED REG. B
3730 021542 013737 004244 004210  MOV      DRVCOD,E.MR3   ;LOAD EXPECTED SHIFTED REG. A
3731 021550 005037 004260          CLR     PARBIT
3732 021554 032704 000001          BIT     #BIT0,R4      ;CHECK FOR PARITY ON WORD
3733 021560          BEQ     5$      ;PARITY ALREADY EVEN
3734 021562 012737 100000 004260  MOV      #BIT15,PARBIT ;SET PARITY BIT
3735 021570 052737 100000 004206  BIS      #BIT15,E.MR2
3736 021576 052737 100000 004210  BIS      #BIT15,E.MR3
3737 021604 013737 004150 001160  5$:      MOV      T.MR3,$TMPO   ;MASK ALL BITS EXCEPT PARITY
3738 021612 042737 077777 001160  BIC      #77777,$TMPO
3739 021620 023737 004260 001160  CMP      PARBIT,$TMPO  ;CHECK IF PARITY CORRECT
3740 021626          BEQ     6$      ; ON MESSAGE A
3741 021630 104056          ERROR  56      ;PARITY ON MESSAGE A INCORRECT
3742 021632 000426          BR     25$      ;CHECK IF LOOP ON ERROR
3743
3744 021634 013737 004146 001160  6$:      MOV      T.MR2,$TMPO   ;MASK ALL BITS EXCEPT PARITY
3745 021642 042737 077777 001160  BIC      #77777,$TMPO
3746 021650 023737 004260 001160  CMP      PARBIT,$TMPO  ;CHECK IF PARITY CORRECT
3747 021656          BEQ     7$      ; ON MESSAGE B
3748 021660 104057          ERROR  57      ;PARITY ON MESSAGE B INCORRECT
3749 021662 000412          BR     25$      ;CHECK IF LOOP ON ERROR
3750
3751 021664 023737 004210 004150  7$:      CMP      E.MR3,T.MR3   ;CHECK IF MESSAGE A CORRECT
3752 021672          BEQ     8$      ;YES, CHECK MESSAGE B
3753 021674 104060          ERROR  60      ;MESSAGE A INCORRECT
3754 021676 023737 004206 004146  8$:      CMP      E.MR2,T.MR2   ;CHECK IF MESSAGE B CORRECT
3755 021704 001401          BEQ     25$      ;YES, CHECK IF LOOP ON ERROR
    
```

```

3756 021706 104061          ERROR 61          ;MESSAGE B INCORRECT
3757 021710 104415          25$: SCOPI        ;CHECK IF LOOP ON ERROR
3758 021712 005237 004244  INC      DRVCOD    ;INCREMENT DRIVE SELECT CODE
3759 021716 022737 000017 004244  CMP      #17,DRVCOD ;CHECK IF FINISHED
3760 021724 103232          BHIS     1$        ;NO, TRY NEXT CONFIGURATION
3761
3762
3763          .SBTTL  **CLASS A COMMAND EXECUTION
3764
3765          ;*****
3766          ;*TEST 42      RELEASE COMMAND IN DIAGNOSTIC MODE
3767          ;*
3768          ;*      CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
3769          ;*      PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND
3770          ;*      STATUS REGISTER 2 WITH DRIVE SELECT = 10. LOAD
3771          ;*      COMMAND AND STATUS REGISTER 1 WITH A SELECT.
3772          ;*      CLOCK COMMAND TO COMPLETION. MAKE SURE UNIT
3773          ;*      FIELD ERROR DOES NOT SET (SACK HIGH). REPEAT FOR
3774          ;*      DRIVE SELECT = 11-17.
3775          ;*
3776          ;*****
3777 021726 000004          TST42: SCOPE
3778 021730 012737 000144 001200  MOV      #100, $TIMES ;DO 100. ITERATIONS
3779 021736 013702 001270          MOV      $BASE,R2    ;LOAD RK611 BASE
3780 021742 012737 000010 004244  MOV      #10,DRVCOD  ;INITIALIZE FOR DESELECT OF DRIVE 0
3781 021750 012737 021756 001110  MOV      #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
3782                                     ; SUBTEST LOOP
3783
3784 021756          1$:
3785 021756 012762 000040 000010  MOV      #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
3786 021764 012762 000040 000026  MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
3787 021772 013762 004244 000010  MOV      DRVCOD,RKCS2(R2) ;LOAD DRIVE SELECTION
3788 022000 012762 000001 000000  MOV      #SELDRV,RKCS1(R2) ;ISSUE DESELECT
3789 022006 012700 000120          MOV      #20.*4,R0    ;LOAD COUNT TO COMPLETE COMMAND
3790 022012 012762 000440 000026  2$: MOV      #DMD!MCLK,RKMR1(R2) ;CLOCK THRU COMMAND
3791 022020 012762 000040 000026  MOV      #DMD,RKMR1(R2)
3792 022026 005300          DEC      R0
3793 022030 001370          BNE     2$
3794 022032 016237 000000 004120  MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3795 022040 016237 000010 004130  MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3796 022046 016237 000012 004132  MOV      RKDS(R2),T.DS  ;STORE DRIVE STATUS REGISTER
3797 022054 016237 000014 004134  MOV      RKER(R2),T.ER  ;STORE ERROR REGISTER
3798 022062 012737 000200 004160  MOV      #RDY,E.CS1    ;LOAD EXPECTED COMMAND AND STATUS REG. 1
3799 022070 013737 004244 004170  MOV      DRVCOD,E.CS2  ;GENERATE EXPECTED COMMAND AND
3800 022076 052737 000100 004170  BIS      #IR,E.CS2     ;STATUS REG. 2
3801 022104 005037 004172          CLR     E.DS          ;LOAD EXPECTED DRIVE STATUS REGISTER
3802 022110 005037 004174          CLR     E.ER          ;LOAD EXPECTED ERROR REGISTER
3803 022114 023737 004160 004120  CMP      E.CS1,T.CS1   ;CHECK COMMAND AND STATUS REG 1 CORRECT
3804 022122 001401          BEQ     3$
3805 022124 104062          ERROR 62
3806 022126 023737 004170 004130  3$: CMP      E.CS2,T.CS2   ;CHECK COMMAND AND STATUS REG. 2 CORRECT
3807 022134 001401          BEQ     4$
3808 022136 104063          ERROR 63
3809 022140 023737 004174 004134  4$: CMP      E.ER,T.ER   ;CHECK ERROR REGISTER CORRECT
3810 022146 001401          BEQ     5$
3811 022150 104064          ERROR 64          ;ERROR REGISTER INCORRECT

```

```

3812 022152 023737 004172 004132 5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG CORRECT
3813 022160 001401 BEQ 5$ ;YES, CHECK IF LOOP ON ERROR
3814 022162 104126 ERROR 126 ;DRIVE STATUS REG INCORRECT
3815 022164 104415 6$: SCOPI ;CHECK IF LOOP ON ERROR
3816 022166 005237 004244 INC DRVCOD ;INCREMENT DRIVE NUMBER
3817 022172 022737 000017 004244 CMP #17,DRVCOD ;CHECK IF ALL DRIVE NUMBERS TESTED
3818 022200 103266 BHIS 1$ ;NO, DO IT FOR NEXT DRIVE NUMBER
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832 022202 000004 *****
3833 022204 012737 000144 001200 *TEST 43 SELECT COMMAND IN DIAGNOSTIC MODE
3834 022212 013702 001270 *
3835 022216 005037 004244 * CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
3836 022222 012737 022230 001110 * PUT CONTROLLER IN DIAGNOSTIC MODE. LOAD COMMAND AND
* STATUS REGISTER 2 WITH DRIVE SELECT = 0. LOAD
* COMMAND AND STATUS REGISTER 1 WITH A SELECT.
* CLOCK COMMAND TO COMPLETION. MAKE SURE MESSAGE SHIFT IS
* NOT DONE DURING THE RECEIVE CYCLE OF DRIVE MESSAGE.
* MAKE SURE NO ERRORS SET. REPEAT FOR DRIVE SELECT = 1-7.
3837
3838
3839 022230 1$:
3840 022230 012762 000040 000010 MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
3841 022236 012762 000040 000026 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
3842 022244 013762 004244 000010 MOV DRVCOD,RKCS2(R2) ;LOAD DRIVE SELECT
3843 022252 012762 000001 000000 MOV #SELDRV,RKCS1(R2) ;ISSUE DRIVE SELECT
3844 022260 012700 000120 MOV #20,#4,R0 ;LOAD COUNT TO DESELECT COMPLETE
3845 022264 012762 000440 000026 2$: MOV #DMD!MCLK,RKMR1(R2) ;CLOCK UNTIL DESELECT FINISHED
3846 022272 012762 000040 000026 MOV #DMD,RKMR1(R2)
3847 022300 005300 DEC R0
3848 022302 001370 BNE 2$
3849 022304 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3850 022312 012737 000001 004160 MOV #SELDRV,E.CS1 ;LOAD EXPECTED COMMAND AND STATUS REG. 1
3851 022320 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK IF READY RESET
3852 022326 001402 BEQ 3$ ;YES, CONTINUE COMMAND
3853 022330 104065 ERROR 6$ ;COMMAND AND STATUS REG. 1 INCORRECT
3854 022332 000566 BR 25$ ;GO CHECK IF LOOP ON ERROR
3855
3856 022334 013703 004244 3$: MOV DRVCOD,R3 ;GENERATE EXPECTED MAINT REG 3
3857 022340 012701 000003 MOV #3,R1
3858 022344 005000 CLR R0
3859 022346 006003 4$: ROR R3
3860 022350 103001 BCC 5$
3861 022352 005200 INC R0
3862 022354 005301 5$: DEC R1
3863 022356 001373 BNE 4$
3864 022360 013737 004244 004210 MOV DRVCOD,E.MR3
3865 022366 032700 000001 BIT #BIT0,R0
3866 022372 001003 BNE 6$
3867 022374 052737 100000 004210 BIS #BIT15,E.MR3

```



```

3924 022724 103402          BLO          TST44          ;;YES, GO TO NEXT TEST
3925 022726 000137 022230  JMP          1$          ;;TRY NEXT DRIVE
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937 022732 000004          TST44: SCOPE
3938 022734 012737 000144 001200  MOV          #100, $TIMES      ;;DO 100. ITERATIONS
3939 022742 013702 001270          MOV          $BASE, R2        ;;LOAD RK611 BASE
3940 022746 012737 000010 004244  MOV          #10, DRVCOD      ;;INITIALIZE FOR DESELECT OF DRIVE 0
3941 022754 012737 022762 001110  MOV          #1$, $LPERR      ;;LOAD LOOP ON ERROR LOCATION FOR
3942                                     ;; SUBTEST LOOP
3943
3944 022762          1$:
3945 022762 012762 000040 000010  MOV          #SCLR, RKCS2(R2) ;;CLEAR RK06 SUBSYSTEM
3946 022770 013762 004244 000010  MOV          DRVCOD, RKCS2(R2) ;;LOAD DRIVE SELECTION
3947 022776 012762 000001 000000  MOV          #SELD, RKCS1(R2) ;;ISSUE DESELECT
3948 023004 013700 004262          MOV          WAITIM, R0      ;;WAIT FOR READY
3949 023010 105762 000000          2$: TSTB          RKCS1(R2)
3950 023014 100402          BMI          3$
3951 023016 005300          DEC          R0
3952 023020 001373          BNE          2$
3953 023022 016237 000000 004120  3$: MOV          RKCS1(R2), T.CS1 ;;STORE COMMAND AND STATUS REG. 1
3954 023030 016237 000010 004130  MOV          RKCS2(R2), T.CS2 ;;STORE COMMAND AND STATUS REG. 2
3955 023036 016237 000012 004132  MOV          RKDS(R2), T.DS   ;;STORE DRIVE STATUS REGISTER
3956 023044 016237 000014 004134  MOV          RKER(R2), T.ER   ;;STORE ERROR REG.
3957 023052 012737 000200 004160  MOV          #RDY, E.CS1     ;;LOAD EXPECTED COMMAND AND STATUS REG. 1
3958 023060 013737 004244 004170  MOV          DRVCOD, E.CS2   ;;GENERATE EXPECTED COMMAND AND STATUS REG. 2
3959 023066 052737 000100 004170  BIS          #IR, E.CS2
3960 023074 005037 004172          CLR          E.DS          ;;LOAD EXPECTED DRIVE STATUS REG
3961 023100 005037 004174          CLR          E.ER          ;;LOAD EXPECTED ERROR REG.
3962 023104 023737 004160 004120  CMP          E.CS1, T.CS1    ;;CHECK COMMAND AND STATUS REG 1 CORRECT
3963 023112 001401          BEQ          4$            ;;YES, CHECK CS2
3964 023114 104074          ERROR       74            ;;CS1 INCORRECT
3965 023116 023737 004170 004130  4$: CMP          E.CS2, T.CS2 ;;CHECK COMMAND AND STATUS REG 2 CORRECT
3966 023124 001401          BEQ          5$            ;;YES, CHECK ERROR REGISTER
3967 023126 104075          ERROR       75            ;;CS2 INCORRECT
3968 023130 023737 004174 004134  5$: CMP          E.ER, T.ER   ;;CHECK ERROR REG CORRECT
3969 023136 001401          BEQ          6$            ;;YES, CHECK DRIVE STATUS REG CORRECT
3970 023140 104076          ERROR       76            ;;ERROR REG INCORRECT
3971 023142 023737 004172 004132  6$: CMP          E.DS, T.DS   ;;CHECK DRIVE STATUS REG CORRECT
3972 023150 001401          BEQ          7$            ;;YES, CHECK IF LOOP ON ERROR
3973 023152 104130          ERROR       130          ;;DRIVE STATUS REGISTER INCORRECT
3974 023154 104415          7$: SCOPI
3975 023156 005237 004244          INC          DRVCOD        ;;INCREMENT DRIVE NUMBER
3976 023162 022737 000017 004244  CMP          #17, DRVCOD    ;;CHECK IF ALL DRIVE NUMBERS TESTED
3977 023170 103274          BHIS        1$            ;;NO, DO IT FOR NEXT DRIVE
3978
3979

```

;;\*\*\*\*\*

E08

```

3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994 023172 000004
3995 023174 012737 000144 001200
3996 023202 013702 001270
3997 023206 012762 000040 000010
3998 023214 012762 000010 000010
3999 023222 013701 004234
4000 023226 012721 023310
4001 023232 012711 000340
4002 023236 005046
4003 023240 012746 023246
4004 023244 000002
4005
4006 023246
4007 023246 012762 000101 000000
4008 023254 013700 004262
4009 023260 105762 000000
4010 023264 100402
4011 023266 005300
4012 023270 001373
4013 023272 012746 000340
4014 023276 012746 023304
4015 023302 000002
4016
4017 023304 104100
4018 023306 000513
4019
4020 023310 062706 000004
4021 023314 016237 000000 004120
4022 023322 016237 000010 004130
4023 023330 016237 000014 004134
4024 023336 012737 000300 004160
4025 023344 012737 000110 004170
4026 023352 005037 004174
4027 023356 023737 004160 004120
4028 023364 001401
4029 023366 104101
4030 023370 023737 004170 004130
4031 023376 001401
4032 023400 104102
4033 023402 023737 004174 004134
4034 023410 001401
4035 023412 104103

```

```

*TEST 45 INTERRUPT AT COMMAND COMPLETION
**
** CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
** LOWER PROCESSOR PRIORITY TO ZERO. ISSUE A RELEASE
** COMMAND WITH INTERRUPT ENABLE SET. MAKE SURE
** INTERRUPT OCCURS. LOWER PRIORITY AFTER INTERRUPT
** AND MAKE SURE INTERRUPT HAS CLEARED.
**
** LOWER PROCESSOR PRIORITY TO ZERO. REISSUE RELEASE
** WITH INTERRUPT ENABLE RESET. MAKE SURE NO INTERRUPT
** OCCURS. SET INTERRUPT ENABLE AND MAKE SURE NO
** INTERRUPT OCCURS.
**
*****
TST45: SCOPE
MOV #100, $TIMES ;; DO 100. ITERATIONS
MOV $BASE, R2 ;; LOAD RK611 BASE
MOV #SCLR, RKCS2(R2) ;; CLEAR RK06 SUBSYSTEM
MOV #10, RKCS2(R2) ;; SET DESELECT BIT
MOV RKVEC, R1 ;; LOAD INTERRUPT VECTOR
MOV #5$, (R1)+
MOV #PR7, (R1)
CLR -(SP) ;; LOAD STACK TO ALLOW ALL INTERRUPTS
MOV #64$, -(SP) ;; LOAD NEXT ADDRESS
RTI ;; CLEAR PSW

64$: MOV #SELDRV!IE, RKCS1(R2) ;; ISSUE SELECT DRIVE
MOV WAITIM, R0 ;; WAIT FOR READY
2$: TSTB RKCS1(R2)
BMI 3$
R0
BNE 2$
3$: MOV #PR7, -(SP) ;; LOCK OUT INTERRUPTS
MOV #4$, -(SP)
RTI

4$: ERROR 100 ;; INTERRUPT DID NOT OCCUR
BR 25$

5$: ADD #4, SP ;; ADJUST STACK
MOV RKCS1(R2), T.CS1 ;; STORE COMMAND AND STATUS REG. 1
MOV RKCS2(R2), T.CS2 ;; STORE COMMAND AND STATUS REG. 2
MOV RKER(R2), T.ER ;; STORE ERROR REG.
MOV #RDY!IE, E.CS1 ;; LOAD EXPECTED CS1
MOV #IR!10, E.CS2 ;; LOAD EXPECTED CS2
CLR E.ER ;; LOAD EXPECTED ERROR
CMP E.CS1, T.CS1 ;; CHECK IF CS1 CORRECT
BEQ 6$ ;; YES, CHECK CS2
ERROR 101 ;; CS1 INCORRECT
6$: CMP E.CS2, T.CS2 ;; CHECK IF CS2 INCORRECT
BEQ 7$ ;; YES, CHECK IF ERROR REG CORRECT
ERROR 102 ;; CS2 INCORRECT
7$: CMP E.ER, T.ER ;; CHECK IF ERROR REG CORRECT
BEQ 8$ ;; YES, CHECK IF INTERRUPT CLEARED
ERROR 103 ;; ERROR REG. INCORRECT

```

# F08

```

4036 023414 012777 023510 160612 9$: MOV #10$,ARKVEC ;LOAD VECTOR FOR UNEXPECTED INTERRUPT
4037 023422 005046 CLR -(SP) ;LOAD STACK TO ALLOW ALL INTERRUPTS
4038 023424 012746 023432 MOV #65$,-(SP) ;LOAD NEXT ADDRESS
4039 023430 000002 RTI ;CLEAR PSW
4040
4041 023432 65$:
4042 023432 000240 NOP ;WAIT FOR INTERRUPT
4043 023434 012777 023520 160572 MOV #15$,ARKVEC ;LOAD VECTOR ADDRESS FOR UNEXPECTED INTERRUPT
4044 023442 012762 000010 000010 MOV #10,RKCS2(R2) ;ISSUE DESELECT
4045 023450 012762 000001 000000 MOV #SELD,ARKCS1(R2)
4046 023456 000240 NOP ;WAIT FOR INTERRUPT
4047 023460 012777 023530 160546 MOV #20$,ARKVEC ;LOAD VECTOR ADDRESS FOR UNEXPECTED INTERRUPT
4048 023466 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
4049 023474 000240 NOP ;ALLOW INTERRUPT TO OCCUR
4050 023476 012746 000340 MOV #PR7,-(SP) ;LOCK OUT INTERRUPT
4051 023502 012746 023536 MOV #25$,-(SP) ;RESTORE TRAP CATCHER
4052 023506 000002 RTI
4053
4054 023510 062706 000004 10$: ADD #4,SP ;ADJUST STACK
4055 023514 104104 ERROR 104 ;UNEXPECTED INTERRUPT
4056 023516 000407 BR 25$ ;RESTORE TRAP CATCHER
4057
4058 023520 062706 000004 15$: ADD #4,SP ;ADJUST STACK
4059 023524 104254 ERROR 254 ;UNEXPECTED INTERRUPT ON DESELECT
4060 023526 000403 BR 25$ ;RESTORE TRAP CATCHER
4061
4062 023530 062706 000004 20$: ADD #4,SP ;ADJUST STACK
4063 023534 104255 ERROR 255 ;UNEXPECTED INTERRUPT WHEN SETTING
4064 ; INTERRUPT ENABLE
4065 023536 012762 000040 000010 25$: MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4066 023544 013701 004234 MOV RKVEC,R1 ;RESTORE TRAP CATCHER
4067 023550 010111 MOV R1,(R1)
4068 023552 062721 000002 ADD #2,(R1)+
4069 023556 005011 CLR (R1)
4070
4071 ;*****
4072 ;*TEST 46 GO CLEAR OF SILO
4073 ;*
4074 ;* CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
4075 ;* WRITE ONE WORD INTO THE SILO. ISSUE A RELEASE COMMAND
4076 ;* WITH INTERRUPT ENABLE RESET. WAIT FOR READY.
4077 ;* READ THE DATA BUFFER TO MAKE SURE THE SILO HAS BEEN
4078 ;* CLEARED. (DATA LATE SET AFTER READ OF DATA BUFFER)
4079 ;*
4080 ;*****
4081 023560 000004 TST46: SCOPE
4082 023562 012737 000144 001200 MOV #100,$TIMES ;DO 100. ITERATIONS
4083 023570 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE
4084 023574 012762 000040 000010 MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4085 023602 005062 000024 CLR RKDB(R2) ;LOAD 1 WORD IN SILO
4086 023606 012762 000010 000010 MOV #10,RKCS2(R2) ;LOAD DESELECT DRIVE 0
4087 023614 012762 000001 000000 MOV #SELD,ARKCS1(R2) ;ISSUE DESELECT
4088 023622 013700 004262 MOV WAITIM,R0 ;WAIT FOR READY
4089 023626 105762 000000 2$: TSTB RKCS1(R2)
4090 023632 100402 BMI 3$
4091 023634 005300 DEC R0
    
```

```

4092 023636 001373          BNE      2$
4093 023640 016237 000000 004120 3$:  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4094 023646 016237 000010 004130  MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4095 023654 016237 000012 004132  MOV     RKDS(R2),T.DS   ;STORE DRIVE STATUS REGISTER
4096 023662 016237 000014 004134  MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
4097 023670 012737 000200 004160  MOV     #RDY,E.CS1     ;LOAD EXPECTED CS1
4098 023676 012737 000110 004170  MOV     #IR!10,E.CS2   ;LOAD EXPECTED CS2
4099 023704 005037 004172          CLR     E.DS           ;LOAD EXPECTED DRIVE STATUS REG
4100 023710 005037 004174          CLR     E.ER           ;LOAD EXPECTED ERROR REGISTER
4101 023714 023737 004170 004130  CMP     E.CS2,T.CS2   ;CHECK IF CS1 CORRECT
4102 023722 001401          BEQ     10$           ;YES, READ WORD FROM SILO
4103 023724 104105          ERROR  10$           ;CS2 INCORRECT
4104 023726 005762 000024          TST     RKDB(R2)      ;READ SILO TO MAKE IT IS CLEAR
4105 023732 016237 000000 004120 10$:  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4106 023740 016237 000010 004130  MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4107 023746 016237 000014 004134  MOV     RKER(R2),T.ER   ;STORE ERROR REG.
4108 023754 012737 100200 004160  MOV     #CERR!RDY,E.CS1 ;LOAD EXPECTED CS1
4109 023762 012737 100110 004170  MOV     #DLT!IR!10,E.CS2 ;LOAD EXPECTED CS2
4110 023770 023737 004170 004130  CMP     E.CS2,T.CS2   ;CHECK IF DATA LATE SET
4111 023776 001401          BEQ     11$           ;YES, CLEAR CONTROLLER REG. 1
4112 024000 104106          ERROR  106           ;DATA LATE NOT SET
4113 024002 012762 100000 000000 11$:  MOV     #CLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4114
4115 ;*****
4116 ;*TEST 47          SEEK COMMAND IN DIAGNOSTIC MODE
4117 ;*
4118 ;*          CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
4119 ;*          PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK WITH CDT SET
4120 ;*          24 SECTOR FORMAT TO CYLINDER 1714, HEAD 7, DRIVE 0.
4121 ;*          MAKE SURE NO STATUS BITS ARE SET AND NO ERROR
4122 ;*          BITS ARE SET.
4123 ;*
4124 ;*****
4125 024010 000004          TST47: SCOPE
4126 024012 012737 000144 001200  MOV     #100,$TIMES   ;DO 100. ITERATIONS
4127 024020 013702 001270          MOV     $BASE,R2     ;LOAD RK611 BASE
4128 024024 012762 000040 000010  MOV     #CLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4129 024032 012762 000040 000026  MOV     #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
4130 024040 012762 001714 000020  MOV     #1714,RKDCYL(R2) ;LOAD CYLINDER ADDRESS
4131 024046 012762 003400 000006  MOV     #3400,RKDA(R2) ;LOAD HEAD 7
4132 024054 012762 012017 000000  MOV     #SEEK!CFMT!CDT,RKCS1(R2) ;ISSUE SEEK CDT SET,24 SECTOR
4133 024062 012700 000120          MOV     #20,*4,R0    ;LOAD COUNT TO DESELECT DECISION
4134 024066 012762 000440 000026 2$:  MOV     #DMD!MCLK,RKMR1(R2)
4135 024074 012762 000040 000026  MOV     #DMD,RKMR1(R2)
4136 024102 005300          DEC     R0
4137 024104 001370          BNE     2$
4138 024106 016237 000000 004120  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4139 024114 012737 012017 004160  MOV     #SEEK!CFMT!CDT,E.CS1 ;LOAD EXPECTED COMMAND AND STATUS REG 1
4140 024122 023737 004160 004120  CMP     E.CS1,T.CS1   ;CHECK IF READY RESET
4141 024130 001402          BEQ     3$           ;YES, CONTINUE COMMAND
4142 024132 104107          ERROR  107
4143 024134 000543          BR     TST50        ;GO ON TO NEXT TEST
4144
4145 024136 012737 071020 004210 3$:  MOV     #S.SEEK!S.FMT!70000,E.MR3 ;LOAD EXPECTED MAINT REG. 3
4146 024144 012737 136300 004206  MOV     #136300,E.MR2 ;LOAD EXPECTED MAINT REG. 2
4147 024152 012701 000003          MOV     #3,R1        ;ISSUE 3 CONTROL CLOCKS

```

```

4148 024156 012700 000004      4$:  MOV      #4,RO
4149 024162 012762 000440 000026  5$:  MOV      #DMD!MCLK,RKMR1(R2)
4150 024170 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4151 024176 005300      DEC      RO
4152 024200 001370      BNE      5$
4153 024202 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4154 024210 016237 000034 004146      MOV      RKMR2(R2),T.MR2 ;STORE MAINT REG 2
4155 024216 016237 000036 004150      MOV      RKMR3(R2),T.MR3 ;STORE MAINT REG 3
4156 024224 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG. 1 CORRECT
4157 024232 001402      BEQ      6$ ;YES, CHECK MAINTENANCE REG. 2
4158 024234 104110      ERROR   110 ;CS1 INCORRECT
4159 024236 000502      BR       TST50 ;GO TO NEXT TEST
4160
4161 024240 023737 004206 004146  6$:  CMP      E.MR2,T.MR2 ;CHECK MAINT REG 2 CORRECT
4162 024246 001402      BEQ      7$ ;YES, CHECK MAINTENANCE REG 3
4163 024250 104111      ERROR   111 ;MAINT REG 2 INCORRECT
4164 024252 000474      BR       TST50 ;GO TO NEXT TEST
4165
4166 024254 023737 004210 004150  7$:  CMP      E.MR3,T.MR3 ;CHECK IF MAINT REG 3 CORRECT
4167 024262 001402      BEQ      8$ ;YES, CHECK COMMAND COMPLETE
4168 024264 104112      ERROR   112 ;MR3 INCORRECT
4169 024266 000466      BR       TST50 ;GO TO NEXT TEST
4170
4171 024270 005301      8$:  DEC      R1 ;CHECK IF COMMAND FINISHED
4172 024272 001331      BNE      4$ ;NO, ISSUE ANOTHER CONTROL CLOCK
4173
4174 024274 012700 000004      MOV      #4,RO ;ISSUE LAST CONTROL CLOCK FOR READY
4175 024300 012762 000440 000026  9$:  MOV      #DMD!MCLK,RKMR1(R2)
4176 024306 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4177 024314 005300      DEC      RO
4178 024316 001370      BNE      9$
4179 024320 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4180 024326 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4181 024334 016237 000012 004132      MOV      RKDS(R2),T.DS ;STORE DRIVE STATUS REGISTER
4182 024342 016237 000014 004134      MOV      RKER(R2),T.ER ;STORE ERROR REGISTER
4183 024350 012737 012216 004160      MOV      #RDY!CFMT!CDT!<SEEK&C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4184 024356 012737 000100 004170      MOV      #IR,E.CS2 ;LOAD EXPECTED CS2
4185 024364 005037 004172      CLR      E.DS ;LOAD EXPECTED DRIVE STATUS REGISTER
4186 024370 005037 004174      CLR      E.ER ;LOAD EXPECTED ERROR REGISTER
4187 024374 023737 004160 004120      CMP      E.CS1,T.CS1 ;CHECK IF COMMAND AND STATUS REG. 2
4188 024402 001401      BEQ      10$ ;YES, CHECK CS2
4189 024404 104113      ERROR   113 ;CS1 INCORRECT
4190 024406 023737 004170 004130 10$:  CMP      E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4191 024414 001401      BEQ      11$ ;YES, CHECK ERROR REG
4192 024416 104114      ERROR   114 ;CS2 INCORRECT
4193 024420 023737 004174 004134 11$:  CMP      E.ER,T.ER ;CHECK ERROR REGISTER
4194 024426 001401      BEQ      12$ ;YES, CHECK DRIVE STATUS REG
4195 024430 104115      ERROR   115 ;ERROR REG. INCORRECT
4196 024432 023737 004172 004132 12$:  CMP      E.DS,T.DS ;CHECK DRIVE STATUS REGISTER CORRECT
4197 024440 001401      BEQ      TST50 ;YES, GO ON TO NEXT TEST
4198 024442 104131      ERROR   131 ;DRIVE STATUS REGISTER INCORRECT
4199

```

.SBTTL \*\*ERROR AND STATUS BIT FORCING WITH DRIVE MESSAGES

```

;*****
;*TEST 50 DRIVE STATUS FROM SHIFT REGISTER

```

4200  
4201  
4202  
4203

```

4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214 024444 000004
4215 024446 012737 000144 001200
4216 024454 013702 001270
4217 024460 012762 000040 000010
4218 024466 012762 000040 000026
4219 024474 012762 000757 000020
4220 024502 012762 000400 000006
4221 024510 012762 000017 000000
4222 024516 012700 000132
4223 024522 012762 000440 000026 1$:
4224 024530 012762 000040 000026
4225 024536 005300
4226 024540 001370
4227 024542 005062 000026
4228 024546 013700 004262
4229 024552 105762 000000 2$:
4230 024556 100412
4231 024560 005300
4232 024562 001373
4233 024564 016237 000000 004120
4234 024572 012737 000216 004160
4235 024600 104132
4236 024602 000460
4237
4238 024604 016237 000000 004120 3$:
4239 024612 016237 000010 004130
4240 024620 016237 000012 004132
4241 024626 016237 000014 004134
4242 024634 012737 000216 004160
4243 024642 012737 000100 004170
4244 024650 012737 104325 004172
4245 024656 005037 004174
4246 024662 032737 004325 004132
4247 024670 001001
4248 024672 104133
4249 024674 023737 004160 004120 4$:
4250 024702 001401
4251 024704 104134
4252 024706 023737 004170 004130 5$:
4253 024714 001401
4254 024716 104135
4255 024720 023737 004174 004134 6$:
4256 024726 001401
4257 024730 104136
4258 024732 023737 004172 004132 7$:
4259 024740 001401

```

```

: *
: * CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
: * RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
: * TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 757, HEAD 1,
: * DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS
: * 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE SPEED LOSS,
: * DRIVE AVAILABLE, VOLUME VALID, OFFSET, DRIVE READY,
: * AND WRITE LOCK ARE SET.
: *
: *****
: ST50: SCOPE
: MOV #100, $TIMES ; DO 100. ITERATIONS
: MOV $BASE, R2 ; LOAD RK611 BASE
: MOV #SCLR, RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
: MOV #DMD, RKMR1(R2) ; PUT RK611 IN MAINT MODE
: MOV #757, RKDCYL(R2) ; LOAD CYLINDER ADDRESS
: MOV #400, RKDA(R2) ; LOAD HEAD ADD =1
: MOV #SEEK, RKCS1(R2) ; ISSUE SEEK
: MOV #22.*4+2, R0 ; ISSUE CLOCKS UNTIL PHASE ADDRESS 6
1$: MOV #DMD!MCLK, RKMR1(R2)
: MOV #DMD, RKMR1(R2)
: DEC R0
: BNE 1$
: CLR RKMR1(R2) ; FINISH COMMAND IN NORMAL MODE
: MOV WAITIM, R0 ; WAIT FOR FOR READY
2$: TSTB RKCS1(R2)
: BMI 3$
: DEC R0
: BNE 2$
: MOV RKCS1(R2), T.CS1 ; STORE COMMAND AND STATUS REG 1
: MOV #RDY!SEEK<↑C<GO>>, E.CS1 ; LOAD EXPECTED CS1
: ERROR 132 ; READY NOT SET
: BR 10$ ; CLEAR RK06 SUBSYSTEM
3$: MOV RKCS1(R2), T.CS1 ; STORE COMMAND AND STATUS REG 1
: MOV RKCS2(R2), T.CS2 ; STORE COMMAND AND STATUS REG 2
: MOV RKDS(R2), T.DS ; STORE DRIVE STATUS REF
: MOV RKER(R2), T.ER ; STORE ERROR REG.
: MOV #RDY!SEEK<↑C<GO>>, E.CS1 ; LOAD EXPECT CS1
: MOV #IR, E.CS2 ; LOAD EXPECTED CS2
: MOV #DRA!OFST!SPDLSS!VV!DRDY!WRL!SVAL, E.DS ; LOAD EXPECTED DRIVE STATUS
: CLR E.ER ; LOAD EXPECTED ERROR REGISTER
: BIT #DRA!OFST!SPDLSS!VV!DRDY!WRL, T.DS ; CHECK LOAD STATUS SET
: BNE 4$ ; YES, CHECK REGISTERS
: ERROR 133 ; NO BIT IN DRIVE STATUS SET
4$: CMP E.CS1, T.CS1 ; CHECK CS1 CORRECT
: BEQ 5$ ; YES, CONTINUE
: ERROR 134 ; CS1 INCORRECT
5$: CMP E.CS2, T.CS2 ; CHECK CS2 CORRECT
: BEQ 6$ ; YES, CONTINUE
: ERROR 135 ; CS2 INCORRECT
6$: CMP E.ER, T.ER ; CHECK ERROR REG CORRECT
: BEQ 7$ ; YES CONTINUE
: ERROR 136 ; ERROR REG INCORRECT
7$: CMP E.DS, T.DS ; CHECK DRIVE STATUS CORRECT
: BEQ 10$ ; CLEAR RK611

```

```

4260 024742 104137          ERROR 137          ;DRIVE STATUS INCORRECT
4261 024744 013737 004120 004220 10$: MOV T.CS1,P.CS1 ;STORE PREVIOUS CONTENTS OF
4262 024752 013737 004130 004222      MOV T.CS2,P.CS2 ;COMMAND AND STATUS REG 1
4263 024760 013737 004132 004224      MOV T.DS,P.DS   ;COMMAND AND STATUS REG 2
4264 024766 013737 004134 004226      MOV T.ER,P.ER   ;DRIVE STATUS REG
4265                                     ;AND ERROR REG
4266 024774 012762 100000 000000      MOV #CCLR,RKCS1(R2) ;CLEAR RK611
4267 025002 016237 000000 004120      MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4268 025010 016237 000010 004130      MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4269 025016 016237 000012 004132      MOV RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
4270 025024 016237 000014 004134      MOV RKER(R2),T.ER   ;STORE ERROR REG
4271 025032 012737 000200 004160      MOV #RDY,E.CS1     ;LOAD EXPECTED CS1
4272 025040 012737 000100 004170      MOV #IR,E.CS2      ;LOAD EXPECTED CS2
4273 025046 005037 004172              CLR E.DS           ;LOAD EXPECTED DRIVE STATUS REG
4274 025052 005037 004174              CLR E.ER           ;LOAD EXPECTED ERROR REG
4275 025056 023737 004160 004120      CMP E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG 1 CORRECT
4276 025064 001401              BEQ 11$           ;YES, CHECK CS2
4277 025066 104224          ERROR 224          ;CS1 INCORRECT
4278 025070 023737 004170 004130 11$: CMP E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG 2 CORRECT
4279 025076 001401              BEQ 12$           ;YES, CHECK DRIVE STATUS REG
4280 025100 104225          ERROR 225          ;CS2 INCORRECT
4281 025102 023737 004172 004132 12$: CMP E.DS,T.DS     ;CHECK IF DRIVE STATUS REG CORRECT
4282 025110 001401              BEQ 13$           ;YES, CHECK ERROR REG
4283 025112 104226          ERROR 226          ;ERROR REG INCORRECT
4284 025114 023737 004174 004134 13$: CMP E.ER,T.ER     ;CHECK IF ERROR REG CORRECT
4285 025122 001401              BEQ TST51         ;YES, GO ON TO NEXT TEST
4286 025124 104227          ERROR 227          ;ERROR REG INCORRECT

```

```

*****
*TEST 51 DRIVE AVAILABLE SETTING

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO A RK06,
* 26 SECTOR FORMAT TO CYLINDER 2, HEAD 0, DRIVE 0.
* CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
* TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE
* AVAILIABLE SETS.

```

```

*****
TST51: SCOPE

```

```

4299 025126 000004          TST51: SCOPE
4300 025130 012737 000144 001200      MOV #100,$TIMES    ;DO 100. ITERATIONS
4301 025136 013702 001270              MOV $BASE,R2       ;LOAD RK611 BASE
4302 025142 012762 000040 000010      MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4303 025150 012762 000040 000026      MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
4304 025156 012762 000002 000020      MOV #2,RKDCYL(R2)  ;LOAD CYLINDER AND
4305 025164 012762 000000 000006      MOV #0,RKDA(R2)    ;LOAD HEAD ADDRESS
4306 025172 012762 000017 000000      MOV #SEEK,RKCS1(R2) ;ISSUE SEEK
4307 025200 012700 000132              MOV #22,*4+2,RO    ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4308 025204 012762 000440 000026 1$: MOV #DMD!MCLK,RKMR1(R2)
4309 025212 012762 000040 000026      MOV #DMD,RKMR1(R2)
4310 025220 005300          DEC RO
4311 025222 001370          BNE 1$
4312 025224 005062 000026          CLR RKMR1(R2)      ;FINISH COMMAND IN NORMAL MODE
4313 025230 013700 004262          MOV WAITIM,RO      ;WAIT FOR READY
4314 025234 105762 000000 2$: TSTB RKCS1(R2)
4315 025240 100402          BMI 3$

```

K08

```

4316 025242 005300          DEC      R0
4317 025244 001373          BNE     2$
4318 025246 016237 000000 004120 3$:  MOV    RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4319 025254 016237 000010 004130      MOV    RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4320 025262 016237 000012 004132      MOV    RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
4321 025270 016237 000014 004134      MOV    RKER(R2),T.ER   ;STORE ERROR REG
4322 025276 012737 000216 004160      MOV    #RDY!SEEK<↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4323 025304 012737 000100 004170      MOV    #IR,E.CS2      ;LOAD EXPECTED CS2
4324 025312 012737 100001 004172      MOV    #SVAl!DRA,E.DS ;LOAD EXPECTED DRIVE STATUS REG
4325 025320 012737 000000 004174      MOV    #D,E.ER        ;LOAD EXPECTED ERROR REG
4326 025326 023737 004160 004120      CMP    E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG.1 CORRECT
4327 025334 001401          BEQ    4$             ;YES, CONTINUE
4328 025336 104140          ERROR  140
4329 025340 023737 004170 004130 4$:  CMP    E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4330 025346 001401          BEQ    5$             ;YES, CONTINUE
4331 025350 104141          ERROR  141
4332 025352 023737 004172 004132 5$:  CMP    E.DS,T.DS     ;CHECK DRIVE STATUS REG. CORRECT
4333 025360 001401          BEQ    6$             ;YES, CONTINUE
4334 025362 104142          ERROR  142
4335 025364 023737 004174 004134 6$:  CMP    E.ER,T.ER     ;CHECK ERROR REGISTER CORRECT
4336 025372 001401          BEQ    7$             ;YES, CLEAR RK611
4337 025374 104143          ERROR  143
4338 025376 013737 004120 004220 7$:  MOV    T.CS1,P.CS1    ;STORE PREVIOUS CONTENTS OF
4339 025404 013737 004130 004222      MOV    T.CS2,P.CS2    ;COMMAND AND STATUS REG 1
4340 025412 013737 004132 004224      MOV    T.DS,P.DS     ;COMMAND AND STATUS REG 2
4341 025420 013737 004134 004226      MOV    T.ER,P.ER     ;DRIVE STATUS REG
4342          ;AND ERROR REG
4343 025426 012762 100000 000000      MOV    #CLR,RKCS1(R2) ;CLEAR RK611
4344 025434 016237 000000 004120      MOV    RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4345 025442 016237 000010 004130      MOV    RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4346 025450 016237 000012 004132      MOV    RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
4347 025456 016237 000014 004134      MOV    RKER(R2),T.ER   ;STORE ERROR REG
4348 025464 012737 000200 004160      MOV    #RDY,E.CS1     ;LOAD EXPECTED CS1
4349 025472 012737 000100 004170      MOV    #IR,E.CS2      ;LOAD EXPECTED CS2
4350 025500 005037 004172          CLR    E.DS           ;LOAD EXPECTED DRIVE STATUS REG
4351 025504 005037 004174          CLR    E.ER           ;LOAD EXPECTED ERROR REG
4352 025510 023737 004160 004120      CMP    E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG 1 CORRECT
4353 025516 001401          BEQ    11$           ;YES, CHECK CS2
4354 025520 104224          ERROR  224           ;CS1 INCORRECT
4355 025522 023737 004170 004130 11$:  CMP    E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG 2 CORRECT
4356 025530 001401          BEQ    12$           ;YES, CHECK DRIVE STATUS REG
4357 025532 104225          ERROR  225           ;CS2 INCORRECT
4358 025534 023737 004172 004132 12$:  CMP    E.DS,T.DS     ;CHECK IF DRIVE STATUS REG CORRECT
4359 025542 001401          BEQ    13$           ;YES, CHECK ERROR REG
4360 025544 104226          ERROR  226           ;ERROR REG INCORRECT
4361 025546 023737 004174 004134 13$:  CMP    E.ER,T.ER     ;CHECK IF ERROR REG CORRECT
4362 025554 001401          BEQ    TST52         ;YES, GO ON TO NEXT TEST
4363 025556 104227          ERROR  227           ;ERROR REG INCORRECT

```

```

*****
*TEST 52      DRIVE BUS PARITY ERROR
*
*      CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
*      PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
*      TO A RK06, 26 SECTOR FORMAT TO CYLINDER 3, HEAD 0.
*      DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
*

```

```

4364
4365
4366
4367
4368
4369
4370
4371

```

```

4372          ;*      TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE BUS
4373          ;*      PARITY, DRIVE AVAILIABLE, AND CONTROLLER ERROR ARE SET.
4374          ;*
4375          ;*****
4376 025560 000004          TST52: SCOPE
4377 025562 012737 000144 001200      MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
4378 025570 013702 001270          MOV      $BASE,R2        ;;LOAD RK611 BASE
4379 025574 012762 000040 000010      MOV      #SCLR,RKCS2(R2) ;;CLEAR RK06 SUBSYSTEM
4380 025602 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;;PUT RK611 IN MAINT MODE
4381 025610 012762 000003 000020      MOV      #3,RKDCYL(R2)  ;;LOAD CYLINDER AND
4382 025616 012762 000000 000006      MOV      #0,RKDA(R2)    ;;LOAD HEAD ADDRESS
4383 025624 012762 000017 000000      MOV      #SEEK,RKCS1(R2) ;;ISSUE SEEK
4384 025632 012700 000132          MOV      #22.*4+2,R0    ;;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4385 025636 012762 000440 000026 1$:  MOV      #DMD!MCLK,RKMR1(R2)
4386 025644 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4387 025652 005300          DEC      R0
4388 025654 001370          BNE     1$
4389 025656 005062 000026          CLR     RKMR1(R2)      ;;FINISH COMMAND IN NORMAL MODE
4390 025662 013700 004262          MOV     WAITIM,R0     ;;WAIT FOR READY
4391 025666 105762 000000          2$:  TSTB   RKCS1(R2)
4392 025672 100402          BMI     3$
4393 025674 005300          DEC     R0
4394 025676 001373          BNE     2$
4395 025700 016237 000000 004120 3$:  MOV     RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG 1
4396 025706 016237 000010 004130      MOV     RKCS2(R2),T.CS2 ;;STORE COMMAND AND STATUS REG 2
4397 025714 016237 000012 004132      MOV     RKDS(R2),T.DS  ;;STORE DRIVE STATUS REG
4398 025722 016237 000014 004134      MOV     RKER(R2),T.ER  ;;STORE ERROR REG
4399 025730 012737 120216 004160      MOV     #CERR!SPAR!RDY!SEEK<↑C<GO>>,E.CS1 ;;LOAD EXPECTED CS1
4400 025736 012737 000100 004170      MOV     #IR.E.CS2      ;;LOAD EXPECTED CS2
4401 025744 012737 100001 004172      MOV     #SVAL!DRA,E.DS ;;LOAD EXPECTED DRIVE STATUS REG
4402 025752 012737 000000 004174      MOV     #0,E.ER       ;;LOAD EXPECTED ERROR REG
4403 025760 023737 004160 004120      CMP     E.CS1,T.CS1   ;;CHECK COMMAND AND STATUS REG.1 CORRECT
4404 025766 001401          BEQ     4$           ;;YES, CONTINUE
4405 025770 104144          ERROR  144
4406 025772 023737 004170 004130 4$:  CMP     E.CS2,T.CS2   ;;CHECK COMMAND AND STATUS REG. 2 CORRECT
4407 026000 001401          BEQ     5$           ;;YES, CONTINUE
4408 026002 104145          ERROR  145
4409 026004 023737 004172 004132 5$:  CMP     E.DS,T.DS    ;;CHECK DRIVE STATUS REG. CORRECT
4410 026012 001401          BEQ     6$           ;;YES, CONTINUE
4411 026014 104146          ERROR  146
4412 026016 023737 004174 004134 6$:  CMP     E.ER,T.ER    ;;CHECK ERROR REGISTER CORRECT
4413 026024 001401          BEQ     7$           ;;YES, CLEAR RK611
4414 026026 104147          ERROR  147
4415 026030 013737 004120 004220 7$:  MOV     T.CS1,P.CS1   ;;STORE PREVIOUS CONTENTS OF
4416 026036 013737 004130 004222      MOV     T.CS2,P.CS2   ;;COMMAND AND STATUS REG 1
4417 026044 013737 004132 004224      MOV     T.DS,P.DS     ;;COMMAND AND STATUS REG 2
4418 026052 013737 004134 004226      MOV     T.ER,P.ER     ;;DRIVE STATUS REG
4419          ; AND ERROR REG
4420 026060 012762 100000 000000      MOV     #CCLR,RKCS1(R2) ;;CLEAR RK611
4421 026066 016237 000000 004120      MOV     RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG 1
4422 026074 016237 000010 004130      MOV     RKCS2(R2),T.CS2 ;;STORE COMMAND AND STATUS REG 2
4423 026102 016237 000012 004132      MOV     RKDS(R2),T.DS  ;;STORE DRIVE STATUS REG
4424 026110 016237 000014 004134      MOV     RKER(R2),T.ER  ;;STORE ERROR REG
4425 026116 012737 000200 004160      MOV     #RDY,E.CS1    ;;LOAD EXPECTED CS1
4426 026124 012737 000100 004170      MOV     #IR,E.CS2     ;;LOAD EXPECTED CS2
4427 026132 005037 004172          CLR     E.DS         ;;LOAD EXPECTED DRIVE STATUS REG
    
```

```

4428 026136 005037 004174          CLR      E.ER          ;LOAD EXPECTED ERROR REG
4429 026142 023737 004160 004120      CMP      E.CS1,T.CS1  ;CHECK COMMAND AND STATUS REG 1 CORRECT
4430 026150 001401          BEQ      11$          ;YES, CHECK CS2
4431 026152 104224          ERROR   224          ;CS1 INCORRECT
4432 026154 023737 004170 004130 11$:      CMP      E.CS2,T.CS2  ;CHECK COMMAND AND STATUS REG 2 CORRECT
4433 026162 001401          BEQ      12$          ;YES, CHECK DRIVE STATUS REG
4434 026164 104225          ERROR   225          ;CS2 INCORRECT
4435 026166 023737 004172 004132 12$:      CMP      E.DS,T.DS    ;CHECK IF DRIVE STATUS REG CORRECT
4436 026174 001401          BEQ      13$          ;YES, CHECK ERROR REG
4437 026176 104226          ERROR   226          ;ERROR REG INCORRECT
4438 026200 023737 004174 004134 13$:      CMP      E.ER,T.ER    ;CHECK IF ERROR REG CORRECT
4439 026206 001401          BEQ      TST53        ;YES, GO ON TO NEXT TEST
4440 026210 104227          ERROR   227          ;ERROR REG INCORRECT
    
```

```

4441
4442
4443  ;*****
4444  ;TEST 53          DRIVE AVAILABLE RESET ERROR
4445  ;
4446  ;          CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
4447  ;          PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SELECT
4448  ;          TO A RK06, 26 SECTOR FORMAT, AND DRIVE 0.
4449  ;          CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
4450  ;          TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILIABLE
4451  ;          IS RESET AND CONTROLLER ERROR IS SET.
4452  ;*****
    
```

```

4453 026212 000004          TST53: SCOPE
4454 026214 012737 000144 001200      MOV      #100,$TIMES  ;DO 100. ITERATIONS
4455 026222 013702 001270          MOV      $BASE,R2    ;LOAD RK611 BASE
4456 026226 012762 000040 000010      MOV      #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4457 026234 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
4458 026242 012762 000001 000000      MOV      #SELDRV,RKCS1(R2) ;ISSUE SELDRV
4459 026250 012700 000132          MOV      #22.*4+2,RO  ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4460 026254 012762 000440 000026 1$:      MOV      #DMD!MCLK,RKMR1(R2)
4461 026262 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4462 026270 005300          DEC      RO
4463 026272 001370          BNE     1$
4464 026274 005062 000026          CLR      RKMR1(R2)    ;FINISH COMMAND IN NORMAL MODE
4465 026300 013700 004262          MOV      WAITIM,RO   ;WAIT FOR READY
4466 026304 105762 000000          2$:      TSTB     RKCS1(R2)
4467 026310 100402          BMI     3$
4468 026312 005300          DEC      RO
4469 026314 001373          BNE     2$
4470 026316 016237 000000 004120 3$:      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4471 026324 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4472 026332 016237 000012 004132      MOV      RKDS(R2),T.DS  ;STORE DRIVE STATUS REG
4473 026340 016237 000014 004134      MOV      RKER(R2),T.ER  ;STORE ERROR REG
4474 026346 012737 100200 004160      MOV      #CERR!RDY!SELDRV<↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4475 026354 012737 000100 004170      MOV      #IR,E.CS2    ;LOAD EXPECTED CS2
4476 026362 012737 100000 004172      MOV      #SVAl!D,E.DS  ;LOAD EXPECTED DRIVE STATUS REG
4477 026370 012737 000000 004174      MOV      #0,E.ER      ;LOAD EXPECTED ERROR REG
4478 026376 023737 004160 004120      CMP      E.CS1,T.CS1  ;CHECK COMMAND AND STATUS REG.1 CORRECT
4479 026404 001401          BEQ      4$          ;YES, CONTINUE
4480 026406 104150          ERROR   150
4481 026410 023737 004170 004130 4$:      CMP      E.CS2,T.CS2  ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4482 026416 001401          BEQ      5$          ;YES, CONTINUE
4483 026420 104151          ERROR   151
    
```

DZR6BB.P11

T53

DRIVE AVAILABLE RESET ERROR

```

4484 026422 023737 004172 004132 5$:  CMP      E.DS,T.DS      ;CHECK DRIVE STATUS REG. CORRECT
4485 026430 001401          BEQ      6$      ;YES, CONTINUE
4486 026432 104152          ERROR    152
4487 026434 023737 004174 004134 6$:  CMP      E.ER,T.ER      ;CHECK ERROR REGISTER CORRECT
4488 026442 001401          BEQ      7$      ;YES, CLEAR RK611
4489 026444 104153          ERROR    153
4490 026446 013737 004120 004220 7$:  MOV      T.CS1,P.CS1  ;STORE PREVIOUS CONTENTS OF
4491 026454 013737 004130 004222          MOV      T.CS2,P.CS2  ;COMMAND AND STATUS REG 1
4492 026462 013737 004132 004224          MOV      T.DS,P.DS    ;COMMAND AND STATUS REG 2
4493 026470 013737 004134 004226          MOV      T.ER,P.ER    ;DRIVE STATUS REG
4494          ;AND ERROR REG
4495 026476 012762 100000 000000          MOV      #CCLR,RKCS1(R2) ;CLEAR RK611
4496 026504 016237 000000 004120          MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4497 026512 016237 000010 004130          MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4498 026520 016237 000012 004132          MOV      RKDS(R2),T.DS  ;STORE DRIVE STATUS REG
4499 026526 016237 000014 004134          MOV      RKER(R2),T.ER  ;STORE ERROR REG
4500 026534 012737 000200 004160          MOV      #RDY,E.CS1    ;LOAD EXPECTED CS1
4501 026542 012737 000100 004170          MOV      #IR,E.CS2    ;LOAD EXPECTED CS2
4502 026550 005037 004172          CLR      E.DS         ;LOAD EXPECTED DRIVE STATUS REG
4503 026554 005037 004174          CLR      E.ER         ;LOAD EXPECTED ERROR REG
4504 026560 023737 004160 004120          CMP      E.CS1,T.CS1  ;CHECK COMMAND AND STATUS REG 1 CORRECT
4505 026566 001401          BEQ      11$        ;YES, CHECK CS2
4506 026570 104224          ERROR    224        ;CS1 INCORRECT
4507 026572 023737 004170 004130 11$:  CMP      E.CS2,T.CS2  ;CHECK COMMAND AND STATUS REG 2 CORRECT
4508 026600 001401          BEQ      12$        ;YES, CHECK DRIVE STATUS REG
4509 026602 104225          ERROR    225        ;CS2 INCORRECT
4510 026604 023737 004172 004132 12$:  CMP      E.DS,T.DS    ;CHECK IF DRIVE STATUS REG CORRECT
4511 026612 001401          BEQ      13$        ;YES, CHECK ERROR REG
4512 026614 104226          ERROR    226        ;ERROR REG INCORRECT
4513 026616 023737 004174 004134 13$:  CMP      E.ER,T.ER    ;CHECK IF ERROR REG CORRECT
4514 026624 001401          BEQ      TST54      ;YES, GO ON TO NEXT TEST
4515 026626 104227          ERROR    227        ;ERROR REG INCORRECT

```

```

*****
*TEST 54      CDT SET DRIVE TYPE
*****

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
* WITH CDT SET, 26 SECTOR FORMAT, TO CYLINDER 23,
* HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE
* UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE
* AND MAKE SURE ONLY DRIVE AVAILIABLE SETS.
*

```

```

*****
TST54: SCOPE
*****

```

```

4528 026630 000004          ;DO 100. ITERATIONS
4529 026632 012737 000144 001200          MOV      #100,$TIMES  ;LOAD RK611 BASE
4530 026640 013702 001270          MOV      $BASE,R2    ;CLEAR RK06 SUBSYSTEM
4531 026644 012762 000040 000010          MOV      #SCLR,RKCS2(R2) ;PUT RK611 IN MAINT MODE
4532 026652 012762 000040 000026          MOV      #DMD,RKMR1(R2) ;LOAD CYLINDER AND
4533 026660 012762 000023 000020          MOV      #23,RKDCYL(R2) ;LOAD HEAD ADDRESS
4534 026666 012762 000000 000006          MOV      #0,RKDA(R2)   ;ISSUE CDT!SEEK
4535 026674 012762 002017 000000          MOV      #CDT!SEEK,RKCS1(R2) ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4536 026702 012700 000132          MOV      #22,*4+2,RO
4537 026706 012762 000440 000026 1$:  MOV      #DMD!MCLK,RKMR1(R2)
4538 026714 012762 000040 000026          MOV      #DMD,RKMR1(R2)
4539 026722 005300          DEC      RO

```

```

4540 026724 001370 BNE 1$
4541 026726 005062 CLR RKMR1(R2) ;FINISH COMMAND IN NORMAL MODE
4542 026732 013700 MOV WAITIM,RO ;WAIT FOR READY
4543 026736 105762 2$: TSTB RKCS1(R2)
4544 026742 100402 BMI 3$
4545 026744 005300 RO
4546 026746 001373 BNE 2$
4547 026750 016237 000000 004120 3$: MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4548 026756 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4549 026764 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
4550 026772 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
4551 027000 012737 002216 004160 MOV #CDT!RDY!CDT!SEEK<↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4552 027006 012737 000100 004170 MOV #IR.E.CS2 ;LOAD EXPECTED CS2
4553 027014 012737 100401 004172 MOV #SVAL!DRA!DDT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
4554 027022 012737 000000 004174 MOV #O.E.ER ;LOAD EXPECTED ERROR REG
4555 027030 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG.1 CORRECT
4556 027036 001401 BEQ 4$ ;YES, CONTINUE
4557 027040 104154 ERROR 154
4558 027042 023737 004170 004130 4$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4559 027050 001401 BEQ 5$ ;YES, CONTINUE
4560 027052 104155 ERROR 155
4561 027054 023737 004172 004132 5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG. CORRECT
4562 027062 001401 BEQ 6$ ;YES, CONTINUE
4563 027064 104156 ERROR 156
4564 027066 023737 004174 004134 6$: CMP E.ER,T.ER ;CHECK ERROR REGISTER CORRECT
4565 027074 001401 BEQ 7$ ;YES, CLEAR RK611
4566 027076 104157 ERROR 157
4567 027100 013737 004120 004220 7$: MOV T.CS1,P.CS1 ;STORE PREVIOUS CONTENTS OF
4568 027106 013737 004130 004222 MOV T.CS2,P.CS2 ;COMMAND AND STATUS REG 1
4569 027114 013737 004132 004224 MOV T.DS,P.DS ;COMMAND AND STATUS REG 2
4570 027122 013737 004134 004226 MOV T.ER,P.ER ;DRIVE STATUS REG
4571 ;AND ERROR REG
4572 027130 012762 100000 000000 MOV #CLR,RKCS1(R2) ;CLEAR RK611
4573 027136 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4574 027144 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4575 027152 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
4576 027160 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
4577 027166 012737 000200 004160 MOV #RDY,E.CS1 ;LOAD EXPECTED CS1
4578 027174 012737 000100 004170 MOV #IR.E.CS2 ;LOAD EXPECTED CS2
4579 027202 005037 004172 CLR E.DS ;LOAD EXPECTED DRIVE STATUS REG
4580 027206 005037 004174 CLR E.ER ;LOAD EXPECTED ERROR REG
4581 027212 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG 1 CORRECT
4582 027220 001401 BEQ 11$ ;YES, CHECK CS2
4583 027222 104224 ERROR 224 ;CS1 INCORRECT
4584 027224 023737 004170 004130 11$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG 2 CORRECT
4585 027232 001401 BEQ 12$ ;YES, CHECK DRIVE STATUS REG
4586 027234 104225 ERROR 225 ;CS2 INCORRECT
4587 027236 023737 004172 004132 12$: CMP E.DS,T.DS ;CHECK IF DRIVE STATUS REG CORRECT
4588 027244 001401 BEQ 13$ ;YES, CHECK ERROR REG
4589 027246 104226 ERROR 226 ;ERROR REG INCORRECT
4590 027250 023737 004174 004134 13$: CMP E.ER,T.ER ;CHECK IF ERROR REG CORRECT
4591 027256 001401 BEQ TST55 ;YES, GO ON TO NEXT TEST
4592 027260 104227 ERROR 227 ;ERROR REG INCORRECT

```

```

*****
;*TEST 55 CDT SET AND DRIVE TYPE ERROR

```

```

4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606 027262 000004
4607 027264 012737 000144 001200
4608 027272 013702 001270
4609 027276 012762 000040 000010
4610 027304 012762 000040 000026
4611 027312 012762 000002 000020
4612 027320 012762 000000 000006
4613 027326 012762 002017 000000
4614 027334 012700 000132
4615 027340 012762 000440 000026 15:
4616 027346 012762 000040 000026
4617 027354 005300
4618 027356 001370
4619 027360 005062 000026
4620 027364 013700 004262
4621 027370 105762 000000 25:
4622 027374 100402
4623 027376 005300
4624 027400 001373
4625 027402 016237 000000 004120 35:
4626 027410 016237 000010 004130
4627 027416 016237 000012 004132
4628 027424 016237 000014 004134
4629 027432 012737 102216 004160
4630 027440 012737 000100 004170
4631 027446 012737 100001 004172
4632 027454 012737 000040 004174
4633 027462 023737 004160 004120
4634 027470 001401
4635 027472 104160
4636 027474 023737 004170 004130 45:
4637 027502 001401
4638 027504 104161
4639 027506 023737 004172 004132 55:
4640 027514 001401
4641 027516 104162
4642 027520 023737 004174 004134 65:
4643 027526 001401
4644 027530 104163
4645 027532 013737 004120 004220 75:
4646 027540 013737 004130 004222
4647 027546 013737 004132 004224
4648 027554 013737 004134 004226
4649
4650 027562 012762 100000 000000
4651 027570 016237 000000 004120

```

```

**
** CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
** PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
** WITH CDT SET, 26 SECTOR FORMAT, TO CYLINDER 2,
** HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE
** UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE
** AND MAKE SURE DRIVE AVAILABLE, DRIVE TYPE ERROR,
** AND CONTROLLER ERROR SET.
**
*****
T55: SCOPE
MOV #100.,$TIMES ;;DO 100. ITERATIONS
MOV $BASE,R2 ;;LOAD RK611 BASE
MOV #SCLR,RKCS2(R2) ;;CLEAR RK06 SUBSYSTEM
MOV #DMD,RKMR1(R2) ;;PUT RK611 IN MAINT MODE
MOV #2,RKDCYL(R2) ;;LOAD CYLINDER AND
MOV #0,RKDA(R2) ;;LOAD HEAD ADDRESS
MOV #CDT!SEEK,RKCS1(R2) ;;ISSUE CDT!SEEK
MOV #22.*4+2,R0 ;;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
15: MOV #DMD!MCLK,RKMR1(R2)
MOV #DMD,RKMR1(R2)
DEC R0
BNE 15
CLR RKMR1(R2) ;;FINISH COMMAND IN NORMAL MODE
MOV WAITIM,R0 ;;WAIT FOR READY
25: TSTB RKCS1(R2)
BMI 35
DEC R0
BNE 25
35: MOV RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG 1
MOV RKCS2(R2),T.CS2 ;;STORE COMMAND AND STATUS REG 2
MOV RKDS(R2),T.DS ;;STORE DRIVE STATUS REG
MOV RKER(R2),T.ER ;;STORE ERROR REG
MOV #CDT!CERR!RDY!CDT!SEEK<↑C<GO>>,E.CS1 ;;LOAD EXPECTED CS1
MOV #IR,E.CS2 ;;LOAD EXPECTED CS2
MOV #SVAL!DRA,E.DS ;;LOAD EXPECTED DRIVE STATUS REG
MOV #DTYE,E.ER ;;LOAD EXPECTED ERROR REG
CMP E.CS1,T.CS1 ;;CHECK COMMAND AND STATUS REG.1 CORRECT
BEQ 45 ;;YES, CONTINUE
45: ERROR 160
CMP E.CS2,T.CS2 ;;CHECK COMMAND AND STATUS REG. 2 CORRECT
BEQ 55 ;;YES, CONTINUE
55: ERROR 161
CMP E.DS,T.DS ;;CHECK DRIVE STATUS REG. CORRECT
BEQ 65 ;;YES, CONTINUE
65: ERROR 162
CMP E.ER,T.ER ;;CHECK ERROR REGISTER CORRECT
BEQ 75 ;;YES, CLEAR RK611
75: ERROR 163
MOV T.CS1,P.CS1 ;;STORE PREVIOUS CONTENTS OF
MOV T.CS2,P.CS2 ;;COMMAND AND STATUS REG 1
MOV T.DS,P.DS ;;COMMAND AND STATUS REG 2
MOV T.ER,P.ER ;;DRIVE STATUS REG
;;AND ERROR REG
MOV #CLR,RKCS1(R2) ;;CLEAR RK611
MOV RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG 1

```

```

4652 027576 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4653 027604 016237 000012 004132      MOV      RKDS(R2),T.DS  ;STORE DRIVE STATUS REG
4654 027612 016237 000014 004134      MOV      RKER(R2),T.ER  ;STORE ERROR REG
4655 027620 012737 000200 004160      MOV      #RDY,E.CS1    ;LOAD EXPECTED CS1
4656 027626 012737 000100 004170      MOV      #IR,E.CS2    ;LOAD EXPECTED CS2
4657 027634 005037 004172      CLR      E.DS         ;LOAD EXPECTED DRIVE STATUS REG
4658 027640 005037 004174      CLR      E.ER         ;LOAD EXPECTED ERROR REG
4659 027644 023737 004160 004120      CMP      E.CS1,T.CS1  ;CHECK COMMAND AND STATUS REG 1 CORRECT
4660 027652 001401      BEQ      11$          ;YES, CHECK CS2
4661 027654 104224      ERROR   224          ;CS1 INCORRECT
4662 027656 023737 004170 004130 11$:    CMP      E.CS2,T.CS2  ;CHECK COMMAND AND STATUS REG 2 CORRECT
4663 027664 001401      BEQ      12$          ;YES, CHECK DRIVE STATUS REG
4664 027666 104225      ERROR   225          ;CS2 INCORRECT
4665 027670 023737 004172 004132 12$:    CMP      E.DS,T.DS    ;CHECK IF DRIVE STATUS REG CORRECT
4666 027676 001401      BEQ      13$          ;YES, CHECK ERROR REG
4667 027700 104226      ERROR   226          ;ERROR REG INCORRECT
4668 027702 023737 004174 004134 13$:    CMP      E.ER,T.ER    ;CHECK IF ERROR REG CORRECT
4669 027710 001401      BEQ      TST56        ;YES, GO ON TO NEXT TEST
4670 027712 104227      ERROR   227          ;ERROR REG INCORRECT

```

```

*****
*TEST 56      RK06 AND DRIVE TYPE ERROR
*
*      CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR
*      PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
*      TO A RK06, 26 SECTOR FORMAT, TO CYLINDER 23,
*      HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE
*      UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC
*      MODE AND MAKE SURE DRIVE AVAILIABLE, DRIVE TYPE ERROR,
*      AND CONTROLLER ERROR SETS.
*****

```

```

4683 *****
4684 027714 000004      TST56: SCOPE
4685 027716 012737 000144 001200      MOV      #100,$TIMES ;DO 100. ITERATIONS
4686 027724 013702 001270      MOV      $BASE,R2    ;LOAD RK611 BASE
4687 027730 012762 000040 000010      MOV      #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4688 027736 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
4689 027744 012762 000023 000020      MOV      #23,RKDCYL(R2) ;LOAD CYLINDER AND
4690 027752 012762 000000 000006      MOV      #0,RKDA(R2)   ;LOAD HEAD ADDRESS
4691 027760 012762 000017 000000      MOV      #SEEK,RKCS1(R2) ;ISSUE SEEK
4692 027766 012700 000132      MOV      #22,*4+2,RO   ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4693 027772 012762 000440 000026 1$:    MOV      #DMD!MCLK,RKMR1(R2)
4694 030000 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4695 030006 005300      DEC      RO
4696 030010 001370      BNE     1$
4697 030012 005062 000026      CLR      RKMR1(R2)    ;FINISH COMMAND IN NORMAL MODE
4698 030016 013700 004262      MOV      WAITIM,RO    ;WAIT FOR READY
4699 030022 105762 000000 2$:    TSTB   RKCS1(R2)
4700 030026 100402      BMI     3$
4701 030030 005300      DEC      RO
4702 030032 001373      BNE     2$
4703 030034 016237 000000 004120 3$:    MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4704 030042 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4705 030050 016237 000012 004132      MOV      RKDS(R2),T.DS  ;STORE DRIVE STATUS REG
4706 030056 016237 000014 004134      MOV      RKER(R2),T.ER  ;STORE ERROR REG
4707 030064 012737 100216 004160      MOV      #CERR!RDY!SEEK<T.C<GO>>,E.CS1 ;LOAD EXPECTED CS1

```

```

4708 030072 012737 000100 004170      MOV      #IR,E.CS2      ;LOAD EXPECTED CS2
4709 030100 012737 100401 004172      MOV      #SVAL!DRA!DDT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
4710 030106 012737 000040 004174      MOV      #DTYE,E.ER     ;LOAD EXPECTED ERROR REG
4711 030114 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG.1 CORRECT
4712 030122 001401                BEQ      4$             ;YES, CONTINUE
4713 030124 104164                ERROR    164
4714 030126 023737 004170 004130 4$:      CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4715 030134 001401                BEQ      5$             ;YES, CONTINUE
4716 030136 104165                ERROR    165
4717 030140 023737 004172 004132 5$:      CMP      E.DS,T.DS      ;CHECK DRIVE STATUS REG. CORRECT
4718 030146 001401                BEQ      6$             ;YES, CONTINUE
4719 030150 104166                ERROR    166
4720 030152 023737 004174 004134 6$:      CMP      E.ER,T.ER      ;CHECK ERROR REGISTER CORRECT
4721 030160 001401                BEQ      7$             ;YES, CLEAR RK611
4722 030162 104167                ERROR    167
4723 030164 013737 004120 004220 7$:      MOV      T.CS1,P.CS1    ;STORE PREVIOUS CONTENTS OF
4724 030172 013737 004130 004222      MOV      T.CS2,P.CS2    ;COMMAND AND STATUS REG 1
4725 030200 013737 004132 004224      MOV      T.DS,P.DS      ;COMMAND AND STATUS REG 2
4726 030206 013737 004134 004226      MOV      T.ER,P.ER      ;DRIVE STATUS REG
4727                                ;AND ERROR REG
4728 030214 012762 100000 000000      MOV      #CLR,RKCS1(R2) ;CLEAR RK611
4729 030222 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4730 030230 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4731 030236 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
4732 030244 016237 000014 004134      MOV      RKR(R2),T.ER    ;STORE ERROR REG
4733 030252 012737 000200 004160      MOV      #RDY,E.CS1     ;LOAD EXPECTED CS1
4734 030260 012737 000100 004170      MOV      #IR,E.CS2     ;LOAD EXPECTED CS2
4735 030266 005037 004172                CLR      E.DS           ;LOAD EXPECTED DRIVE STATUS REG
4736 030272 005037 004174                CLR      E.ER           ;LOAD EXPECTED ERROR REG
4737 030276 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG 1 CORRECT
4738 030304 001401                BEQ      11$            ;YES, CHECK CS2
4739 030306 104224                ERROR    224            ;CS1 INCORRECT
4740 030310 023737 004170 004130 11$:      CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG 2 CORRECT
4741 030316 001401                BEQ      12$            ;YES, CHECK DRIVE STATUS REG
4742 030320 104225                ERROR    225            ;CS2 INCORRECT
4743 030322 023737 004172 004132 12$:      CMP      E.DS,T.DS      ;CHECK IF DRIVE STATUS REG CORRECT
4744 030330 001401                BEQ      13$            ;YES, CHECK ERROR REG
4745 030332 104226                ERROR    226            ;ERROR REG INCORRECT
4746 030334 023737 004174 004134 13$:      CMP      E.ER,T.ER      ;CHECK IF ERROR REG CORRECT
4747 030342 001401                BEQ      TST5?          ;YES, GO ON TO NEXT TEST
4748 030344 104227                ERROR    227            ;ERROR REG INCORRECT

```

```

4749
4750
4751 ;*****
4752 ;TEST 57      SPEED LOSS FROM SHIFT REG.
4753 ;
4754 ;      CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
4755 ;      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A SEEK TO A RK06,
4756 ;      26 SECTOR FORMAT, TO CYLINDER 3, HEAD 1, DRIVE 0.
4757 ;      CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.  TURN
4758 ;      OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE AND
4759 ;      SPEED LOSS ARE SET.
4760 ;*****
4761 030346 000004      †TST5: SCOPE
4762 030350 012737 000144 001200      MOV      #100,$TIMES   ;;DO 100. ITERATIONS
4763 030356 013702 001270                MOV      $BASE,R2      ;LOAD RK611 BASE

```



```

4820 030754 023737 004172 004132 12$: CMP E.DS,T.DS ;CHECK IF DRIVE STATUS REG CORRECT
4821 030762 001401 BEQ 13$ ;YES, CHECK ERROR REG
4822 030764 104226 ERROR 226 ;ERROR REG INCORRECT
4823 030766 023737 004174 004134 13$: CMP E.ER,T.ER ;CHECK IF ERROR REG CORRECT
4824 030774 001401 BEQ TST60 ;YES, GO ON TO NEXT TEST
4825 030776 104227 ERROR 227 ;ERROR REG INCORRECT

```

```

*****
:TEST 60 DRIVE OFF TRACK FROM SHIFT REG.

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK TO A RK06,
* 26 SECTOR FORMAT, TO CYLINDER 3, HEAD 2, DRIVE 0.
* CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
* TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE
* AND DRIVE OFF TRACK ARE SET.

```

```

*****
TST60: SCOPE

```

```

4838 031000 000004 TST60: MOV #100, $TIMES ;DO 100. ITERATIONS
4839 031002 012737 000144 001200 MOV $BASE,R2 ;LOAD RK611 BASE
4840 031010 013702 001270 MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4841 031014 012762 000040 000010 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
4842 031022 012762 000040 000026 MOV #3,RKDCYL(R2) ;LOAD CYLINDER AND
4843 031030 012762 000003 000020 MOV #1000,RKDA(R2) ;LOAD HEAD ADDRESS
4844 031036 012762 001000 000006 MOV #SEEK,RKCS1(R2) ;ISSUE SEEK
4845 031044 012762 000017 000000 MOV #22, #4+2, R0 ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4846 031052 012700 000132 MOV #DMD!MCLK,RKMR1(R2)
4847 031056 012762 000440 000026 1$: MOV #DMD,RKMR1(R2)
4848 031064 012762 000040 000026 MOV R0
4849 031072 005300 DEC 1$
4850 031074 001370 BNE 1$
4851 031076 005062 000026 CLR RKMR1(R2) ;FINISH COMMAND IN NORMAL MODE
4852 031102 013700 004262 MOV WAITIM,R0 ;WAIT FOR READY
4853 031106 105762 000000 2$: TSTB RKCS1(R2)
4854 031112 100402 BMI 3$
4855 031114 005300 DEC R0
4856 031116 001373 BNE 2$
4857 031120 016237 000000 004120 3$: MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4858 031126 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4859 031134 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
4860 031142 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
4861 031150 012737 000216 004160 MOV #RDY!SEEK&<↑C<GO>, E.CS1 ;LOAD EXPECTED CS1
4862 031156 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED CS2
4863 031164 012737 100041 004172 MOV #SVAL!DRA!DROT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
4864 031172 012737 000000 004174 MOV #0,E.ER ;LOAD EXPECTED ERROR REG
4865 031200 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG.1 CORRECT
4866 031206 001401 BEQ 4$ ;YES, CONTINUE
4867 031210 104174 ERROR 174
4868 031212 023737 004170 004130 4$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4869 031220 001401 BEQ 5$ ;YES, CONTINUE
4870 031222 104175 ERROR 175
4871 031224 023737 004172 004132 5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG. CORRECT
4872 031232 001401 BEQ 6$ ;YES, CONTINUE
4873 031234 104176 ERROR 176
4874 031236 023737 004174 004134 6$: CMP E.ER,T.ER ;CHECK ERROR REGISTER CORRECT
4875 031244 001401 BEQ 7$ ;YES, CLEAR RK611

```

```

4876 031246 104177          ERROR 177
4877 031250 013737 004120 004220 7$: MOV T.CS1,P.CS1 ;STORE PREVIOUS CONTENTS OF
4878 031256 013737 004130 004222 MOV T.CS2,P.CS2 ;COMMAND AND STATUS REG 1
4879 031264 013737 004132 004224 MOV T.DS,P.DS ;COMMAND AND STATUS REG 2
4880 031272 013737 004134 004226 MOV T.ER,P.ER ;DRIVE STATUS REG
4881 ;AND ERROR REG
4882 031300 012762 100000 000000 MOV #CLR,RKCS1(R2) ;CLEAR RK611
4883 031306 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4884 031314 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4885 031322 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
4886 031330 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
4887 031336 012737 000200 004160 MOV #RDY,E.CS1 ;LOAD EXPECTED CS1
4888 031344 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED CS2
4889 031352 005037 004172 CLR E.DS ;LOAD EXPECTED DRIVE STATUS REG
4890 031356 005037 004174 CLR E.ER ;LOAD EXPECTED ERROR REG
4891 031362 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG 1 CORRECT
4892 031370 001401 BEQ 11$ ;YES, CHECK CS2
4893 031372 104224 ERROR 224 ;CS1 INCORRECT
4894 031374 023737 004170 004130 11$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG 2 CORRECT
4895 031402 001401 BEQ 12$ ;YES, CHECK DRIVE STATUS REG
4896 031404 104225 ERROR 225 ;CS2 INCORRECT
4897 031406 023737 004172 004132 12$: CMP E.DS,T.DS ;CHECK IF DRIVE STATUS REG CORRECT
4898 031414 001401 BEQ 13$ ;YES, CHECK ERROR REG
4899 031416 104226 ERROR 226 ;ERROR REG INCORRECT
4900 031420 023737 004174 004134 13$: CMP E.ER,T.ER ;CHECK IF ERROR REG CORRECT
4901 031426 001401 BEQ TST61 ;YES, GO ON TO NEXT TEST
4902 031430 104227 ERROR 227 ;ERROR REG INCORRECT

```

```

*****
;TEST 61 WRITE LOCK ERROR FROM SHIFT REG.

```

```

;
; CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
; PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A PACK ACKNOWLEDGE
; TO A RK06, 26 SECTOR FORMAT, WITH CYLINDER 0,
; HEAD 1, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL
; PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE
; SURE SPEED LOSS, WRITE LOCK ERROR AND CONTROLLER ERROR
; ARE SET WITH DRIVE AVAILIABLE RESET.

```

```

*****
TST61: SCOPE

```

```

4916 031432 000004          MOV #100,$TIMES ;DO 100. ITERATIONS
4917 031434 012737 000144 001200 MOV $BASE,R2 ;LOAD RK611 BASE
4918 031442 013702 001270 MOV #CLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
4919 031446 012762 000040 000010 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
4920 031454 012762 000040 000026 MOV #0,RKDCYL(R2) ;LOAD CYLINDER AND
4921 031462 012762 000000 000020 MOV #400,RKDA(R2) ;LOAD HEAD ADDRESS
4922 031470 012762 000400 000006 MOV #PACK,RKCS1(R2) ;ISSUE PACK
4923 031476 012762 000003 000000 MOV #22,*4+2,RO ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
4924 031504 012700 000132 MOV #DMD:MCLK,RKMR1(R2)
4925 031510 012762 000440 000026 1$: MOV #DMD,RKMR1(R2)
4926 031516 012762 000040 000026 MOV RO
4927 031524 005300 DEC 1$
4928 031526 001370 BNE 1$
4929 031530 005062 000026 CLR RKMR1(R2) ;FINISH COMMAND IN NORMAL MODE
4930 031534 013700 004262 MOV WAITIM,RO ;WAIT FOR READY
4931 031540 105762 000000 2$: TSTB RKCS1(R2)

```

```

4932 031544 100402 BMI 3$
4933 031546 005300 DEC R0
4934 031550 001373 BNE 2$
4935 031552 016237 000000 004120 3$: MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4936 031560 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4937 031566 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
4938 031574 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
4939 031602 012737 100202 004160 MOV #CERR!RDY!PACK&<C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4940 031610 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED CS2
4941 031616 012737 100020 004172 MOV #SVAL!SPDLSS,E.DS ;LOAD EXPECTED DRIVE STATUS REG
4942 031624 012737 004000 004174 MOV #WLE,E.ER ;LOAD EXPECTED ERROR REG
4943 031632 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG.1 CORRECT
4944 031640 001401 BEQ 4$ ;YES, CONTINUE
4945 031642 104200 ERROR 200
4946 031644 023737 004170 004130 4$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG. 2 CORRECT
4947 031652 001401 BEQ 5$ ;YES, CONTINUE
4948 031654 104201 ERROR 201
4949 031656 023737 004172 004132 5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG. CORRECT
4950 031664 001401 BEQ 6$ ;YES, CONTINUE
4951 031666 104202 ERROR 202
4952 031670 023737 004174 004134 6$: CMP E.ER,T.ER ;CHECK ERROR REGISTER CORRECT
4953 031676 001401 BEQ 7$ ;YES, CLEAR RK611
4954 031700 104203 ERROR 203
4955 031702 013737 004120 004220 7$: MOV T.CS1,P.CS1 ;STORE PREVIOUS CONTENTS OF
4956 031710 013737 004130 004222 MOV T.CS2,P.CS2 ;COMMAND AND STATUS REG 1
4957 031716 013737 004132 004224 MOV T.DS,P.DS ;COMMAND AND STATUS REG 2
4958 031724 013737 004134 004226 MOV T.ER,P.ER ;DRIVE STATUS REG
4959 ;AND ERROR REG
4960 031732 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611
4961 031740 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4962 031746 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4963 031754 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
4964 031762 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
4965 031770 012737 000200 004160 MOV #RDY,E.CS1 ;LOAD EXPECTED CS1
4966 031776 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED CS2
4967 032004 005037 004172 CLR E.DS ;LOAD EXPECTED DRIVE STATUS REG
4968 032010 005037 004174 CLR E.ER ;LOAD EXPECTED ERROR REG
4969 032014 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG 1 CORRECT
4970 032022 001401 BEQ 11$ ;YES, CHECK CS2
4971 032024 104224 ERROR 224 ;CS1 INCORRECT
4972 032026 023737 004170 004130 11$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG 2 CORRECT
4973 032034 001401 BEQ 12$ ;YES, CHECK DRIVE STATUS REG
4974 032036 104225 ERROR 225 ;CS2 INCORRECT
4975 032040 023737 004172 004132 12$: CMP E.DS,T.DS ;CHECK IF DRIVE STATUS REG CORRECT
4976 032046 001401 BEQ 13$ ;YES, CHECK ERROR REG
4977 032050 104226 ERROR 226 ;ERROR REG INCORRECT
4978 032052 023737 004174 004134 13$: CMP E.ER,T.ER ;CHECK IF ERROR REG CORRECT
4979 032060 001401 BEQ TST62 ;YES, GO ON TO NEXT TEST
4980 032062 104227 ERROR 227 ;ERROR REG INCORRECT

```

```

4981
4982 *****
4983 *TEST 62 SEEK INCOMPLETE
4984 *
4985 * CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
4986 * PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE AN UNLOAD
4987 * TO A RK06, 26 SECTOR FORMAT, WITH CYLINDER 0,

```

```

4988      *      HEAD 1, DRIVE 0, CLOCK IN DIAGNOSTIC MODE UNTIL
4989      *      PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE
4990      *      SURE SPEED LOSS, SEEK INCOMPLETE, AND CONTROLLER ERROR
4991      *      ARE SET WITH DRIVE AVAILIABLE RESET.
4992      *
4993      * *****
4994      * ST62: SCOPE
4995      *      MOV      #100, $TIMES      ;; DO 100. ITERATIONS
4996      *      MOV      $BASE, R2        ;; LOAD RK611 BASE
4997      *      MOV      #SCLR, RKCS2(R2) ;; CLEAR RK06 SUBSYSTEM
4998      *      MOV      #DMD, RKMR1(R2)  ;; PUT RK611 IN MAINT MODE
4999      *      MOV      #0, RKDCYL(R2)   ;; LOAD CYLINDER AND
5000      *      MOV      #400, RKDA(R2)   ;; LOAD HEAD ADDRESS
5001      *      MOV      #UNLOAD, RKCS1(R2) ;; ISSUE UNLOAD
5002      *      MOV      #22, *4+2, R0    ;; ISSUE CLOCKS UNTIL PHASE ADDRESS 6
5003      *      MOV      #DMD!MCLK, RKMR1(R2)
5004      *      MOV      #DMD, RKMR1(R2)
5005      *      DEC      R0
5006      *      BNE     1$
5007      *      CLR     RKMR1(R2)        ;; FINISH COMMAND IN NORMAL MODE
5008      *      MOV     WAITIM, R0      ;; WAIT FOR READY
5009      *      TSTB   RKCS1(R2)
5010      *      BMI     3$
5011      *      DEC     R0
5012      *      BNE     2$
5013      *      MOV     RKCS1(R2), T.CS1 ;; STORE COMMAND AND STATUS REG 1
5014      *      MOV     RKCS2(R2), T.CS2 ;; STORE COMMAND AND STATUS REG 2
5015      *      MOV     RKDS(R2), T.DS   ;; STORE DRIVE STATUS REG
5016      *      MOV     RKER(R2), T.ER   ;; STORE ERROR REG
5017      *      MOV     #CERR!RDY!UNLOAD<↑C<GO>>, E.CS1 ;; LOAD EXPECTED CS1
5018      *      MOV     #IR, E.CS2      ;; LOAD EXPECTED CS2
5019      *      MOV     #SVAL!SPDLSS, E.DS ;; LOAD EXPECTED DRIVE STATUS REG
5020      *      MOV     #SKI, E.ER      ;; LOAD EXPECTED ERROR REG
5021      *      CMP     E.CS1, T.CS1    ;; CHECK COMMAND AND STATUS REG.1 CORRECT
5022      *      BEQ     4$              ;; YES, CONTINUE
5023      *      ERROR   204
5024      *      CMP     E.CS2, T.CS2    ;; CHECK COMMAND AND STATUS REG. 2 CORRECT
5025      *      BEQ     5$              ;; YES, CONTINUE
5026      *      ERROR   205
5027      *      CMP     E.DS, T.DS     ;; CHECK DRIVE STATUS REG. CORRECT
5028      *      BEQ     6$              ;; YES, CONTINUE
5029      *      ERROR   206
5030      *      CMP     E.ER, T.ER     ;; CHECK ERROR REGISTER CORRECT
5031      *      BEQ     7$              ;; YES, CLEAR RK611
5032      *      ERROR   207
5033      *      MOV     T.CS1, P.CS1    ;; STORE PREVIOUS CONTENTS OF
5034      *      MOV     T.CS2, P.CS2    ;; COMMAND AND STATUS REG 1
5035      *      MOV     T.DS, P.DS     ;; COMMAND AND STATUS REG 2
5036      *      MOV     T.ER, P.ER     ;; DRIVE STATUS REG
5037      *      MOV     #CCLR, RKCS1(R2) ;; CLEAR RK611
5038      *      MOV     RKCS1(R2), T.CS1 ;; STORE COMMAND AND STATUS REG 1
5039      *      MOV     RKCS2(R2), T.CS2 ;; STORE COMMAND AND STATUS REG 2
5040      *      MOV     RKDS(R2), T.DS   ;; STORE DRIVE STATUS REG
5041      *      MOV     RKER(R2), T.ER   ;; STORE ERROR REG
5042      *      MOV     #RDY, E.CS1     ;; LOAD EXPECTED CS1
5043      *

```

```

5044 032430 012737 000100 004170      MOV  #IR.E.CS2      ;LOAD EXPECTED CS2
5045 032436 005037 004172      CLR  E.DS           ;LOAD EXPECTED DRIVE STATUS REG
5046 032442 005037 004174      CLR  E.ER           ;LOAD EXPECTED ERROR REG
5047 032446 023737 004160 004120      CMP  E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG 1 CORRECT
5048 032454 001401                BEQ  11$            ;YES, CHECK CS2
5049 032456 104224                ERROR 224           ;CS1 INCORRECT
5050 032460 023737 004170 004130 11$:      CMP  E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG 2 CORRECT
5051 032466 001401                BEQ  12$            ;YES, CHECK DRIVE STATUS REG
5052 032470 104225                ERROR 225           ;CS2 INCORRECT
5053 032472 023737 004172 004132 12$:      CMP  E.DS,T.DS      ;CHECK IF DRIVE STATUS REG CORRECT
5054 032500 001401                BEQ  13$            ;YES, CHECK ERROR REG
5055 032502 104226                ERROR 226           ;ERROR REG INCORRECT
5056 032504 023737 004174 004134 13$:      CMP  E.ER,T.ER      ;CHECK IF ERROR REG CORRECT
5057 032512 001401                BEQ  TST63          ;YES, GO ON TO NEXT TEST
5058 032514 104227                ERROR 227           ;ERROR REG INCORRECT

```

```

*****
*TEST 63      NON-EXECUTABLE DRIVE FUNCTION FROM SHIFT REG.
*
*          CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR
*          PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE
*          A DRIVE CLEAR TO A RK06, 26 SECTOR FORMAT,
*          WITH CYLINDER 0, HEAD 1, DRIVE 0. CLOCK IN DIAGNOSTIC
*          MODE UNTIL PHASE ADDRESS 6. TURN OFF DIAGNOSTIC
*          MODE AND MAKE SURE SPEED LOSS, NON-EXECUTABLE DRIVE FUNCTION, AND
*          CONTROLLER ERROR ARE SET WITH DRIVE AVAILAIBLE RESET.
*****

```

```

5072 032516 000004                TST63: SCOPE
5073 032520 012737 000144 001200      MOV  #100.,$TIMES    ;DO 100. ITERATIONS
5074 032526 013702 001270                MOV  $BASE,R2        ;LOAD RK611 BASE
5075 032532 012762 000040 000010      MOV  #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
5076 032540 012762 000040 000026      MOV  #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
5077 032546 012762 000000 000020      MOV  #0,RKDCYL(R2)  ;LOAD CYLINDER AND
5078 032554 012762 000400 000006      MOV  #400,RKDA(R2)  ;LOAD HEAD ADDRESS
5079 032562 012762 000005 000000      MOV  #CLEAR,RKCS1(R2) ;ISSUE CLEAR
5080 032570 012700 000132                MOV  #22.*4+2,RO     ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
5081 032574 012762 000440 000026 1$:      MOV  #DMD!MCLK,RKMR1(R2)
5082 032602 012762 000040 000026      MOV  #DMD,RKMR1(R2)
5083 032610 005300                DEC  RO
5084 032612 001370                BNE  1$
5085 032614 005062 000026                CLR  RKMR1(R2)       ;FINISH COMMAND IN NORMAL MODE
5086 032620 013700 004262                MOV  WAITIM,RO       ;WAIT FOR READY
5087 032624 105762 000000 2$:      TSTB  RKCS1(R2)
5088 032630 100402                BMI  3$
5089 032632 005300                DEC  RO
5090 032634 001373                BNE  2$
5091 032636 016237 000000 004120 3$:      MOV  RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
5092 032644 016237 000010 004130      MOV  RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
5093 032652 016237 000012 004132      MOV  RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
5094 032660 016237 000014 004134      MOV  RKER(R2),T.ER   ;STORE ERROR REG
5095 032666 012737 100204 004160      MOV  #CERR!RDY!CLEAR<C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5096 032674 012737 000100 004170      MOV  #IR.E.CS2      ;LOAD EXPECTED CS2
5097 032702 012737 100020 004172      MOV  #SVAL!SPDLSS,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5098 032710 012737 000004 004174      MOV  #NXF.E.ER      ;LOAD EXPECTED ERROR REG
5099 032716 023737 004160 004120      CMP  E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG.1 CORRECT

```

```

5100 032724 001401 BEQ 4$ ;YES, CONTINUE
5101 032726 104210 ERROR 210
5102 032730 023737 004170 004130 4$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG. 2 CORRECT
5103 032736 001401 BEQ 5$ ;YES, CONTINUE
5104 032740 104211 ERROR 211
5105 032742 023737 004172 004132 5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG. CORRECT
5106 032750 001401 BEQ 6$ ;YES, CONTINUE
5107 032752 104212 ERROR 212
5108 032754 023737 004174 004134 6$: CMP E.ER,T.ER ;CHECK ERROR REGISTER CORRECT
5109 032762 001401 BEQ 7$ ;YES, CLEAR RK611
5110 032764 104213 ERROR 213
5111 032766 013737 004120 004220 7$: MOV T.CS1,P.CS1 ;STORE PREVIOUS CONTENTS OF
5112 032774 013737 004130 004222 MOV T.CS2,P.CS2 ;COMMAND AND STATUS REG 1
5113 033002 013737 004132 004224 MOV T.DS,P.DS ;COMMAND AND STATUS REG 2
5114 033010 013737 004134 004226 MOV T.ER,P.ER ;DRIVE STATUS REG
5115 ;AND ERROR REG
5116 033016 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611
5117 033024 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
5118 033032 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
5119 033040 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
5120 033046 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
5121 033054 012737 000200 004160 MOV #RDY,E.CS1 ;LOAD EXPECTED CS1
5122 033062 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED CS2
5123 033070 005037 004172 CLR E.DS ;LOAD EXPECTED DRIVE STATUS REG
5124 033074 005037 004174 CLR E.ER ;LOAD EXPECTED ERROR REG
5125 033100 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG 1 CORRECT
5126 033106 001401 BEQ 11$ ;YES, CHECK CS2
5127 033110 104224 ERROR 224 ;CS1 INCORRECT
5128 033112 023737 004170 004130 11$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG 2 CORRECT
5129 033120 001401 BEQ 12$ ;YES, CHECK DRIVE STATUS REG
5130 033122 104225 ERROR 225 ;CS2 INCORRECT
5131 033124 023737 004172 004132 12$: CMP E.DS,T.DS ;CHECK IF DRIVE STATUS REG CORRECT
5132 033132 001401 BEQ 13$ ;YES, CHECK ERROR REG
5133 033134 104226 ERROR 226 ;ERROR REG INCORRECT
5134 033136 023737 004174 004134 13$: CMP E.ER,T.ER ;CHECK IF ERROR REG CORRECT
5135 033144 001401 BEQ TST64 ;YES, GO ON TO NEXT TEST
5136 033146 104227 ERROR 227 ;ERROR REG INCORRECT

```

```

*****
*TEST 64 AC LOW AND C-D PARITY FROM SHIFT REG.
*

```

```

* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR, PUT RK611
* CONTROLLER IN DIAGNOSTIC MODE. ISSUE A START SPINDLE
* TO AN RK06, IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0,
* DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
* TURN OFF DIAGNOSTIC MODE AND MAKE SURE AC LOW, DRIVE
* DETECTED SERCOM PARITY, AND CONTROLLER ERROR SET WITH
* DRIVE AVAILABLE RESET.
*

```

```

*****
TST64: SCOPE

```

```

5150 033150 000004 MOV #100,$TIMES ;DO 100. ITERATIONS
5151 033152 012737 000144 001200 MOV $BASE,R2 ;LOAD RK611 BASE
5152 033160 013702 001270 MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
5153 033164 012762 000040 000010 MOV #DMD,RKMRI(R2) ;PUT RK611 IN MAINT MODE
5154 033172 012762 000040 000026 MOV #SRTSPL!CFMT,RKCS1(R2) ;ISSUE SRTSPL!CFMT
5155 033200 012762 010011 000000

```



5212 033564 104227

ERROR 227 ;ERROR REG INCORRECT

5213  
5214  
5215  
5216  
5217  
5218  
5219  
5220  
5221  
5222  
5223  
5224  
5225

```
*****
*TEST 65 ILLEGAL DISK ADDRESS ERROR FROM SHIFT REG.
*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
* RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A RECALIBRATE
* TO AN RK06, IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 1,
* DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE
* ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE
* SPEED LOSS, ILLEGAL DISK ADDRESS ERROR, AND CONTROLLER
* ERROR ARE SET WITH DRIVE AVAILABLE RESET.
*****
```

5226 033566 000004  
5227 033570 012737 000144 001200  
5228 033576 013702 001270  
5229 033602 012762 000040 000010  
5230 033610 012762 000040 000026  
5231 033616 012762 000000 000020  
5232 033624 012762 000400 000006  
5233 033632 012762 000013 000000  
5234 033640 012700 000132  
5235 033644 012762 000440 000026 1\$:  
5236 033652 012762 000040 000026  
5237 033660 005300  
5238 033662 001370  
5239 033664 005062 000026  
5240 033670 013700 004262  
5241 033674 105762 000000 2\$:  
5242 033700 100402  
5243 033702 005300  
5244 033704 001373  
5245 033706 016237 000000 004120 3\$:  
5246 033714 016237 000010 004130  
5247 033722 016237 000012 004132  
5248 033730 016237 000014 004134  
5249 033736 012737 100212 004160  
5250 033744 012737 000100 004170  
5251 033752 012737 100020 004172  
5252 033760 012737 002000 004174  
5253 033766 023737 004160 004120  
5254 033774 001401  
5255 033776 104220  
5256 034000 023737 004170 004130 4\$:  
5257 034006 001401  
5258 034010 104221  
5259 034012 023737 004172 004132 5\$:  
5260 034020 001401  
5261 034022 104222  
5262 034024 023737 004174 004134 6\$:  
5263 034032 001401  
5264 034034 104223  
5265 034036 013737 004120 004220 7\$:  
5266 034044 013737 004130 004222  
5267 034052 013737 004132 004224

```
TST65: SCOPE
MOV #100,$TIMES ;DO 100. ITERATIONS
MOV $BASE,R2 ;LOAD RK611 BASE
MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINT MODE
MOV #0,RKDCYL(R2) ;LOAD CYLINDER AND
MOV #400,RKDA(R2) ;LOAD HEAD ADDRESS
MOV #RECAL,RKCS1(R2) ;ISSUE RECAL
MOV #22,*4+2,R0 ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
1$: MOV #DMD:MCLK,RKMR1(R2)
MOV #DMD,RKMR1(R2)
DEC R0
BNE 1$
CLR RKMR1(R2) ;FINISH COMMAND IN NORMAL MODE
MOV WAITIM,R0 ;WAIT FOR READY
2$: TSTB RKCS1(R2)
BMI 3$
DEC R0
BNE 2$
3$: MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
MOV RKER(R2),T.ER ;STORE ERROR REG
MOV #CERR:RDY:RECAL<C<GO>>,E.CS1 ;LOAD EXPECTED CS1
MOV #IR,E.CS2 ;LOAD EXPECTED CS2
MOV #SVAL:SPDLSS,E.DS ;LOAD EXPECTED DRIVE STATUS REG
MOV #IDAE,E.ER ;LOAD EXPECTED ERROR REG
CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG.1 CORRECT
BEQ 4$ ;YES, CONTINUE
ERROR 220
4$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG. 2 CORRECT
BEQ 5$ ;YES, CONTINUE
ERROR 221
5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG. CORRECT
BEQ 6$ ;YES, CONTINUE
ERROR 222
6$: CMP E.ER,T.ER ;CHECK ERROR REGISTER CORRECT
BEQ 7$ ;YES, CLEAR RK611
ERROR 223
7$: MOV T.CS1,P.CS1 ;STORE PREVIOUS CONTENTS OF
MOV T.CS2,P.CS2 ; COMMAND AND STATUS REG 1
MOV T.DS,P.DS ; COMMAND AND STATUS REG 2
```

```

5268 034060 013737 004134 004226      MOV      T.ER,P.ER      ; DRIVE STATUS REG
5269                                     ; AND ERROR REG
5270 034066 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ; CLEAR RK611
5271 034074 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ; STORE COMMAND AND STATUS REG 1
5272 034102 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ; STORE COMMAND AND STATUS REG 2
5273 034110 016237 000012 004132      MOV      RKDS(R2),T.DS   ; STORE DRIVE STATUS REG
5274 034116 016237 000014 004134      MOV      RKER(R2),T.ER   ; STORE ERROR REG
5275 034124 012737 000200 004160      MOV      #RDY,E.CS1     ; LOAD EXPECTED CS1
5276 034132 012737 000100 004170      MOV      #IR,E.CS2     ; LOAD EXPECTED CS2
5277 034140 005037 004172      CLR      E.DS          ; LOAD EXPECTED DRIVE STATUS REG
5278 034144 005037 004174      CLR      E.ER          ; LOAD EXPECTED ERROR REG
5279 034150 023737 004160 004120      CMP      E.CS1,T.CS1    ; CHECK COMMAND AND STATUS REG 1 CORRECT
5280 034156 001401 11$      BEQ      11$          ; YES, CHECK CS2
5281 034160 104224      ERROR    224          ; CS1 INCORRECT
5282 034162 023737 004170 004130 11$:    CMP      E.CS2,T.CS2    ; CHECK COMMAND AND STATUS REG 2 CORRECT
5283 034170 001401 12$      BEQ      12$          ; YES, CHECK DRIVE STATUS REG
5284 034172 104225      ERROR    225          ; CS2 INCORRECT
5285 034174 023737 004172 004132 12$:    CMP      E.DS,T.DS     ; CHECK IF DRIVE STATUS REG CORRECT
5286 034202 001401 13$      BEQ      13$          ; YES, CHECK ERROR REG
5287 034204 104226      ERROR    226          ; ERROR REG INCORRECT
5288 034206 023737 004174 004134 13$:    CMP      E.ER,T.ER     ; CHECK IF ERROR REG CORRECT
5289 034214 001401      BEQ      TST66        ; YES, GO ON TO NEXT TEST
5290 034216 104227      ERROR    227          ; ERROR REG INCORRECT

```

```

*****
*TEST 66      IDAE DETECTION IN RK611 CONTROLLER (PART 1)

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
* RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A
* SEEK TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 1003,
* HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL
* PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE
* SURE DRIVE AVAILABLE, ILLEGAL DISK ADDRESS ERROR,
* AND CONTROLLER ERROR ARE SET.

```

```

5303 *****
5304 034220 000004      TST66: SCOPE
5305 034222 012737 000144 001200      MOV      #100.,$TIMES ; DO 100. ITERATIONS
5306 034230 013702 001270      MOV      $BASE,R2     ; LOAD RK611 BASE
5307 034234 012762 000040 000010      MOV      #SCLR,RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
5308 034242 012762 000040 000026      MOV      #DMD,RKMR1(R2) ; PUT RK611 IN MAINTENANCE MODE
5309 034250 012762 001002 000020      MOV      #1002,RKDCYL(R2) ; LOAD CYLINDER ADDRESS
5310 034256 012737 001002 004252      MOV      #1002,CYLIN
5311 034264 012737 000000 004250      MOV      #0,HDCODE    ; LOAD HEAD ADDRESS
5312 034272 005046      CLR      -(SP)
5313 034274 113766 004250 000001      MOV      HDCODE,1(SP)
5314 034302 012662 000006      MOV      (SP)+,RKDA(R2)
5315 034306 012737 000006 004266      MOV      #6,DRV Typ   ; LOAD DRIVE TYPE FOR PRINT OUT
5316 034314 012762 000017 000000      MOV      #SEEK,RKCS1(R2) ; ISSUE SEEK TO RK06
5317 034322 012700 000132      MOV      #22,*4+2,RO   ; ISSUE CLOCK TO GET THROUGH PHASE 6
5318 034326 012762 000440 000026 1$:    MOV      #DMD:MCLK,RKMR1(R2)
5319 034334 012762 000040 000026      MOV      #DMD,RKMR1(R2)
5320 034342 005300      DEC      RO
5321 034344 001370      BNE     1$
5322 034346 005062 000026      CLR      RKMR1(R2)    ; ALLOW COMMAND TO FINISH
5323 034352 013700 004262      MOV      WAITIM,RO    ; LOAD WAIT TIME

```

```

5324 034356 105762 000000 2$: TSTB RKCS1(R2) ;WAIT FOR READY
5325 034362 100402 BMI 3$
5326 034364 005300 DEC RD
5327 034366 001373 BNE 2$
5328 034370 016237 000000 004120 3$: MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5329 034376 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5330 034404 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG
5331 034412 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
5332 034420 012737 100216 004160 MOV #CERR!RDY!<SEEK&↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5333
5334 034426 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED COMMAND AND STATUS REG.2
5335 034434 012737 100001 004172 MOV #SVAl!DRA,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5336 034442 012737 002000 004174 MOV #IDAE,E.ER ;LOAD EXPECTED ERROR REG
5337 034450 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG1 CORRECT
5338 034456 001401 BEQ 4$ ;YES, CHECK CS2
5339 034460 104230 ERROR 230 ;CS1 INCORRECT
5340 034462 023737 004170 004130 4$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG2 CORRECT
5341 034470 001401 BEQ 5$ ;YES, CHECK DRIVE STATUS REG.
5342 034472 104231 ERROR 231 ;CS2 INCORRECT
5343 034474 023737 004172 004132 5$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG. CORRECT
5344 034502 001401 BEQ 6$ ;YES, CHECK ERROR REG
5345 034504 104232 ERROR 232 ;DRIVE STATUS REG. INCORRECT
5346 034506 023737 004174 004134 6$: CMP E.ER,T.ER ;CHECK ERROR REG. CORRECT
5347 034514 001401 BEQ 7$ ;YES, CHECK CONTROLLER CLEAR
5348 034516 104233 ERROR 233 ;ERROR REG. INCORRECT
5349 034520 013737 004120 004220 7$: MOV T.CS1,P.CS1 ;STORE PREVIOUS VALUES OF
5350 034526 013737 004130 004222 MOV T.CS2,P.CS2 ;COMMAND AND STATUS REG.1
5351 034534 013737 004132 004224 MOV T.DS,P.DS ;COMMAND AND STATUS REG.2
5352 034542 013737 004134 004226 MOV T.ER,P.ER ;DRIVE STATUS REG.
5353 ;ERROR REG.
5354 034550 012762 100000 000000 MOV #CLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
5355 034556 016237 000000 004120 MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5356 034564 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5357 034572 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG.
5358 034600 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
5359 034606 012737 000200 004160 MOV #RDY,E.CS1 ;LOAD EXPECTED CS1
5360 034614 012737 000100 004170 MOV #IR,E.CS2 ;LOAD EXPECTED CS2
5361 034622 005037 004172 CLR E.DS ;LOAD EXPECTED DRIVE STATUS REG.
5362 034626 005037 004174 CLR E.ER ;LOAD EXPECTED ERROR REG.
5363 034632 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG.1 CORRECT
5364 034640 001401 BEQ 10$ ;YES, CHECK CS2
5365 034642 104224 ERROR 224 ;CS1 INCORRECT
5366 034644 023737 004170 004130 10$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG2 CORRECT
5367 034652 001401 BEQ 11$ ;YES, CHECK DRIVE STATUS REG
5368 034654 104225 ERROR 225 ;CS2 INCORRECT
5369 034656 023737 004172 004132 11$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG CORRECT
5370 034664 001401 BEQ 12$ ;YES, CHECK ERROR REGISTER
5371 034666 104226 ERROR 226 ;DRIVE STATUS REG INCORRECT
5372 034670 023737 004174 004134 12$: CMP E.ER,T.ER ;CHECK ERROR REG CORRECT
5373 034676 001401 BEQ TST67 ;YES, GO ON TO NEXT TEST
5374 034700 104227 ERROR 227 ;ERROR REG. INCORRECT

```

```

*****
*TEST 67 IDAE DETECTION IN RK611 CONTROLLER (PART 2)
*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT

```

```

5375
5376
5377
5378
5379

```

D10

```

5380          ;*      RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A SEEK
5381          ;*      WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 1022, HEAD
5382          ;*      0, DRIVE 0.  CLOCK IN DIAGNOSTIC MODE UNTIL PHASE
5383          ;*      ADDRESS 6.  TURN OFF DIAGNOSTIC MODE AND MAKE SURE
5384          ;*      DRIVE AVAILABLE AND POSITIONING IN PROGRESS, ARE SET
5385          ;*      WITH ILLEGAL DISK ADDRESS ERROR RESET.
5386          ;*
5387          ;*
5388          ;*****
5388 034702 000004          TST67: SCOPE
5389 034704 012737 000144 001200      MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
5390 034712 013702 001270          MOV      $BASE,R2          ;;LOAD RK611 BASE
5391 034716 012762 000040 000010      MOV      #SCLR,RKCS2(R2)  ;;CLEAR RK06 SUBSYSTEM
5392 034724 012762 000040 000026      MOV      #DMD,RKMR1(R2)  ;;PUT RK611 IN MAINTENANCE MODE
5393 034732 012762 001022 000020      MOV      #1022,RKDCYL(R2) ;;LOAD CYLINDER ADDRESS
5394 034740 012737 001022 004252      MOV      #1022,CYLIN
5395 034746 012737 000000 004250      MOV      #0,HDCODE        ;LOAD HEAD ADDRESS
5396 034754 005046          CLR      -(SP)
5397 034756 113766 004250 000001      MOVVB   HDCODE,1(SP)
5398 034764 012662 000006          MOV      (SP)+,RKDA(R2)
5399 034770 012737 000007 004266      MOV      #7,DRV TYP      ;LOAD DRIVE TYPE FOR PRINT OUT
5400 034776 012762 002017 000000      MOV      #CDT!SEEK,RKCS1(R2) ;;ISSUE SEEK TO RK06
5401 035004 012700 000132          MOV      #22.*4+2,R0      ;ISSUE CLOCK TO GET THROUGH PHASE 6
5402 035010 012762 000440 000026 1$:  MOV      #DMD!MCLK,RKMR1(R2)
5403 035016 012762 000040 000026      MOV      #DMD,RKMR1(R2)
5404 035024 005300          DEC      R0
5405 035026 001370          BNE     1$
5406 035030 005062 000026          CLR      RKMR1(R2)        ;ALLOW COMMAND TO FINISH
5407 035034 013700 004262          MOV      WAITIM,R0        ;LOAD WAIT TIME
5408 035040 105762 000000 2$:  TSTB    RKCS1(R2)        ;WAIT FOR READY
5409 035044 100402          BMI     3$
5410 035046 005300          DEC      R0
5411 035050 001373          BNE     2$
5412 035052 016237 000000 004120 3$:  MOV      RKCS1(R2),T.CS1  ;STORE COMMAND AND STATUS REG.1
5413 035060 016237 000010 004130      MOV      RKCS2(R2),T.CS2  ;STORE COMMAND AND STATUS REG.2
5414 035066 016237 000012 004132      MOV      RKDS(R2),T.DS    ;STORE DRIVE STATUS REG
5415 035074 016237 000014 004134      MOV      RKER(R2),T.ER    ;STORE ERROR REG
5416 035102 012737 002216 004160      MOV      #CDT!RDY!<SEEK&↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5417
5418 035110 012737 000100 004170      MOV      #IR,E.CS2        ;LOAD EXPECTED COMMAND AND STATUS REG.2
5419 035116 012737 120401 004172      MOV      #SVAL!DRA!PIP!DDT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5420 035124 012737 000000 004174      MOV      #0,E.ER          ;LOAD EXPECTED ERROR REG
5421 035132 023737 004160 004120      CMP      E.CS1,T.CS1      ;CHECK COMMAND AND STATUS REG1 CORRECT
5422 035140 001401          BEQ     4$                ;YES, CHECK CS2
5423 035142 104230          ERROR   230               ;CS1 INCORRECT
5424 035144 023737 004170 004130 4$:  CMP      E.CS2,T.CS2      ;CHECK COMMAND AND STATUS REG2 CORRECT
5425 035152 001401          BEQ     5$                ;YES, CHECK DRIVE STATUS REG.
5426 035154 104231          ERROR   231               ;CS2 INCORRECT
5427 035156 023737 004172 004132 5$:  CMP      E.DS,T.DS        ;CHECK DRIVE STATUS REG. CORRECT
5428 035164 001401          BEQ     6$                ;YES, CHECK ERROR REG
5429 035166 104232          ERROR   232               ;DRIVE STATUS REG. INCORRECT
5430 035170 023737 004174 004134 6$:  CMP      E.ER,T.ER        ;CHECK ERROR REG. CORRECT
5431 035176 001401          BEQ     7$                ;YES, CHECK CONTROLLER CLEAR
5432 035200 104233          ERROR   233               ;ERROR REG. INCORRECT
5433 035202 013737 004120 004220 7$:  MOV      T.CS1,P.CS1      ;STORE PREVIOUS VALUES OF
5434 035210 013737 004130 004222          MOV      T.CS2,P.CS2      ;COMMAND AND STATUS REG.1
5435 035216 013737 004132 004224          MOV      T.DS,P.DS        ;COMMAND AND STATUS REG.2

```

```

5436 035224 013737 004134 004226      MOV      T.ER,P.ER      ; DRIVE STATUS REG.
5437                                     ; ERROR REG.
5438 035232 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ; ISSUE CONTROLLER CLEAR
5439 035240 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ; STORE COMMAND AND STATUS REG.1
5440 035246 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ; STORE COMMAND AND STATUS REG.2
5441 035254 016237 000012 004132      MOV      RKDS(R2),T.DS   ; STORE DRIVE STATUS REG.
5442 035262 016237 000014 004134      MOV      RKER(R2),T.ER   ; STORE ERROR REG
5443 035270 012737 000200 004160      MOV      #RDY,E.CS1     ; LOAD EXPECTED CS1
5444 035276 012737 000100 004170      MOV      #IR,E.CS2      ; LOAD EXPECTED CS2
5445 035304 005037 004172      CLR      E.DS           ; LOAD EXPECTED DRIVE STATUS REG.
5446 035310 005037 004174      CLR      E.ER           ; LOAD EXPECTED ERROR REG.
5447 035314 023737 004160 004120      CMP      E.CS1,T.CS1    ; CHECK COMMAND AND STATUS REG.1 CORRECT
5448 035322 001401      BEQ      10$           ; YES, CHECK CS2
5449 035324 104224      ERROR   224           ; CS1 INCORRECT
5450 035326 023737 004170 004130 10$:    CMP      E.CS2,T.CS2    ; CHECK COMMAND AND STATUS REG2 CORRECT
5451 035334 001401      BEQ      11$           ; YES, CHECK DRIVE STATUS REG
5452 035336 104225      ERROR   225           ; CS2 INCORRECT
5453 035340 023737 004172 004132 11$:    CMP      E.DS,T.DS      ; CHECK DRIVE STATUS REG CORRECT
5454 035346 001401      BEQ      12$           ; YES, CHECK ERROR REGISTER
5455 035350 104226      ERROR   226           ; DRIVE STATUS REG INCORRECT
5456 035352 023737 004174 004134 12$:    CMP      E.ER,T.ER      ; CHECK ERROR REG CORRECT
5457 035360 001401      BEQ      TST70         ; YES, GO ON TO NEXT TEST
5458 035362 104227      ERROR   227           ; ERROR REG. INCORRECT

```

```

*****
*TEST 70      IDAE DETECTION IN RK611 CONTROLLER (PART 3)

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
* RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A SEEK
* TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 2,
* HEAD 3, DRIVE 0.  CLOCK IN DIAGNOSTIC MODE UNTIL
* PHASE ADDRESS 6.  TURN OFF DIAGNOSTIC MODE AND MAKE
* SURE DRIVE AVAILABLE, DRIVE OFF TRACK, SPEED LOSS,
* ILLEGAL DISK ADDRESS ERROR, AND CONTROLLER ERROR ARE
* SET.

```

```

*****
TST70: SCOPE

```

```

5473 035364 000004      MOV      #100.,$TIMES   ; DO 100. ITERATIONS
5474 035366 012737 000144 001200      MOV      $BASE,R2      ; LOAD RK611 BASE
5475 035374 013702 001270      MOV      #SCLR,RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
5476 035400 012762 000040 000010      MOV      #DMD,RKMR1(R2) ; PUT RK611 IN MAINTENANCE MODE
5477 035406 012762 000040 000026      MOV      #2,RKDCYL(R2) ; LOAD CYLINDER ADDRESS
5478 035414 012762 000002 000020      MOV      #2,CYLIN      ;
5479 035422 012737 000002 004252      MOV      #3,HDCODE     ; LOAD HEAD ADDRESS
5480 035430 012737 000003 004250      CLR      -(SP)
5481 035436 005046      MOV      HDCODE,1(SP)
5482 035440 113766 004250 000001      MOV      (SP)+,RKDA(R2)
5483 035446 012662 000006      MOV      #6,DRVTYPE    ; LOAD DRIVE TYPE FOR PRINT OUT
5484 035452 012737 000006 004266      MOV      #SEEK,RKCS1(R2) ; ISSUE SEEK TO RK06
5485 035460 012762 000017 000000      MOV      #22,*4+2,RO   ; ISSUE CLOCK TO GET THROUGH PHASE 6
5486 035466 012700 000132      MOV      #DMD!MCLK,RKMR1(R2)
5487 035472 012762 000440 000026 1$:    MOV      #DMD,RKMR1(R2)
5488 035500 012762 000040 000026      DEC      RO
5489 035506 005300      BNE     1$
5490 035510 001370      CLR     RKMR1(R2)     ; ALLOW COMMAND TO FINISH
5491 035512 005062 000026

```

```

5492 035516 013700 004262      MOV      WAITIM,RO      ;LOAD WAIT TIME
5493 035522 105762 000000      2$:     TSTB      RKCS1(R2)      ;WAIT FOR READY
5494 035526 100402      BMI      3$
5495 035530 005300      DEC      RO
5496 035532 001373      BNE      2$
5497 035534 016237 000000 004120 3$:     MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5498 035542 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5499 035550 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
5500 035556 016237 000014 004134      MOV      RKER(R2),T.ER   ;STORE ERROR REG
5501 035564 012737 100216 004160      MOV      #CERR!RDY!<SEEK&T<C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5502
5503 035572 012737 000100 004170      MOV      #IR,E.CS2      ;LOAD EXPECTED COMMAND AND STATUS REG.2
5504 035600 012737 100061 004172      MOV      #SVAL!DRA!DROT!SPDLSS,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5505 035606 012737 002000 004174      MOV      #IDAE,E.ER     ;LOAD EXPECTED ERROR REG
5506 035614 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG1 CORRECT
5507 035622 001401      BEQ      4$            ;YES, CHECK CS2
5508 035624 104230      ERROR    230          ;CS1 INCORRECT
5509 035626 023737 004170 004130 4$:     CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5510 035634 001401      BEQ      5$            ;YES, CHECK DRIVE STATUS REG.
5511 035636 104231      ERROR    231          ;CS2 INCORRECT
5512 035640 023737 004172 004132 5$:     CMP      E.DS,T.DS     ;CHECK DRIVE STATUS REG. CORRECT
5513 035646 001401      BEQ      6$            ;YES, CHECK ERROR REG
5514 035650 104232      ERROR    232          ;DRIVE STATUS REG. INCORRECT
5515 035652 023737 004174 004134 6$:     CMP      E.ER,T.ER     ;CHECK ERROR REG. CORRECT
5516 035660 001401      BEQ      7$            ;YES, CHECK CONTROLLER CLEAR
5517 035662 104233      ERROR    233          ;ERROR REG. INCORRECT
5518 035664 013737 004120 004220 7$:     MOV      T.CS1,P.CS1    ;STORE PREVIOUS VALUES OF
5519 035672 013737 004130 004222      MOV      T.CS2,P.CS2    ;COMMAND AND STATUS REG.1
5520 035700 013737 004132 004224      MOV      T.DS,P.DS     ;COMMAND AND STATUS REG.2
5521 035706 013737 004134 004226      MOV      T.ER,P.ER     ;DRIVE STATUS REG.
5522                                     ;ERROR REG.
5523 035714 012762 100000 000000      MOV      #CLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
5524 035722 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5525 035730 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5526 035736 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG.
5527 035744 016237 000014 004134      MOV      RKER(R2),T.ER   ;STORE ERROR REG
5528 035752 012737 000200 004160      MOV      #RDY,E.CS1     ;LOAD EXPECTED CS1
5529 035760 012737 000100 004170      MOV      #IR,E.CS2     ;LOAD EXPECTED CS2
5530 035766 005037 004172      CLR      E.DS          ;LOAD EXPECTED DRIVE STATUS REG.
5531 035772 005037 004174      CLR      E.ER          ;LOAD EXPECTED ERROR REG.
5532 035776 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG.1 CORRECT
5533 036004 001401      BEQ      10$           ;YES, CHECK CS2
5534 036006 104224      ERROR    224          ;CS1 INCORRECT
5535 036010 023737 004170 004130 10$:    CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5536 036016 001401      BEQ      11$           ;YES, CHECK DRIVE STATUS REG
5537 036020 104225      ERROR    225          ;CS2 INCORRECT
5538 036022 023737 004172 004132 11$:    CMP      E.DS,T.DS     ;CHECK DRIVE STATUS REG CORRECT
5539 036030 001401      BEQ      12$           ;YES, CHECK ERROR REGISTER
5540 036032 104226      ERROR    226          ;DRIVE STATUS REG INCORRECT
5541 036034 023737 004174 004134 12$:    CMP      E.ER,T.ER     ;CHECK ERROR REG CORRECT
5542 036042 001401      BEQ      TST71         ;YES, GO ON TO NEXT TEST
5543 036044 104227      ERROR    227          ;ERROR REG. INCORRECT
5544
5545                                     ;*****
5546                                     ;*TEST 71      IDAE DETECTION IN RK611 CONTROLLER (PART 4)
5547                                     ;*

```

G10

```

5548      *      CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
5549      *      RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
5550      *      WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 23, HEAD 3,
5551      *      DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
5552      *      TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE AVAILABLE,
5553      *      DRIVE OFF TRACK, AND SPEED LOSS ARE SET WITH ILLEGAL DISK
5554      *      ADDRESS ERROR RESET.
5555      *
5556      * *****
5557 036046 000004          TST71: SCOPE
5558 036050 012737 000144 001200      MOV      #100, $TIMES      ;; DO 100. ITERATIONS
5559 036056 013702 001270          MOV      $BASE, R2        ;; LOAD RK611 BASE
5560 036062 012762 000040 000010      MOV      #SCLR, RKCS2(R2) ;; CLEAR RK06 SUBSYSTEM
5561 036070 012762 000040 000026      MOV      #DMD, RKMR1(R2) ;; PUT RK611 IN MAINTENANCE MODE
5562 036076 012762 000023 000020      MOV      #23, RKDCYL(R2) ;; LOAD CYLINDER ADDRESS
5563 036104 012737 000023 004252      MOV      #23, CYLIN
5564 036112 012737 000003 004250      MOV      #3, HDCODE      ;; LOAD HEAD ADDRESS
5565 036120 005046          CLR      -(SP)
5566 036122 113766 004250 000001      MOVB    HDCODE, 1(SP)
5567 036130 012662 000006          MOV      (SP)+, RKDA(R2)
5568 036134 012737 000007 004266      MOV      #7, DRVTYP      ;; LOAD DRIVE TYPE FOR PRINT OUT
5569 036142 012762 002017 000000      MOV      #CDT!SEEK, RKCS1(R2) ;; ISSUE SEEK TO RK06
5570 036150 012700 000132          MOV      #22, *4+2, RD    ;; ISSUE CLOCK TO GET THROUGH PHASE 6
5571 036154 012762 000440 000026 1$:  MOV      #DMD!MCLK, RKMR1(R2)
5572 036162 012762 000040 000026      MOV      #DMD, RKMR1(R2)
5573 036170 005300          DEC     RD
5574 036172 001370          BNE    1$
5575 036174 005062 000026          CLR     RKMR1(R2)      ;; ALLOW COMMAND TO FINISH
5576 036200 013700 004262          MOV     WAITIM, RD     ;; LOAD WAIT TIME
5577 036204 105762 000000          TSTB   RKCS1(R2)      ;; WAIT FOR READY
5578 036210 100402          BMI    3$
5579 036212 005300          DEC     RD
5580 036214 001373          BNE    2$
5581 036216 016237 000000 004120 3$:  MOV     RKCS1(R2), T.CS1 ;; STORE COMMAND AND STATUS REG.1
5582 036224 016237 000010 004130      MOV     RKCS2(R2), T.CS2 ;; STORE COMMAND AND STATUS REG.2
5583 036232 016237 000012 004132      MOV     RKDS(R2), T.DS  ;; STORE DRIVE STATUS REG
5584 036240 016237 000014 004134      MOV     RKER(R2), T.ER  ;; STORE ERROR REG
5585 036246 012737 002216 004160      MOV     #CDT!RDY!<SEEK&↑C<GO>>, E.CS1 ;; LOAD EXPECTED CS1
5586
5587 036254 012737 000100 004170      MOV     #IR, E.CS2     ;; LOAD EXPECTED COMMAND AND STATUS REG.2
5588 036262 012737 100461 004172      MOV     #SVAL!DRA!DROT!SPDLSS!DDT, E.DS ;; LOAD EXPECTED DRIVE STATUS REG
5589 036270 012737 000000 004174      MOV     #0, E.ER      ;; LOAD EXPECTED ERROR REG
5590 036276 023737 004160 004120      CMP     E.CS1, T.CS1   ;; CHECK COMMAND AND STATUS REG1 CORRECT
5591 036304 001401          BEQ    4$             ;; YES, CHECK CS2
5592 036306 104230          ERROR  230           ;; CS1 INCORRECT
5593 036310 023737 004170 004130 4$:  CMP     E.CS2, T.CS2   ;; CHECK COMMAND AND STATUS REG2 CORRECT
5594 036316 001401          BEQ    5$             ;; YES, CHECK DRIVE STATUS REG.
5595 036320 104231          ERROR  231           ;; CS2 INCORRECT
5596 036322 023737 004172 004132 5$:  CMP     E.DS, T.DS    ;; CHECK DRIVE STATUS REG. CORRECT
5597 036330 001401          BEQ    6$             ;; YES, CHECK ERROR REG
5598 036332 104232          ERROR  232           ;; DRIVE STATUS REG. INCORRECT
5599 036334 023737 004174 004134 6$:  CMP     E.ER, T.ER    ;; CHECK ERROR REG. CORRECT
5600 036342 001401          BEQ    7$             ;; YES, CHECK CONTROLLER CLEAR
5601 036344 104233          ERROR  233           ;; ERROR REG. INCORRECT
5602 036346 013737 004120 004220 7$:  MOV     T.CS1, P.CS1   ;; STORE PREVIOUS VALUES OF
5603 036354 013737 004130 004222      MOV     T.CS2, P.CS2   ;; COMMAND AND STATUS REG.1

```

```

5604 036362 013737 004132 004224      MOV      T.DS,P.DS      ; COMMAND AND STATUS REG.2
5605 036370 013737 004134 004226      MOV      T.ER,P.ER      ; DRIVE STATUS REG.
5606                                     ; ERROR REG.
5607 036376 012762 100000 000000      MOV      #CLR,RKCS1(R2) ; ISSUE CONTROLLER CLEAR
5608 036404 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ; STORE COMMAND AND STATUS REG.1
5609 036412 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ; STORE COMMAND AND STATUS REG.2
5610 036420 016237 000012 004132      MOV      RKDS(R2),T.DS   ; STORE DRIVE STATUS REG.
5611 036426 016237 000014 004134      MOV      RKER(R2),T.ER   ; STORE ERROR REG
5612 036434 012737 000200 004160      MOV      #RDY,E.CS1     ; LOAD EXPECTED CS1
5613 036442 012737 000100 004170      MOV      #IR,E.CS2     ; LOAD EXPECTED CS2
5614 036450 005037 004172             CLR      E.DS           ; LOAD EXPECTED DRIVE STATUS REG.
5615 036454 005037 004174             CLR      E.ER           ; LOAD EXPECTED ERROR REG.
5616 036460 023737 004160 004120      CMP      E.CS1,T.CS1    ; CHECK COMMAND AND STATUS REG.1 CORRECT
5617 036466 001401             BEQ      10$            ; YES, CHECK CS2
5618 036470 104224             ERROR    224           ; CS1 INCORRECT
5619 036472 023737 004170 004130 10$:    CMP      E.CS2,T.CS2    ; CHECK COMMAND AND STATUS REG2 CORRECT
5620 036500 001401             BEQ      11$            ; YES, CHECK DRIVE STATUS REG
5621 036502 104225             ERROR    225           ; CS2 INCORRECT
5622 036504 023737 004172 004132 11$:    CMP      E.DS,T.DS     ; CHECK DRIVE STATUS REG CORRECT
5623 036512 001401             BEQ      12$            ; YES, CHECK ERROR REGISTER
5624 036514 104226             ERROR    226           ; DRIVE STATUS REG INCORRECT
5625 036516 023737 004174 004134 12$:    CMP      E.ER,T.ER     ; CHECK ERROR REG CORRECT
5626 036524 001401             BEQ      TST72         ; YES, GO ON TO NEXT TEST
5627 036526 104227             ERROR    227           ; ERROR REG. INCORRECT

```

```

*****
*TEST 72      IDAE DETECTION IN RK611 CONTROLLER (PART 5)

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
* PUT RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A
* SEEK WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 22,
* HEAD 4, DRIVE D.  CLOCK IN DIAGNOSTIC MODE UNTIL
* PHASE ADDRESS 6.  TURN OFF DIAGNOSTIC MODE AND
* MAKE SURE DRIVE AVAILABLE AND UNSAFE ARE SET WITH
* ILLEGAL DISK ADDRESS ERROR RESET.

```

```

*****
TST72: SCOPE

```

```

5641 036530 000004             MOV      #100.,STIMES   ; DO 100. ITERATIONS
5642 036532 012737 000144 001200      MOV      $BASE,R2      ; LOAD RK611 BASE
5643 036540 013702 001270             MOV      #CLR,RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
5644 036544 012762 000040 000010      MOV      #DMD,RKMR1(R2) ; PUT RK611 IN MAINTENANCE MODE
5645 036552 012762 000040 000026      MOV      #22,RKDCYL(R2) ; LOAD CYLINDER ADDRESS
5646 036560 012762 000022 000020      MOV      #22,CYLIN
5647 036566 012737 000022 004252      MOV      #4,HDCODE     ; LOAD HEAD ADDRESS
5648 036574 012737 000004 004250      MOV      -(SP)
5649 036602 005046             CLRB    HDCODE,1(SP)
5650 036604 113766 004250 000001      MOV      (SP)+,RKDA(R2)
5651 036612 012662 000006             MOV      #7,DRVTYP     ; LOAD DRIVE TYPE FOR PRINT OUT
5652 036616 012737 000007 004266      MOV      #CDT!SEEK,RKCS1(R2) ; ISSUE SEEK TO RK06
5653 036624 012762 002017 000000      MOV      #22.*4+2,RO   ; ISSUE CLOCK TO GET THROUGH PHASE 6
5654 036632 012700 000132             MOV      #DMD!MCLK,RKMR1(R2)
5655 036636 012762 000440 000026 1$:    MOV      #DMD,RKMR1(R2)
5656 036644 012762 000040 000026             MOV      RO
5657 036652 005300             DEC     1$
5658 036654 001370             BNE    1$
5659 036656 005062 000026             CLR     RKMR1(R2)     ; ALLOW COMMAND TO FINISH

```

```

5660 036662 013700 004262      MOV      WAITIM,RO      ;LOAD WAIT TIME
5661 036666 105762 000000      2$:     TSTB      RKCS1(R2)      ;WAIT FOR READY
5662 036672 100402      BMI      3$
5663 036674 005300      DEC      RO
5664 036676 001373      BNE      2$
5665 036700 016237 000000 004120 3$:     MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5666 036706 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5667 036714 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
5668 036722 016237 000014 004134      MOV      RKER(R2),T.ER   ;STORE ERROR REG
5669 036730 012737 002216 004160      MOV      #CDT!RDY!<SEEK&↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5670
5671 036736 012737 000100 004170      MOV      #IR,E.CS2      ;LOAD EXPECTED COMMAND AND STATUS REG.2
5672 036744 012737 100401 004172      MOV      #SVAL!DRA!DDT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5673 036752 012737 040000 004174      MOV      #UNS,E.ER      ;LOAD EXPECTED ERROR REG
5674 036760 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG1 CORRECT
5675 036766 001401      BEQ      4$
5676 036770 104230      ERROR   230            ;CS1 INCORRECT
5677 036772 023737 004170 004130 4$:     CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5678 037000 001401      BEQ      5$
5679 037002 104231      ERROR   231            ;CS2 INCORRECT
5680 037004 023737 004172 004132 5$:     CMP      E.DS,T.DS      ;CHECK DRIVE STATUS REG. CORRECT
5681 037012 001401      BEQ      6$
5682 037014 104232      ERROR   232            ;DRIVE STATUS REG. INCORRECT
5683 037016 023737 004174 004134 6$:     CMP      E.ER,T.ER      ;CHECK ERROR REG. CORRECT
5684 037024 001401      BEQ      7$
5685 037026 104233      ERROR   233            ;YES, CHECK CONTROLLER CLEAR
5686 037030 013737 004120 004220 7$:     MOV      T.CS1,P.CS1    ;STORE PREVIOUS VALUES OF
5687 037036 013737 004130 004222      MOV      T.CS2,P.CS2    ;COMMAND AND STATUS REG.1
5688 037044 013737 004132 004224      MOV      T.DS,P.DS      ;COMMAND AND STATUS REG.2
5689 037052 013737 004134 004226      MOV      T.ER,P.ER      ;DRIVE STATUS REG.
5690      ;ERROR REG.
5691 037060 012762 100000 000000      MOV      #CLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
5692 037066 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5693 037074 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5694 037102 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG.
5695 037110 016237 000014 004134      MOV      RKER(R2),T.ER   ;STORE ERROR REG
5696 037116 012737 000200 004160      MOV      #RDY,E.CS1     ;LOAD EXPECTED CS1
5697 037124 012737 000100 004170      MOV      #IR,E.CS2     ;LOAD EXPECTED CS2
5698 037132 005037 004172      CLR      E.DS          ;LOAD EXPECTED DRIVE STATUS REG.
5699 037136 005037 004174      CLR      E.ER          ;LOAD EXPECTED ERROR REG.
5700 037142 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG.1 CORRECT
5701 037150 001401      BEQ      10$
5702 037152 104224      ERROR   224            ;YES, CHECK CS2
5703 037154 023737 004170 004130 10$:    CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5704 037162 001401      BEQ      11$
5705 037164 104225      ERROR   225            ;CS1 INCORRECT
5706 037166 023737 004172 004132 11$:    CMP      E.DS,T.DS      ;CHECK DRIVE STATUS REG CORRECT
5707 037174 001401      BEQ      12$
5708 037176 104226      ERROR   226            ;YES, CHECK ERROR REGISTER
5709 037200 023737 004174 004134 12$:    CMP      E.ER,T.ER      ;DRIVE STATUS REG INCORRECT
5710 037206 001401      BEQ      TST73         ;CHECK ERROR REG CORRECT
5711 037210 104227      ERROR   227            ;YES, GO ON TO NEXT TEST
5712      ;ERROR REG. INCORRECT

```

```

*****
; *TEST 73      IDAE DETECTION IN RK611 CONTROLLER (PART 6)
; *

```

```

5712
5713
5714
5715

```

```

5716          : * CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
5717          : * RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
5718          : * TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 3, HEAD
5719          : * 4, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE
5720          : * ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE SURE
5721          : * DRIVE AVAILABLE. UNSAFE, ILLEGAL DISK ADDRESS ERROR
5722          : * AND CONTROLLER ERROR ARE SET.
5723          : *
5724          : *
5725          : *****
5725 037212 000004 TST73: SCOPE
5726 037214 012737 000144 001200 MOV #100, $TIMES ; DO 100. ITERATIONS
5727 037222 013702 001270 MOV $BASE, R2 ; LOAD RK611 BASE
5728 037226 012762 000040 000010 MOV #SCLR, RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
5729 037234 012762 000040 000026 MOV #DMD, RKMR1(R2) ; PUT RK611 IN MAINTENANCE MODE
5730 037242 012762 000003 000020 MOV #3, RKDCYL(R2) ; LOAD CYLINDER ADDRESS
5731 037250 012737 000003 004252 MOV #3, CYLIN
5732 037256 012737 000004 004250 MOV #4, HDCODE ; LOAD HEAD ADDRESS
5733 037264 005046 CLR -(SP)
5734 037266 113766 004250 000001 MOVB HDCODE, 1(SP)
5735 037274 012662 000006 MOV (SP)+, RKDA(R2)
5736 037300 012737 000006 004266 MOV #6, DRVTYP ; LOAD DRIVE TYPE FOR PRINT OUT
5737 037306 012762 000017 000000 MOV #SEEK, RKCS1(R2) ; ISSUE SEEK TO RK06
5738 037314 012700 000132 MOV #22, *4+2, RD ; ISSUE CLOCK TO GET THROUGH PHASE 6
5739 037320 012762 000440 000026 1$: MOV #DMD!MCLK, RKMR1(R2)
5740 037326 012762 000040 000026 MOV #DMD, RKMR1(R2)
5741 037334 005300 DEC RD
5742 037336 001370 BNE 1$
5743 037340 005062 000026 CLR RKMR1(R2) ; ALLOW COMMAND TO FINISH
5744 037344 013700 004262 MOV WAITIM, RD ; LOAD WAIT TIME
5745 037350 105762 000000 2$: TSTB RKCS1(R2) ; WAIT FOR READY
5746 037354 100402 BMI 3$
5747 037356 005300 DEC RD
5748 037360 001373 BNE 2$
5749 037362 016237 000000 004120 3$: MOV RKCS1(R2), T.CS1 ; STORE COMMAND AND STATUS REG.1
5750 037370 016237 000010 004130 MOV RKCS2(R2), T.CS2 ; STORE COMMAND AND STATUS REG.2
5751 037376 016237 000012 004132 MOV RKDS(R2), T.DS ; STORE DRIVE STATUS REG
5752 037404 016237 000014 004134 MOV RKER(R2), T.ER ; STORE ERROR REG
5753 037412 012737 100216 004160 MOV #CERR!RDY!<SEEK&T.C<GO>>, E.CS1 ; LOAD EXPECTED CS1
5754
5755 037420 012737 000100 004170 MOV #IR, E.CS2 ; LOAD EXPECTED COMMAND AND STATUS REG.2
5756 037426 012737 100001 004172 MOV #SVAL!DRA, E.DS ; LOAD EXPECTED DRIVE STATUS REG
5757 037434 012737 042000 004174 MOV #UNS!IDAE, E.ER ; LOAD EXPECTED ERROR REG
5758 037442 023737 004160 004120 CMP E.CS1, T.CS1 ; CHECK COMMAND AND STATUS REG1 CORRECT
5759 037450 001401 BEQ 4$ ; YES, CHECK CS2
5760 037452 104230 ERROR 230 ; CS1 INCORRECT
5761 037454 023737 004170 004130 4$: CMP E.CS2, T.CS2 ; CHECK COMMAND AND STATUS REG2 CORRECT
5762 037462 001401 BEQ 5$ ; YES, CHECK DRIVE STATUS REG.
5763 037464 104231 ERROR 231 ; CS2 INCORRECT
5764 037466 023737 004172 004132 5$: CMP E.DS, T.DS ; CHECK DRIVE STATUS REG. CORRECT
5765 037474 001401 BEQ 6$ ; YES, CHECK ERROR REG
5766 037476 104232 ERROR 232 ; DRIVE STATUS REG. INCORRECT
5767 037500 023737 004174 004134 6$: CMP E.ER, T.ER ; CHECK ERROR REG. CORRECT
5768 037506 001401 BEQ 7$ ; YES, CHECK CONTROLLER CLEAR
5769 037510 104233 ERROR 233 ; ERROR REG. INCORRECT
5770 037512 013737 004120 004220 7$: MOV T.CS1, P.CS1 ; STORE PREVIOUS VALUES OF
5771 037520 013737 004130 004222 MOV T.CS2, P.CS2 ; COMMAND AND STATUS REG.1

```

K10

```

5772 037526 013737 004132 004224      MOV      T.DS,P.DS      ; COMMAND AND STATUS REG.2
5773 037534 013737 004134 004226      MOV      T.ER,P.ER    ; DRIVE STATUS REG.
5774                                     ; ERROR REG.
5775 037542 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ; ISSUE CONTROLLER CLEAR
5776 037550 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ; STORE COMMAND AND STATUS REG.1
5777 037556 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ; STORE COMMAND AND STATUS REG.2
5778 037564 016237 000012 004132      MOV      RKDS(R2),T.DS  ; STORE DRIVE STATUS REG.
5779 037572 016237 000014 004134      MOV      RKER(R2),T.ER  ; STORE ERROR REG
5780 037600 012737 000200 004160      MOV      #RDY,E.CS1    ; LOAD EXPECTED CS1
5781 037606 012737 000100 004170      MOV      #IR,E.CS2    ; LOAD EXPECTED CS2
5782 037614 005037 004172          CLR      E.DS         ; LOAD EXPECTED DRIVE STATUS REG.
5783 037620 005037 004174          CLR      E.ER         ; LOAD EXPECTED ERROR REG.
5784 037624 023737 004160 004120      CMP      E.CS1,T.CS1  ; CHECK COMMAND AND STATUS REG.1 CORRECT
5785 037632 001401          BEQ      10$         ; YES, CHECK CS2
5786 037634 104224          ERROR   224         ; CS1 INCORRECT
5787 037636 023737 004170 004130 10$:    CMP      E.CS2,T.CS2  ; CHECK COMMAND AND STATUS REG2 CORRECT
5788 037644 001401          BEQ      11$         ; YES, CHECK DRIVE STATUS REG
5789 037646 104225          ERROR   225         ; CS2 INCORRECT
5790 037650 023737 004172 004132 11$:    CMP      E.DS,T.DS    ; CHECK DRIVE STATUS REG CORRECT
5791 037656 001401          BEQ      12$         ; YES, CHECK ERROR REGISTER
5792 037660 104226          ERROR   226         ; DRIVE STATUS REG INCORRECT
5793 037662 023737 004174 004134 12$:    CMP      E.ER,T.ER    ; CHECK ERROR REG CORRECT
5794 037670 001401          BEQ      TST74       ; YES, GO ON TO NEXT TEST
5795 037672 104227          ERROR   227         ; ERROR REG. INCORRECT

```

```

*****
*TEST 74      IDAE DETECTION IN RK611 CONTROLLER (PART 7)
*

```

```

* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT
* RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A SEEK
* WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 23, HEAD 5,
* DRIVE 0.  CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS
* 6.  TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE
* AVAILABLE, UNSAFE, SPEED LOSS, ILLEGAL DISK ADDRESS
* ERROR, AND CONTROLLER ERROR ARE SET.
*

```

```

*****
TST74: SCOPE

```

```

5809 037674 000004      TST74:  MOV      #100,$TIMES  ;; DO 100. ITERATIONS
5810 037676 012737 000144 001200      MOV      $BASE,R2     ; LOAD RK611 BASE
5811 037704 013702 001270          MOV      #SCLR,RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
5812 037710 012762 000040 000010      MOV      #DMD,RKMR1(R2) ; PUT RK611 IN MAINTENANCE MODE
5813 037716 012762 000040 000026      MOV      #23,RKDCYL(R2) ; LOAD CYLINDER ADDRESS
5814 037724 012762 000023 000020      MOV      #23,CYLIN    ;
5815 037732 012737 000023 004252      MOV      #5,HDCODE    ; LOAD HEAD ADDRESS
5816 037740 012737 000005 004250      MOV      -(SP)        ;
5817 037746 005046          CLR      HDCODE,1(SP) ;
5818 037750 113766 004250 000001      MOV      (SP)+,RKDA(R2) ;
5819 037756 012662 000006          MOV      #7,DRVTYP   ; LOAD DRIVE TYPE FOR PRINT OUT
5820 037762 012737 000007 004266      MOV      #CDT!SEEK,RKCS1(R2) ; ISSUE SEEK TO RK06
5821 037770 012762 002017 000000      MOV      #22,*4+2,R0  ; ISSUE CLOCK TO GET THROUGH PHASE 6
5822 037776 012700 000132          MOV      #DMD!MCLK,RKMR1(R2) ;
5823 040002 012762 000440 000026 1$:    MOV      #DMD,RKMR1(R2) ;
5824 040010 012762 000040 000026      MOV      R0          ;
5825 040016 005300          DEC      R0          ;
5826 040020 001370          BNE     1$          ;
5827 040022 005062 000026          CLR      RKMR1(R2)   ; ALLOW COMMAND TO FINISH

```

```

5828 040026 013700 004262          MOV      WAITIM,RO      ;LOAD WAIT TIME
5829 040032 105762 000000          2$:     TSTB      RKCS1(R2)      ;WAIT FOR READY
5830 040036 100402          BMI      3$
5831 040040 005300          DEC      RO
5832 040042 001373          BNE      2$
5833 040044 016237 000000 004120 3$:     MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5834 040052 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5835 040060 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
5836 040066 016237 000014 004134      MOV      RKER(R2),T.ER   ;STORE ERROR REG
5837 040074 012737 102216 004160      MOV      #CERR!CDT!RDY!<SEEK&↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5838
5839 040102 012737 000100 004170      MOV      #IR,E.CS2      ;LOAD EXPECTED COMMAND AND STATUS REG.2
5840 040110 012737 100421 004172      MOV      #SVAL!DRA!SPDLSS!DDT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5841 040116 012737 042000 004174      MOV      #UNS!IDAE,E.ER  ;LOAD EXPECTED ERROR REG
5842 040124 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG1 CORRECT
5843 040132 001401          BEQ      4$
5844 040134 104230          ERROR    230           ;CS1 INCORRECT
5845 040136 023737 004170 004130 4$:     CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5846 040144 001401          BEQ      5$
5847 040146 104231          ERROR    231           ;CS2 INCORRECT
5848 040150 023737 004172 004132 5$:     CMP      E.DS,T.DS      ;CHECK DRIVE STATUS REG. CORRECT
5849 040156 001401          BEQ      6$
5850 040160 104232          ERROR    232           ;DRIVE STATUS REG. INCORRECT
5851 040162 023737 004174 004134 6$:     CMP      E.ER,T.ER      ;CHECK ERROR REG. CORRECT
5852 040170 001401          BEQ      7$
5853 040172 104233          ERROR    233           ;ERROR REG. INCORRECT
5854 040174 013737 004120 004220 7$:     MOV      T.CS1,P.CS1    ;STORE PREVIOUS VALUES OF
5855 040202 013737 004130 004222      MOV      T.CS2,P.CS2    ;COMMAND AND STATUS REG.1
5856 040210 013737 004132 004224      MOV      T.DS,P.DS      ;COMMAND AND STATUS REG.2
5857 040216 013737 004134 004226      MOV      T.ER,P.ER      ;DRIVE STATUS REG.
5858                                ;ERROR REG.
5859 040224 012762 100000 000000      MOV      #CLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
5860 040232 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5861 040240 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5862 040246 016237 000012 004132      MOV      RKDS(R2),T.DS   ;STORE DRIVE STATUS REG.
5863 040254 016237 000014 004134      MOV      RKER(R2),T.ER   ;STORE ERROR REG
5864 040262 012737 000200 004160      MOV      #RDY,E.CS1     ;LOAD EXPECTED CS1
5865 040270 012737 000100 004170      MOV      #IR,E.CS2     ;LOAD EXPECTED CS2
5866 040276 005037 004172          CLR      E.DS          ;LOAD EXPECTED DRIVE STATUS REG.
5867 040302 005037 004174          CLR      E.ER          ;LOAD EXPECTED ERROR REG.
5868 040306 023737 004160 004120      CMP      E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG.1 CORRECT
5869 040314 001401          BEQ      10$
5870 040316 104224          ERROR    224           ;CS1 INCORRECT
5871 040320 023737 004170 004130 10$:    CMP      E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5872 040326 001401          BEQ      11$
5873 040330 104225          ERROR    225           ;CS2 INCORRECT
5874 040332 023737 004172 004132 11$:    CMP      E.DS,T.DS      ;CHECK DRIVE STATUS REG CORRECT
5875 040340 001401          BEQ      12$
5876 040342 104226          ERROR    226           ;DRIVE STATUS REG INCORRECT
5877 040344 023737 004174 004134 12$:    CMP      E.ER,T.ER      ;CHECK ERROR REG CORRECT
5878 040352 001401          BEQ      TST75
5879 040354 104227          ERROR    227           ;ERROR REG. INCORRECT
5880
5881
5882
5883

```

```

*****
;TEST 75 IDAE DETECTION IN RK611 CONTROLLER (PART 8)
;

```

# M10

```

5884          ;*      CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT
5885          ;*      RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A SEEK
5886          ;*      WITH CDT SET IN 26 SECTOR FORMAT, CYLINDER 23, HEAD 6,
5887          ;*      DRIVE 0.  CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS
5888          ;*      6.  TURN OFF DIAGNOSTIC MODE AND MAKE SURE DRIVE
5889          ;*      AVAILABLE, UNSAFE, DRIVE OFF TRACK, ILLEGAL
5890          ;*      DISK ADDRESS ERROR, AND CONTROLLER CLEAR ARE SET.
5891          ;*
5892          ;*****
5893          ST75: SCOPE
5894          040356 000004          MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
5895          040360 012737 000144 001200      MOV      $BASE,R2        ;LOAD RK611 BASE
5896          040366 013702 001270          MOV      #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
5897          040372 012762 000040 000010      MOV      #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
5898          040400 012762 000040 000026      MOV      #23,RKDCYL(R2) ;LOAD CYLINDER ADDRESS
5899          040406 012762 000023 000020      MOV      #23,CYLIN
5900          040414 012737 000023 004252      MOV      #6,HDCODE      ;LOAD HEAD ADDRESS
5901          040422 012737 000006 004250      CLR      -(SP)
5902          040430 005046          MOVVB   HDCODE,1(SP)
5903          040432 113766 004250 000001      MOV      (SP)+,RKDA(R2)
5904          040440 012662 000006          MOV      #7,DRV TYP      ;LOAD DRIVE TYPE FOR PRINT OUT
5905          040444 012737 000007 004266      MOV      #CDT!SEEK,RKCS1(R2) ;ISSUE SEEK TO RK06
5906          040452 012762 002017 000000      MOV      #22,*4+2,R0    ;ISSUE CLOCK TO GET THROUGH PHASE 6
5907          040460 012700 000132          MOV      #DMD!MCLK,RKMR1(R2)
5908          040464 012762 000440 000026 1$:    MOV      #DMD,RKMR1(R2)
5909          040472 012762 000040 000026      DEC      R0
5910          040500 005300          BNE     1$
5911          040502 001370          CLR      RKMR1(R2)      ;ALLOW COMMAND TO FINISH
5912          040504 005062 000026          MOV      WAITIM,R0     ;LOAD WAIT TIME
5913          040510 013700 004262          TSTB    RKCS1(R2)     ;WAIT FOR READY
5914          040514 105762 000000 2$:
5915          040520 100402          BMI     3$
5916          040522 005300          DEC     R0
5917          040524 001373 2$:
5918          040526 016237 000000 004120 3$:    MOV      RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
5919          040534 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
5920          040542 016237 000012 004132      MOV      RKDS(R2),T.DS  ;STORE DRIVE STATUS REG
5921          040550 016237 000014 004134      MOV      RKER(R2),T.ER  ;STORE ERROR REG
5922          040556 012737 102216 004160      MOV      #CERR!CDT!RDY!<SEEK&↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5923          040564 012737 000100 004170      MOV      #IR,E.CS2     ;LOAD EXPECTED COMMAND AND STATUS REG.2
5924          040572 012737 100441 004172      MOV      #SVAL!DRA!DROT!DOT,E.DS ;LOAD EXPECTED DRIVE STATUS REG
5925          040600 012737 042000 004174      MOV      #UNS!IDAE,E.ER ;LOAD EXPECTED ERROR REG
5926          040606 023737 004160 004120      CMP     E.CS1,T.CS1    ;CHECK COMMAND AND STATUS REG1 CORRECT
5927          040614 001401          BEQ     4$             ;YES, CHECK CS2
5928          040616 104230          ERROR  230            ;CS1 INCORRECT
5929          040620 023737 004170 004130 4$:    CMP     E.CS2,T.CS2    ;CHECK COMMAND AND STATUS REG2 CORRECT
5930          040626 001401          BEQ     5$             ;YES, CHECK DRIVE STATUS REG.
5931          040630 104231          ERROR  231            ;CS2 INCORRECT
5932          040632 023737 004172 004132 5$:    CMP     E.DS,T.DS     ;CHECK DRIVE STATUS REG. CORRECT
5933          040640 001401          BEQ     6$             ;YES, CHECK ERROR REG
5934          040642 104232          ERROR  232            ;DRIVE STATUS REG. INCORRECT
5935          040644 023737 004174 004134 6$:    CMP     E.ER,T.ER     ;CHECK ERROR REG. CORRECT
5936          040652 001401          BEQ     7$             ;YES, CHECK CONTROLLER CLEAR
5937          040654 104233          ERROR  233            ;ERROR REG. INCORRECT
5938          040656 013737 004120 004220 7$:    MOV      T.CS1,P.CS1  ;STORE PREVIOUS VALUES OF
5939          040664 013737 004130 004222      MOV      T.CS2,P.CS2  ; COMMAND AND STATUS REG.1
    
```

```

5940 040672 013737 004132 004224      MOV      T.DS,P.DS      ; COMMAND AND STATUS REG.2
5941 040700 013737 004134 004226      MOV      T.ER,P.ER      ; DRIVE STATUS REG.
5942                                     ; ERROR REG.
5943 040706 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ; ISSUE CONTROLLER CLEAR
5944 040714 016237 000000 004120      MOV      RKCS1(R2),T.CS1 ; STORE COMMAND AND STATUS REG.1
5945 040722 016237 000010 004130      MOV      RKCS2(R2),T.CS2 ; STORE COMMAND AND STATUS REG.2
5946 040730 016237 000012 004132      MOV      RKDS(R2),T.DS   ; STORE DRIVE STATUS REG.
5947 040736 016237 000014 004134      MOV      RKER(R2),T.ER   ; STORE ERROR REG
5948 040744 012737 000200 004160      MOV      #RDY,E.CS1     ; LOAD EXPECTED CS1
5949 040752 012737 000100 004170      MOV      #IR,E.CS2     ; LOAD EXPECTED CS2
5950 040760 005037 004172          CLR      E.DS          ; LOAD EXPECTED DRIVE STATUS REG.
5951 040764 005037 004174          CLR      E.ER          ; LOAD EXPECTED ERROR REG.
5952 040770 023737 004160 004120      CMP      E.CS1,T.CS1    ; CHECK COMMAND AND STATUS REG.1 CORRECT
5953 040776 001401          BEQ      10$           ; YES, CHECK CS2
5954 041000 104224          ERROR    224           ; CS1 INCORRECT
5955 041002 023737 004170 004130 10$:      CMP      E.CS2,T.CS2    ; CHECK COMMAND AND STATUS REG2 CORRECT
5956 041010 001401          BEQ      11$           ; YES, CHECK DRIVE STATUS REG
5957 041012 104225          ERROR    225           ; CS2 INCORRECT
5958 041014 023737 004172 004132 11$:      CMP      E.DS,T.DS     ; CHECK DRIVE STATUS REG CORRECT
5959 041022 001401          BEQ      12$           ; YES, CHECK ERROR REGISTER
5960 041024 104226          ERROR    226           ; DRIVE STATUS REG INCORRECT
5961 041026 023737 004174 004134 12$:      CMP      E.ER,T.ER     ; CHECK ERROR REG CORRECT
5962 041034 001401          BEQ      TST76         ; YES, GO ON TO NEXT TEST
5963 041036 104227          ERROR    227           ; ERROR REG. INCORRECT

```

```

*****
*TEST 76      NON-STANDARD MESSAGE RECEIVING

```

```

*
* CLEAR RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT
* RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SEEK
* WITH CDT SET IN 24 SECTOR FORMAT, CYLINDER 1757, HEAD 7,
* DRIVE 1. CLOCK IN DIAGNOSTIC MODE UNTIL PHASE ADDRESS 6.
* TURN OFF DIAGNOSTIC MODE AND MAKE SURE NO ERRORS SET
* AND DRIVE STATUS IS NOT REPORTED. REPEAT FOR DRIVES
* 2 AND 4.

```

```

*****

```

```

5977 041040 000004      TST76:  SCOPE
5978 041042 012737 000144 001200      MOV      #100,$TIMES    ; DO 100. ITERATIONS
5979 041050 013702 001270          MOV      $BASE,R2      ; LOAD RK611 BASE
5980 041054 012737 000001 004244      MOV      #1,DRVCOD     ; LOAD INITIAL DRIVE CODE
5981 041062 012737 041070 001110      MOV      #1$,SLPERR    ; LOAD LOOP ON ERROR LOCATION FOR
5982                                     ; SUBTEST LOOP
5983
5984 041070          1$:
5985 041070 012762 000040 000010      MOV      #SCLR,RKCS2(R2) ; CLEAR RK06 SUBSYSTEM
5986 041076 012762 000040 000026      MOV      #DMD,RKMR1(R2) ; PUT RK611 IN MAINTENANCE MODE
5987 041104 012762 001757 000020      MOV      #1757,RKDCYL(R2) ; LOAD CYLINDER ADDRESS REG
5988 041112 012762 003400 000006      MOV      #3400,RKDA(R2) ; LOAD HEAD 7
5989 041120 013762 004244 000010      MOV      DRVCOD,RKCS2(R2) ; LOAD DRIVE NUMBER
5990 041126 012762 002017 000000      MOV      #CDT!SEEK,RKCS1(R2) ; ISSUE A SEEK WITH CDT SET
5991 041134 012700 000132          MOV      #22.*4+2,R0   ; ISSUE CLOCKS THROUGH PHASE 6
5992 041140 012762 000440 000026 2$:      MOV      #DMD!MCLK,RKMR1(R2)
5993 041146 012762 000040 000026      MOV      #DMD,RKMR1(R2)
5994 041154 005300          DEC      R0
5995 041156 001370          BNE     2$

```

```

5996 041160 005062 000026 CLR RKMR1(R2) ;ALLOW COMMAND TO FINISH
5997 041164 013700 004262 MOV WAITIM,RO ;LOAD WAIT TIME
5998 041170 105762 000000 3$: TSTB RKCS1(R2) ;WAIT FOR READY
5999 041174 100402 BMI 4$
6000 041176 005300 DEC RO
6001 041200 001373 BNE 3$
6002 041202 016237 000000 004120 4$: MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
6003 041210 016237 000010 004130 MOV RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
6004 041216 016237 000012 004132 MOV RKDS(R2),T.DS ;STORE DRIVE STATUS REG.
6005 041224 016237 000014 004134 MOV RKER(R2),T.ER ;STORE ERROR REG
6006 041232 012737 002216 004160 MOV #CDT!RDY!<SEEK&↑C<GO>>,E.CS1 ;LOAD EXPECTED CS1
6007 041240 013737 004244 004170 MOV DRVCOD,E.CS2 ;LOAD EXPECTED CS2
6008 041246 052737 000100 004170 BIS #IR,E.CS2
6009 041254 005037 004172 CLR E.DS ;LOAD EXPECTED DRIVE STATUS REG.
6010 041260 005037 004174 CLR E.ER ;LOAD EXPECTED ERROR REG.
6011 041264 023737 004160 004120 CMP E.CS1,T.CS1 ;CHECK COMMAND AND STATUS REG.1 CORRECT
6012 041272 001401 BEQ 5$ ;YES, CHECK CS2
6013 041274 104234 ERROR 234 ;CS1 INCORRECT
6014 041276 023737 004170 004130 5$: CMP E.CS2,T.CS2 ;CHECK COMMAND AND STATUS REG.2 CORRECT
6015 041304 001401 BEQ 6$ ;YES, CHECK DRIVE STATUS REG.
6016 041306 104235 ERROR 235 ;CS2 INCORRECT
6017 041310 023737 004172 004132 6$: CMP E.DS,T.DS ;CHECK DRIVE STATUS REG CORRECT
6018 041316 001401 BEQ 7$ ;YES, CHECK ERROR REG
6019 041320 104236 ERROR 236 ;DRIVE STATUS REG INCORRECT
6020 041322 023737 004174 004134 7$: CMP E.ER,T.ER ;CHECK IF ERROR CORRECT
6021 041330 001401 BEQ 8$ ;YES, CHECK IF LOOP ON ERROR
6022 041332 104237 ERROR 237 ;ERROR REG INCORRECT
6023 041334 104415 8$: SCOPE ;CHECK IF LOOP ON ERROR
6024 041336 006337 004244 ASL DRVCOD ;GENERATE NEXT DRIVE COME
6025 041342 032737 000010 004244 BIT #BIT3,DRVCOD ;CHECK IF FINISHED
6026 041350 001647 BEQ 1$ ;NO, TRY NEXT COME

```

```

*****
*TEST 77 DRIVE BUS PARITY ON NON-STANDARD MESSAGE
*****

```

```

*
* CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.
* PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE
* A SEEK TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 2,
* HEAD 0, DRIVE 1. CLOCK IN DIAGNOSTIC MODE UNTIL
* PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE AND MAKE
* SURE DRIVE BUS PARITY ERROR AND CONTROLLER ERROR SETS.
*

```

```

*****

```

```

6039 041352 000004 000144 001200 †ST77: SCOPE
6040 041354 012737 000144 001200 MOV #100.,$TIMES ;DO 100. ITERATIONS
6041 041362 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE
6042 041366 012762 000040 000010 MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM
6043 041374 012762 000040 000026 MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE
6044 041402 012762 000002 000020 MOV #2,RKDCYL(R2) ;LOAD CYLINDER ADDRESS REG
6045 041410 012762 000001 000010 MOV #1,RKCS2(R2) ;LOAD DRIVE NUMBER 1
6046 041416 012762 000017 000000 MOV #SEEK,RKCS1(R2) ;ISSUE SEEK
6047 041424 012700 000132 MOV #22.*4+2,RO ;ISSUE CLOCKS THROUGH PHASE 6
6048 041430 012762 000440 000026 1$: MOV #DMD!MCLK,RKMR1(R2)
6049 041436 012762 000040 000026 MOV #DMD,RKMR1(R2)
6050 041444 005300 DEC RO
6051 041446 001370 BNE 1$

```

# C11

6052	041450	005062	000026			CLR	RKMR1(R2)	:ALLOW COMMAND TO FINISH
6053	041454	013700	004262			MOV	WAITIM,RO	:LOAD WAIT TIME
6054	041460	105762	000000	3\$:		TSTB	RKCS1(R2)	:WAIT FOR READY
6055	041464	100402				BMI	4\$	
6056	041466	005300				RO		
6057	041470	001373				BNE	3\$	
6058	041472	016237	000000	004120	4\$:	MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG.1
6059	041500	016237	000010	004130		MOV	RKCS2(R2),T.CS2	:STORE COMMAND AND STATUS REG.2
6060	041506	016237	000012	004132		MOV	RKDS(R2),T.DS	:STORE DRIVE STATUS REG.
6061	041514	016237	000014	004134		MOV	RKER(R2),T.ER	:STORE ERROR REG.
6062	041522	012737	120216	004160		MOV	#CERR!SPAR!RDY!SEEK&↑C<GO>,E.CS1	:LOAD EXPECTED CS1
6063	041530	012737	000101	004170		MOV	#IR!1,E.CS2	:LOAD EXPECTED CS1
6064	041536	005037	004172			CLR	E.DS	:LOAD EXPECTED DRIVE STATUS REG.
6065	041542	005037	004174			CLR	E.ER	:LOAD EXPECTED ERROR REG.
6066	041546	023737	004160	004120		CMP	E.CS1,T.CS1	:CHECK COMMAND AND STATUS REG.1 CORRECT
6067	041554	001401				BEQ	5\$	:YES, CHECK CS2
6068	041556	104240				ERROR	240	:CS1 INCORRECT
6069	041560	023737	004170	004130	5\$:	CMP	E.CS2,T.CS2	:CHECK COMMAND AND STATUS REG.2 CORRECT
6070	041566	001401				BEQ	6\$	:YES, CHECK DRIVE STATUS REG
6071	041570	104241				ERROR	241	:CS2 INCORRECT
6072	041572	023737	004172	004132	6\$:	CMP	E.DS,T.DS	:CHECK DRIVE STATUS REG. CORRECT
6073	041600	001401				BEQ	7\$	:YES, CHECK ERROR REG.
6074	041602	104242				ERROR	242	:DRIVE STATUS REG. INCORRECT
6075	041604	023737	004174	004134	7\$:	CMP	E.ER,T.ER	:CHECK ERROR REG CORRECT
6076	041612	001401				BEQ	8\$	:YES, CLEAR RK611
6077	041614	104243				ERROR	243	:ERROR REG. INCORRECT
6078	041616	013737	004120	004220	8\$:	MOV	T.CS1,P.CS1	:STORE PREVIOUS CS1, CS2,
6079	041624	013737	004130	004222		MOV	T.CS2,P.CS2	:DRIVE STATUS REG.,
6080	041632	013737	004132	004224		MOV	T.DS,P.DS	:AND ERROR REG.
6081	041640	013737	004134	004226		MOV	T.ER,P.ER	
6082	041646	012762	100000	000000		MOV	#CLR,RKCS1(R2)	:CLEAR RK611
6083	041654	016237	000000	004120		MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG.1
6084	041662	016237	000010	004130		MOV	RKCS2(R2),T.CS2	:STORE COMMAND AND STATUS REG.2
6085	041670	016237	000012	004132		MOV	RKDS(R2),T.DS	:STORE DRIVE STATUS REG.
6086	041676	016237	000014	004134		MOV	RKER(R2),T.ER	:STORE ERROR REG.
6087	041704	012737	000200	004160		MOV	#RDY,E.CS1	:LOAD EXPECTED CS1
6088	041712	012737	000100	004170		MOV	#IR,E.CS2	:LOAD EXPECTED CS2
6089	041720	005037	004172			CLR	E.DS	:LOAD EXPECTED DRIVE STATUS REG.
6090	041724	005037	004174			CLR	E.ER	:LOAD EXPECTED ERROR REG.
6091	041730	023737	004160	004120		CMP	E.CS1,T.CS1	:CHECK COMMAND AND STATUS REG.1 CORRECT
6092	041736	001401				BEQ	10\$	:YES, CHECK CS2
6093	041740	104224				ERROR	224	:CS1 INCORRECT
6094	041742	023737	004170	004130	10\$:	CMP	E.CS2,T.CS2	:CHECK COMMAND AND STATUS REG.2 CORRECT
6095	041750	001401				BEQ	11\$	:YES, CHECK DRIVE STATUS REG
6096	041752	104225				ERROR	225	:CS2 INCORRECT
6097	041754	023737	004172	004132	11\$:	CMP	E.DS,T.DS	:CHECK DRIVE STATUS REG CORRECT
6098	041762	001401				BEQ	12\$	:YES, CHECK ERROR REG
6099	041764	104226				ERROR	226	:DRIVE STATUS REG. INCORRECT
6100	041766	023737	004174	004134	12\$:	CMP	E.ER,T.ER	:CHECK ERROR CORRECT
6101	041774	001401				BEQ	TST100	:YES, GO ON TO NEXT TEST
6102	041776	104227				ERROR	227	:ERROR REG INCORRECT

```

*****
*TEST 100      NON-EXISTENT DRIVE (DRIVE MESSAGE TIME OUT)
*
*
*      CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.

```

6103  
6104  
6105  
6106  
6107

```

6108      : *      PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE
6109      : *      A SELECT TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0,
6110      : *      HEAD 0, DRIVE 0.  CLOCK IN DIAGNOSTIC MODE UNTIL
6111      : *      PHASE ADDRESS 5.  TURN OFF DIAGNOSTIC MODE
6112      : *      AND MAKE SURE NON-EXISTENT DRIVE AND CONTROLLER
6113      : *      ERROR ARE SET.  THIS TEST CHECKS NON-EXISTENT DRIVE
6114      : *      DUE TO DRIVE MESSAGE TIME OUT.
6115      : *
6116      : *****
6117      042000 000004  TS100: SCOPE
6118      042002 012737 000144 001200  MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
6119      042010 013702 001270      MOV      $BASE,R2      ;;LOAD RK611 BASE
6120      042014 012762 000040 000010  MOV      #SCLR,RKCS2(R2) ;;CLEAR RK06 SUBSYSTEM
6121      042022 012762 000040 000026  MOV      #DMD,RKMR1(R2) ;;PUT RK611 IN MAINTENANCE MODE
6122      042030 012762 000001 000000  MOV      #SELDRV,RKCS1(R2) ;;ISSUE SELECT DRIVE
6123      042036 012700 000124      MOV      #21.*4,RO      ;;ISSUE CLOCKS THROUGH PHASE 4
6124      042042 012762 000440 000026 1$:  MOV      #DMD!MCLK,RKMR1(R2)
6125      042050 012762 000040 000026  MOV      #DMD,RKMR1(R2)
6126      042056 005300      DEC      RO
6127      042060 001370      BNE     1$
6128      042062 005062 000026      CLR     RKMR1(R2)      ;;ALLOW COMMAND TO FINISH
6129      042066 013700 004262      MOV     WAITIM,RO      ;;LOAD WAIT TIME
6130      042072 105762 000000      2$:  TSTB   RKCS1(R2)      ;;WAIT FOR READY
6131      042076 100402      BMI     3$
6132      042100 005300      DEC     RO
6133      042102 001373      BNE     2$
6134      042104 013700 004264      3$:  MOV     STALL,RO      ;;STALL 100 USEC FOR MESSAGE TIME OUT
6135      042110 005300      4$:  DEC     RO
6136      042112 001376      BNE     4$
6137      042114 016237 000000 004120  MOV     RKCS1(R2),T.CS1 ;;STORE COMMAND AND STATUS REG.1
6138      042122 016237 000010 004130  MOV     RKCS2(R2),T.CS2 ;;STORE COMMAND AND STATUS REG.2
6139      042130 016237 000012 004132  MOV     RKDS(R2),T.DS   ;;STORE DRIVE STATUS REG
6140      042136 016237 000014 004134  MOV     RKER(R2),T.ER   ;;STORE ERROR REG
6141      042144 012737 100200 004160  MOV     #CERR!RDY,E.CS1 ;;LOAD EXPECTED CS1
6142      042152 032737 020000 004120  BIT     #SPAR,T.CS1    ;;CHECK FOR BUS PARITY ERROR
6143      042160 001403      BEQ     5$
6144      042162 052737 020000 004160  BIS     #SPAR,E.CS1    ;;PUT BUS PARITY ERROR IN EXPECTED CS1
6145      042170 012737 010100 004170  5$:  MOV     #NED!IR,E.CS2 ;;LOAD EXPECTED CS2
6146      042176 012737 100000 004172  MOV     #SVAL,E.DS     ;;LOAD EXPECTED DRIVE STATUS REG.
6147      042204 005037 004174      CLR     E.ER          ;;LOAD EXPECTED ERROR REG.
6148      042210 023737 004160 004120  CMP     E.CS1,T.CS1    ;;CHECK COMMAND AND STATUS REG.1 CORRECT
6149      042216 001401      BEQ     6$
6150      042220 104244      ERROR  244          ;;YES, CHECK CS2
6151      042222 023737 004170 004130  6$:  CMP     E.CS2,T.CS2    ;;CHECK COMMAND AND STATUS REG.2 CORRECT
6152      042230 001401      BEQ     7$
6153      042232 104245      ERROR  245          ;;YES, CHECK DRIVE STATUS REG
6154      042234 023737 004172 004132  7$:  CMP     E.DS,T.DS     ;;CHECK DRIVE STATUS REG CORRECT
6155      042242 001401      BEQ     8$
6156      042244 104246      ERROR  246          ;;YES, CHECK ERROR REG.
6157      042246 023737 004174 004134  8$:  CMP     E.ER,T.ER     ;;DRIVE STATUS INCORRECT
6158      042254 001401      BEQ     9$
6159      042256 104247      ERROR  247          ;;CHECK ERROR REG CORRECT
6160      042260 013737 004120 004220  9$:  MOV     T.CS1,P.CS1    ;;YES, ISSUE CONTROLLER CLEAR
6161      042266 013737 004130 004222  MOV     T.CS2,P.CS2    ;;ERROR REG INCORRECT
6162      042274 013737 004132 004224  MOV     T.DS,P.DS     ;;STORE PREVIOUS CS1,CS2
6163      042302 013737 004134 004226  MOV     T.ER,P.ER     ;;DRIVE STATUS REG.,
        ;; AND ERROR REG.

```

# E11

6164	042310	012762	100000	000000		MOV	#CLR,RKCS1(R2)	:ISSUE CONTROLLER CLEAR
6165	042316	016237	000000	004120		MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG.1
6166	042324	016237	000010	004130		MOV	RKCS2(R2),T.CS2	:STORE COMMAND AND STATUS REG.2
6167	042332	016237	000012	004132		MOV	RKDS(R2),T.DS	:STORE DRIVE STATUS REG.
6168	042340	016237	000014	004134		MOV	RKER(R2),T.ER	:STORE ERROR REG.
6169	042346	012737	000200	004160		MOV	#RDY,E.CS1	:LOAD EXPECTED CS1
6170	042354	012737	000100	004170		MOV	#IR,E.CS2	:LOAD EXPECTED CS2
6171	042362	005037	004172			CLR	E.DS	:LOAD EXPECTED DRIVE STATUS REG.
6172	042366	005037	004174			CLR	E.ER	:LOAD EXPECTED ERROR REG.
6173	042372	023737	004160	004120		CMP	E.CS1,T.CS1	:CHECK COMMAND AND STATUS REG1 CORRECT
6174	042400	001401				BEQ	10\$	:YES, CHECK CS2
6175	042402	104224				ERROR	224	:CS1 INCORRECT
6176	042404	023737	004170	004130	10\$:	CMP	E.CS2,T.CS2	:CHECK COMMAND AND STATUS REG.2 CORRECT
6177	042412	001401				BEQ	11\$	:YES, CHECK DRIVE STATUS REG.
6178	042414	104225				ERROR	225	:CS2 INCORRECT
6179	042416	023737	004172	004132	11\$:	CMP	E.DS,T.DS	:CHECK DRIVE STATUS REG CORRECT
6180	042424	001401				BEQ	12\$	:YES, CHECK ERROR REG
6181	042426	104226				ERROR	226	:DRIVE STATUS INCORRECT
6182	042430	023737	004174	004134	12\$:	CMP	E.ER,T.ER	:CHECK ERROR REG CORRECT
6183	042436	001401				BEQ	TST101	:YES, GO ON TO NEXT TEST
6184	042440	104227				ERROR	227	:ERROR MESSAGE INCORRECT

6185  
6186  
6187  
6188  
6189  
6190  
6191  
6192  
6193  
6194  
6195  
6196  
6197  
6198  
6199  
6200

```
*****  
*TEST 101 NON-EXISTENT DRIVE AND NO SACK  
*  
* CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR. PUT  
* THE RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A  
* SELECT TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0,  
* HEAD 0, DRIVE 0. CLOCK IN DIAGNOSTIC MODE UNTIL  
* PHASE ADDRESS 4. TURN OFF DIAGNOSTIC MODE AND MAKE SURE  
* NON-EXISTENT DRIVE AND CONTROLLER ERROR ARE SET.  
*  
* THIS TEST EXERCISES THE NON-EXISTENT DRIVE LOGIC  
* DUE TO RELEASE BIT RESET AND SACK RESET BUT THE PASSING  
* OF THIS TEST DOES GUARENTEE THAT THIS SITUATION DID  
* INDEED CAUSE A NON-EXISTENT DRIVE.  
*  
*****
```

6201  
6202  
6203  
6204  
6205  
6206  
6207  
6208  
6209  
6210  
6211  
6212  
6213  
6214  
6215  
6216  
6217  
6218  
6219

```
*****  
*TST101: SCOPE  
*  
* MOV #100,$TIMES ;DO 100. ITERATIONS  
* MOV $BASE,R2 ;LOAD RK611 BASE  
* MOV #SCLR,RKCS2(R2) ;CLEAR RK06 SUBSYSTEM  
* MOV #DMD,RKMR1(R2) ;PUT RK611 IN MAINTENANCE MODE  
* MOV #SELDRV,RKCS1(R2) ;ISSUE SELECT DRIVE  
* MOV #19,*4+2,R0 ;ISSUE CLOCKS THROUGH PHASE 3  
* MOV #DMD!MCLK,RKMR1(R2)  
* MOV #DMD,RKMR1(R2)  
* DEC R0  
* BNE 1$  
* CLR RKMR1(R2) ;ALLOW COMMAND TO FINISH  
* MOV WAITIM,R0 ;LOAD WAIT TIME  
* TSTB RKCS1(R2) ;WAIT FOR READY  
* BMI 4$  
* DEC R0  
* BNE 3$  
* MOV RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
```

F11

6220	042554	016237	000010	004130		MOV	RKCS2(R2), T.CS2	:STORE COMMAND AND STATUS REG.2
6221	042562	016237	000012	004132		MOV	RKDS(R2), T.DS	:STORE DRIVE STATUS REG
6222	042570	016237	000014	004134		MOV	RKER(R2), T.ER	:STORE ERROR REG
6223	042576	012737	100200	004160		MOV	#CERR!RDY, E.CS1	:LOAD EXPECTED CS1
6224	042604	012737	010100	004170		MOV	#NED!IR, E.CS2	:LOAD EXPECTED CS2
6225	042612	005037	004172			CLR	E.DS	:LOAD EXPECTED DRIVE STATUS REG.
6226	042616	005037	004174			CLR	E.ER	:LOAD EXPECTED ERROR REG.
6227	042622	023737	004160	004120		CMP	E.CS1, T.CS1	:CHECK COMMAND AND STATUS REG.1 CORRECT
6228	042630	001401				BEQ	5\$	:YES, CHECK CS2
6229	042632	104250				ERROR	250	:CS1 INCORRECT
6230	042634	023737	004170	004130	5\$:	CMP	E.CS2, T.CS2	:CHECK COMMAND AND STATUS REG.2 CORRECT
6231	042642	001401				BEQ	6\$	:YES, CHECK DRIVE STATUS REG
6232	042644	104251				ERROR	251	:CS2 INCORRECT
6233	042646	023737	004172	004132	6\$:	CMP	E.DS, T.DS	:CHECK DRIVE STATUS REG CORRECT
6234	042654	001401				BEQ	7\$	:YES, CHECK ERROR REG.
6235	042656	104252				ERROR	252	:DRIVE STATUS INCORRECT
6236	042660	023737	004174	004134	7\$:	CMP	E.ER, T.ER	:CHECK ERROR REG CORRECT
6237	042666	001401				BEQ	8\$	:YES, ISSUE CONTROLLER CLEAR
6238	042670	104253				ERROR	253	:ERROR REG INCORRECT
6239	042672	013737	004120	004220	8\$:	MOV	T.CS1, P.CS1	:STORE PREVIOUS CS1, CS2
6240	042700	013737	004130	004222		MOV	T.CS2, P.CS2	:DRIVE STATUS REG.,
6241	042706	013737	004132	004224		MOV	T.DS, P.DS	:AND ERROR REG.
6242	042714	013737	004134	004226		MOV	T.ER, P.ER	
6243	042722	012762	100000	000000		MOV	#CLR, RKCS1(R2)	:ISSUE CONTROLLER CLEAR
6244	042730	016237	000000	004120		MOV	RKCS1(R2), T.CS1	:STORE COMMAND AND STATUS REG.1
6245	042736	016237	000010	004130		MOV	RKCS2(R2), T.CS2	:STORE COMMAND AND STATUS REG.2
6246	042744	016237	000012	004132		MOV	RKDS(R2), T.DS	:STORE DRIVE STATUS REG.
6247	042752	016237	000014	004134		MOV	RKER(R2), T.ER	:STORE ERROR REG.
6248	042760	012737	000200	004160		MOV	#RDY, E.CS1	:LOAD EXPECTED CS1
6249	042766	012737	000100	004170		MOV	#IR, E.CS2	:LOAD EXPECTED CS2
6250	042774	005037	004172			CLR	E.DS	:LOAD EXPECTED DRIVE STATUS REG.
6251	043000	005037	004174			CLR	E.ER	:LOAD EXPECTED ERROR REG.
6252	043004	023737	004160	004120		CMP	E.CS1, T.CS1	:CHECK COMMAND AND STATUS REG1 CORRECT
6253	043012	001401				BEQ	10\$	:YES, CHECK CS2
6254	043014	104224				ERROR	224	:CS1 INCORRECT
6255	043016	023737	004170	004130	10\$:	CMP	E.CS2, T.CS2	:CHECK COMMAND AND STATUS REG.2 CORRECT
6256	043024	001401				BEQ	11\$	:YES, CHECK DRIVE STATUS REG.
6257	043026	104225				ERROR	225	:CS2 INCORRECT
6258	043030	023737	004172	004132	11\$:	CMP	E.DS, T.DS	:CHECK DRIVE STATUS REG CORRECT
6259	043036	001401				BEQ	12\$	:YES, CHECK ERROR REG
6260	043040	104226				ERROR	226	:DRIVE STATUS INCORRECT
6261	043042	023737	004174	004134	12\$:	CMP	E.ER, T.ER	:CHECK ERROR REG CORRECT
6262	043050	001401				BEQ	TST102	:YES, GO ON TO NEXT TEST
6263	043052	104227				ERROR	227	:ERROR MESSAGE INCORRECT

.SBTTL \*\*ILLEGAL FUNCTION CODE TEST

```

*****
:TEST 102      ILLEGAL FUNCTION CODE
:
:      CLEAR RK611 WITH A CONTROLLER CLEAR.  ISSUE AN ILLEGAL
:      COMMAND IN NORMAL MODE AND MAKE SURE COMMAND FINISHES
:      SETTING CONTROLLER READY WITH PROPER ERROR CONDITIONS.
:
*****
TST102: SCOPE

```

6275 043054 000004



```

6313
6314
6315
6316
6317
6318
6319
6320
6321
6322 043312
6323 043312 000004
6324 043314 005037 0011C2
6325 043320 005037 001200
6326 043324 005237 001222
6327 043330 042737 100000 001222
6328 043336 005327
6329 043340 000001
6330 043342 003063
6331 043344 012737
6332 043346 000001
6333 043350 043340
6334 043352 104401 043360
6335 043356 000407
6336
6337 043376
6338 043376 013746 001222
6339
6340 043402 104405
6341 043404 104401 043412
6342 043410 000421
6343
6344 043454
6345 043454 013746 001112
6346
6347 043460 104405
6348 043462 104401 001211
6349 043466 005037 001112
6350 043472 013700 000042
6351 043476 001405
6352 043500 000005
6353 043502 004710
6354 043504 000240
6355 043506 000240
6356 043510 000240
6357 043512
6358 043512 000137
6359 043514 005270
6360 043516 377 377 000
6361 043522
6362
6363
6364
6365 043522 012737 043574 000004
6366 043530 012737 000340 000006
6367 043536 012703 172100
6368

```

```

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

SEOP:
SCOPE
CLR $STNM ;;ZERO THE TEST NUMBER
CLR $TIMES ;;ZERO THE NUMBER OF ITERATIONS
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #10000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?

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

SENDCT: .WORD 1
SEOPCT
TYPE ,65$ ;;TYPE ASCIZ STRING
BR ,64$ ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <12><15>/END PASS #/
64$:
MOV $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
;;TYPE PASS NUMBER
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE ,67$ ;;TYPE ASCIZ STRING
BR ,66$ ;;GET OVER THE ASCIZ
;;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$:
MOV $ERTTL,-(SP) ;;SAVE $ERTTL FOR TYPEOUT
;;TOTAL NUMBER OF ERRORS
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE $CRLF ;;TYPE CARRIAGE RETURN, LINE FEED
CLR $ERTTL ;;CLEAR ERROR TOTAL
$GET42: MOV ,2#42,R0 ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
SENDAD: JSR FC,(R0) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11

$DOAGN:
JMP ,2(PC)+ ;;RETURN
$RTNAD: .WORD NEWPAS
$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
.EVEN

.SBTTL CHECK FOR MEMORY CHECK ENABLE OPTION
CHKPAR: MOV #20$,ERRVEC ;SET VECTOR FOR MEMORY PARITY CHECK
MOV #PR7,ERRVEC+2
MOV #MEMBAS,R3 ;LOAD REGISTER TO DETERMINE IF
; MEMORY CHECK ENABLE AVAILIABLE

```

```

6369 043542 012704 000020          MOV      #16.,R4          ;LOAD COUNT
6370 043546 012723 000001          MOV      #PAR.EN,(R3)+   ;EMABLE MEMORY CHECK
6371 043552 012737 043612 000114    MOV      #MEMERR,MEMVEC ;LOAD MEMORY CHECK VECTOR
6372 043560 012737 000340 000116    MOV      #PR7,MEMVEC+2
6373 043566 005304          DEC      R4              ;CHECK IF FINISHED
6374 043570 001366          BNE     16$             ;NO, SET UP NEXT MEMORY PARITY MODULE
6375 043572 000401          BR      22$            ;RESTORE TRAP VECTOR
6376
6377 043574 022626          20$:    CMP      (SP)+,(SP)+   ;ADJUST STACK
6378 043576 012737 000006 000004    22$:    MOV      #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
6379 043604 005037 000006          CLR     ERRVEC+2
6380 043610 000207          RTS     PC              ;RETURN
6381
6382          .SBTTL MEMORY CHECK ENABLE TRAP
6383
6384 043612 012737 043626 001202    MEMERR: MOV      #10$, $ESCAPE ;LOAD ESCAPE
6385 043620 011637 004272          MOV      (SP),TRAPPC    ;STORE PC
6386 043624 104262          ERROR   262            ;REPORT MEM PARITY ERROR
6387 043626 005037 001202          10$:    CLR     $ESCAPE       ;CLEAR ESCAPE
6388 043632 032777 001000 135300    BIT     #SW9,$SWR       ;CHECK IF LOOP ON ERROR
6389 043640 001001          BNE     15$            ;YES, FORCE STACK AND TRY AGAIN
6390 043642 000002          RTI
6391
6392 043644 012706 001100          15$:    MOV      #STACK,SP    ;INITIALIZE STACK
6393 043650 000177 135234          JMP     @SLPERR        ;LOOP ON ERROR
6394
6395          .SBTTL SCOPE HANDLER ROUTINE
6396
6397          ;;*****
6398          ;;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
6399          ;;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
6400          ;;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
6401          ;;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
6402          ;;*SW14=1      LOOP ON TEST
6403          ;;*SW11=1      INHIBIT ITERATIONS
6404          ;;*SW09=1      LOOP ON ERROR
6405          ;;*SW08=1      LOOP ON TEST IN SWR<7:0>
6406          ;;*CALL
6407          ;;*      SCOPE          ;;SCOPE=IOT
6408
6409          $SCOPE:
6410 043654 104407          CKSWR
6411 043656 032777 040000 135254    1$:    BIT     #BIT14,$SWR   ;;TEST FOR CHANGE IN SOFT-SWR
6412 043664 001131          BNE     $OVER          ;;LOOP ON PRESENT TEST?
6413          ;;*****START OF CODE FOR THE XOR TESTER*****
6414 043666 000416          $XTSTR: BR      6$      ;;YES IF SW14=1
6415          ;;IF RUNNING ON THE "XOR" TESTER CHANGE
6416 043670 013746 000004          MOV     @#ERRVEC,-(SP) ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
6417 043674 012737 043714 000004    MOV     #5$,@#ERRVEC   ;;SAVE THE CONTENTS OF THE ERROR VECTOR
6418 043702 005737 177060          TST    @#177060        ;;SET FOR TIMEOUT
6419 043706 012637 000004          MOV     (SP)+,@#ERRVEC ;;TIME OUT ON XOR?
6420 043712 000500          BR     $SVLAD          ;;RESTORE THE ERROR VECTOR
6421 043714 022626          5$:    CMP     (SP)+,(SP)+   ;;GO TO THE NEXT TEST
6422 043716 012637 000004    MOV     (SP)+,@#ERRVEC ;;CLEAR THE STACK AFTER A TIME OUT
6423 043722 000440          BR     7$             ;;RESTORE THE ERROR VECTOR
6424 043724          6$:    ;;*****END OF CODE FOR THE XOR TESTER*****

```

6425	043724	032777	000400	135206		BIT	#BIT08, @SWR	:: LOOP ON SPEC. TEST?
6426	043732	001421				BEQ	2\$	:: BR IF NO
6427	043734	005046				CLR	-(SP)	:: CLEAR A TEMP. LOCATION
6428	043736	117716	135176			MOVB	@SWR, (SP)	:: PICKUP THE DESIRED TEST NUMBER
6429	043742	001414				BEQ	8\$	:: BRANCH IF BAD TEST NUMBER IN SWR
6430	043744	022716	000102			CMP	#102, (SP)	:: CHECK THE NUMBER IN THE SWR
6431	043750	002411				BLT	8\$	:: BRANCH IF TEST NUMBER IS OUT OF RANGE
6432	043752	011637	001102			MOV	(SP), \$STSTM	:: UPDATE THE TEST NUMBER
6433	043756	005316				DEC	(SP)	:: BACKUP BY ONE
6434	043760	006316				ASL	(SP)	:: SCALE THE TEST NUMBER AS AN INDEX
6435	043762	062716	044166			ADD	#\$SW08TBL, (SP)	:: FORM THE ADDRESS OF TEST POINTER
6436	043766	013637	001106			MOV	@(SP)+, \$LPADR	:: SET LOOP ADDRESS TO DESIRED TEST
6437	043772	000466				BR	\$OVER	:: GO LOOP ON THE TEST
6438	043774	005726			8\$:	TST	(SP)+	:: CLEAN THE BAD TEST NUMBER OFF OF THE STACK
6439	043776	105737	001103		2\$:	TSTB	\$ERFLG	:: HAS AN ERROR OCCURRED?
6440	044002	001421				BEQ	3\$	:: BR IF NO
6441	044004	123737	001115	001103		CMPB	\$ERMAX, \$ERFLG	:: MAX. ERRORS FOR THIS TEST OCCURRED?
6442	044012	101015				BHI	3\$	:: BR IF NO
6443	044014	032777	001000	135116		BIT	#BIT09, @SWR	:: LOOP ON ERROR?
6444	044022	001404				BEQ	4\$	:: BR IF NO
6445	044024	013737	001110	001106	7\$:	MOV	\$LPERR, \$LPADR	:: SET LOOP ADDRESS TO LAST SCOPE
6446	044032	000446				BR	\$OVER	
6447	044034	105037	001103		4\$:	CLRB	\$ERFLG	:: ZERO THE ERROR FLAG
6448	044040	005037	001200			CLR	\$TIMES	:: CLEAR THE NUMBER OF ITERATIONS TO MAKE
6449	044044	000415				BR	1\$	:: ESCAPE TO THE NEXT TEST
6450	044046	032777	004000	135064	3\$:	BIT	#BIT11, @SWR	:: INHIBIT ITERATIONS?
6451	044054	001011				BNE	1\$	:: BR IF YES
6452	044056	005737	001222			TST	\$PASS	:: IF FIRST PASS OF PROGRAM
6453	044062	001406				BEQ	1\$	:: INHIBIT ITERATIONS
6454	044064	005237	001104			INC	\$ICNT	:: INCREMENT ITERATION COUNT
6455	044070	023737	001200	001104		CMP	\$TIMES, \$ICNT	:: CHECK THE NUMBER OF ITERATIONS MADE
6456	044076	002024				BGE	\$OVER	:: BR IF MORE ITERATION REQUIRED
6457	044100	012737	000001	001104	1\$:	MOV	#1, \$ICNT	:: REINITIALIZE THE ITERATION COUNTER
6458	044106	013737	044166	001200		MOV	\$MXCNT, \$TIMES	:: SET NUMBER OF ITERATIONS TO DO
6459	044114	105237	001102		\$SVLAD:	INCB	\$STSTM	:: COUNT TEST NUMBERS
6460	044120	113737	001102	001220		MOVB	\$STSTM, \$TESTN	:: SET TEST NUMBER IN APT MAILBOX
6461	044126	011637	001106			MOV	(SP), \$LPADR	:: SAVE SCOPE LOOP ADDRESS
6462	044132	011637	001110			MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS
6463	044136	005037	001202			CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
6464	044142	112737	000001	001115		MOVB	#1, \$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
6465	044150	013777	001102	134764	\$OVER:	MOV	\$STSTM, @DISPLAY	:: DISPLAY TEST NUMBER
6466	044156	013716	001106			MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS
6467	044162	000002				RTI		:: FIXES PS
6468	044164	003720			\$MXCNT:	2000.		:: MAX. NUMBER OF ITERATIONS
6469	044166				\$SW08TBL:			
6470	044166	005310				.WORD	TST1+2	:: STARTING ADDRESS OF TEST 1
6471	044170	005614				.WORD	TST2+2	:: STARTING ADDRESS OF TEST 2
6472	044172	006074				.WORD	TST3+2	:: STARTING ADDRESS OF TEST 3
6473	044174	006340				.WORD	TST4+2	:: STARTING ADDRESS OF TEST 4
6474	044176	006652				.WORD	TST5+2	:: STARTING ADDRESS OF TEST 5
6475	044200	007214				.WORD	TST6+2	:: STARTING ADDRESS OF TEST 6
6476	044202	007532				.WORD	TST7+2	:: STARTING ADDRESS OF TEST 7
6477	044204	010050				.WORD	TST10+2	:: STARTING ADDRESS OF TEST 10
6478	044206	010366				.WORD	TST11+2	:: STARTING ADDRESS OF TEST 11
6479	044210	010632				.WORD	TST12+2	:: STARTING ADDRESS OF TEST 12
6480	044212	011076				.WORD	TST13+2	:: STARTING ADDRESS OF TEST 13

6481	044214	011414	.WORD	TST14+2	:::STARTING ADDRESS OF TEST 14
6482	044216	011724	.WORD	TST15+2	:::STARTING ADDRESS OF TEST 15
6483	044220	012256	.WORD	TST16+2	:::STARTING ADDRESS OF TEST 16
6484	044222	012620	.WORD	TST17+2	:::STARTING ADDRESS OF TEST 17
6485	044224	013064	.WORD	TST20+2	:::STARTING ADDRESS OF TEST 20
6486	044226	013344	.WORD	TST21+2	:::STARTING ADDRESS OF TEST 21
6487	044230	013624	.WORD	TST22+2	:::STARTING ADDRESS OF TEST 22
6488	044232	014104	.WORD	TST23+2	:::STARTING ADDRESS OF TEST 23
6489	044234	014364	.WORD	TST24+2	:::STARTING ADDRESS OF TEST 24
6490	044236	014644	.WORD	TST25+2	:::STARTING ADDRESS OF TEST 25
6491	044240	015156	.WORD	TST26+2	:::STARTING ADDRESS OF TEST 26
6492	044242	015470	.WORD	TST27+2	:::STARTING ADDRESS OF TEST 27
6493	044244	016002	.WORD	TST30+2	:::STARTING ADDRESS OF TEST 30
6494	044246	016314	.WORD	TST31+2	:::STARTING ADDRESS OF TEST 31
6495	044250	016626	.WORD	TST32+2	:::STARTING ADDRESS OF TEST 32
6496	044252	017124	.WORD	TST33+2	:::STARTING ADDRESS OF TEST 33
6497	044254	017436	.WORD	TST34+2	:::STARTING ADDRESS OF TEST 34
6498	044256	017674	.WORD	TST35+2	:::STARTING ADDRESS OF TEST 35
6499	044260	020140	.WORD	TST36+2	:::STARTING ADDRESS OF TEST 36
6500	044262	020474	.WORD	TST37+2	:::STARTING ADDRESS OF TEST 37
6501	044264	021040	.WORD	TST40+2	:::STARTING ADDRESS OF TEST 40
6502	044266	021364	.WORD	TST41+2	:::STARTING ADDRESS OF TEST 41
6503	044270	021730	.WORD	TST42+2	:::STARTING ADDRESS OF TEST 42
6504	044272	022204	.WORD	TST43+2	:::STARTING ADDRESS OF TEST 43
6505	044274	022734	.WORD	TST44+2	:::STARTING ADDRESS OF TEST 44
6506	044276	023174	.WORD	TST45+2	:::STARTING ADDRESS OF TEST 45
6507	044300	023562	.WORD	TST46+2	:::STARTING ADDRESS OF TEST 46
6508	044302	024012	.WORD	TST47+2	:::STARTING ADDRESS OF TEST 47
6509	044304	024446	.WORD	TST50+2	:::STARTING ADDRESS OF TEST 50
6510	044306	025130	.WORD	TST51+2	:::STARTING ADDRESS OF TEST 51
6511	044310	025562	.WORD	TST52+2	:::STARTING ADDRESS OF TEST 52
6512	044312	026214	.WORD	TST53+2	:::STARTING ADDRESS OF TEST 53
6513	044314	026632	.WORD	TST54+2	:::STARTING ADDRESS OF TEST 54
6514	044316	027264	.WORD	TST55+2	:::STARTING ADDRESS OF TEST 55
6515	044320	027716	.WORD	TST56+2	:::STARTING ADDRESS OF TEST 56
6516	044322	030350	.WORD	TST57+2	:::STARTING ADDRESS OF TEST 57
6517	044324	031002	.WORD	TST60+2	:::STARTING ADDRESS OF TEST 60
6518	044326	031434	.WORD	TST61+2	:::STARTING ADDRESS OF TEST 61
6519	044330	032066	.WORD	TST62+2	:::STARTING ADDRESS OF TEST 62
6520	044332	032520	.WORD	TST63+2	:::STARTING ADDRESS OF TEST 63
6521	044334	033152	.WORD	TST64+2	:::STARTING ADDRESS OF TEST 64
6522	044336	033570	.WORD	TST65+2	:::STARTING ADDRESS OF TEST 65
6523	044340	034222	.WORD	TST66+2	:::STARTING ADDRESS OF TEST 66
6524	044342	034704	.WORD	TST67+2	:::STARTING ADDRESS OF TEST 67
6525	044344	035366	.WORD	TST70+2	:::STARTING ADDRESS OF TEST 70
6526	044346	036050	.WORD	TST71+2	:::STARTING ADDRESS OF TEST 71
6527	044350	036532	.WORD	TST72+2	:::STARTING ADDRESS OF TEST 72
6528	044352	037214	.WORD	TST73+2	:::STARTING ADDRESS OF TEST 73
6529	044354	037676	.WORD	TST74+2	:::STARTING ADDRESS OF TEST 74
6530	044356	040360	.WORD	TST75+2	:::STARTING ADDRESS OF TEST 75
6531	044360	041042	.WORD	TST76+2	:::STARTING ADDRESS OF TEST 76
6532	044362	041354	.WORD	TST77+2	:::STARTING ADDRESS OF TEST 77
6533	044364	042002	.WORD	TST100+2	:::STARTING ADDRESS OF TEST 100
6534	044366	042444	.WORD	TST101+2	:::STARTING ADDRESS OF TEST 101
6535	044370	043056	.WORD	TST102+2	:::STARTING ADDRESS OF TEST 102
6536			:::*****		

```

6537          .SBTTL LOOP ON INTERNAL ERROR
6538
6539 044372 032777 001000 134540 SCOP1$: BIT      #SW9,ASWR      ;CHECK IF LOOP ON ERROR
6540 044400 001405          BEQ      5$          ;NO, CONTINUE
6541 044402 105737 001103          TSTB   $ERFLG      ;CHECK IF ERROR OCCURRED
6542 044406 001402          BEQ      5$          ;NO, CONTINUE
6543 044410 013716 001110          MOV     $LPERR,(SP) ;LOAD ERROR RETURN
6544 044414 000002          5$: RTI          ;RETURN
6545          .SBTTL APT COMMUNICATIONS ROUTINE
6546
6547          ;*****
6548 044416 112737 000001 044662 $ATY1: MOVB   #1,$FFLG      ;;TO REPORT FATAL ERROR
6549 044424 112737 000001 044660 $ATY3: MOVB   #1,$MFLG      ;;TO TYPE A MESSAGE
6550 044432 000403          BR      $ATYC
6551 044434 112737 000001 044662 $ATY4: MOVB   #1,$FFLG      ;;TO ONLY REPORT FATAL ERROR
6552 044442          $ATYC:
6553 044442 010046          MOV     RO,-(SP)      ;;PUSH RO ON STACK
6554 044444 010146          MOV     R1,-(SP)      ;;PUSH R1 ON STACK
6555 044446 105737 044660          TSTB   $MFLG      ;;SHOULD TYPE A MESSAGE?
6556 044452 001450          BEQ     5$          ;;IF NOT: BR
6557 044454 122737 000001 001234 CMPB   #APTENV,$ENV      ;;OPERATING UNDER APT?
6558 044462 001031          BNE    3$          ;;IF NOT: BR
6559 044464 132737 000100 001235 BITB   #APTPOOL,$ENVM    ;;SHOULD SPOOL MESSAGES?
6560 044472 001425          BEQ     3$          ;;IF NOT: BR
6561 044474 017600 000004          MOV     @4(SP),RO      ;;GET MESSAGE ADDR.
6562 044500 062766 000002 000004 ADD     #2,4(SP)      ;;BUMP RETURN ADDR.
6563 044506 005737 001214          1$: TST     $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
6564 044512 001375          BNE    1$          ;;IF NOT: WAIT
6565 044514 010037 001230          MOV     RO,$MSGAD      ;;PUT ADDR IN MAILBOX
6566 044520 105720          2$: TSTB   (RO)+      ;;FIND END OF MESSAGE
6567 044522 001376          BNE    2$
6568 044524 163700 001230          SUB     $MSGAD,RO      ;;SUB START OF MESSAGE
6569 044530 006200          ASR    RO          ;;GET MESSAGE LNGTH IN WORDS
6570 044532 010037 001232          MOV     RO,$MSGGLT      ;;PUT LENGTH IN MAILBOX
6571 044536 012737 000004 001214 MOV     #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
6572 044544 000413          BR      5$
6573 044546 017637 000004 044572 3$: MOV     @4(SP),4$      ;;PUT MSG ADDR IN JSR LINKAGE
6574 044554 062766 000002 000004 ADD     #2,4(SP)      ;;BUMP RETURN ADDRESS
6575 044562 013746 177776          MOV     177776,-(SP)  ;;PUSH 177776 ON STACK
6576 044566 004737 045346          JSR    PC,$TYPE      ;;CALL TYPE MACRO
6577 044572 000000          4$: .WORD 0
6578 044574          5$:
6579 044574 105737 044662          10$: TSTB   $FFLG      ;;SHOULD REPORT FATAL ERROR?
6580 044600 001416          BEQ    12$          ;;IF NOT: BR
6581 044602 005737 001234          TST    $ENV          ;;RUNNING UNDER APT?
6582 044606 001413          BEQ    12$          ;;IF NOT: BR
6583 044610 005737 001214          11$: TST     $MSGTYPE      ;;FINISHED LAST MESSAGE?
6584 044614 001375          BNE    11$          ;;IF NOT: WAIT
6585 044616 017637 000004 001216 MOV     @4(SP),$FATAL  ;;GET ERROR #
6586 044624 062766 000002 000004 ADD     #2,4(SP)      ;;BUMP RETURN ADDR.
6587 044632 005237 001214          INC    $MSGTYPE      ;;TELL APT TO TAKE ERROR
6588 044636 105037 044662          12$: CLRB   $FFLG      ;;CLEAR FATAL FLAG
6589 044642 105037 044661          CLRB   $LFLG      ;;CLEAR LOG FLAG
6590 044646 105037 044660          CLRB   $MFLG      ;;CLEAR MESSAGE FLAG
6591 044652 012601          MOV    (SP)+,R1      ;;POP STACK INTO R1
6592 044654 012600          MOV    (SP)+,RO      ;;POP STACK INTO RO

```

```

6593 044656 000207          RTS      PC          ;;RETURN
6594 044660          000      $MFLG: .BYTE 0      ;;MESSG. FLAG
6595 044661          000      $LFLG: .BYTE 0      ;;LOG FLAG
6596 044662          000      $FFLG: .BYTE 0      ;;FATAL FLAG
6597          044664          .EVEN
6598          000200      APTSIZE=200
6599          000001      APTENV=001
6600          000100      APTSPOOL=100
6601          000040      APTCSUP=040
6602          .SBTTL  ERROR HANDLER ROUTINE
6603
6604          ;;*****
6605          ;;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
6606          ;;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
6607          ;;*AND GO TO TYPERR ON ERROR
6608          ;;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
6609          ;;*SW15=1      HALT ON ERROR
6610          ;;*SW13=1      INHIBIT ERROR TYPEOUTS
6611          ;;*SW10=1      BELL ON ERROR
6612          ;;*SW09=1      LOOP ON ERROR
6613          ;;*CALL
6614          ;;*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
6615
6616 044664          $ERROR:
6617 044664 104407          CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
6618 044666 105237 001103      7$:      INCB      $ERFLG          ;; SET THE ERROR FLAG
6619 044672 001775          BEQ      7$          ;; DON'T LET THE FLAG GO TO ZERO
6620 044674 013777 001102 134240      MOV      $TSTNM, @DISPLAY  ;; DISPLAY TEST NUMBER AND ERROR FLAG
6621 044702 032777 02000 134230      BIT      #BIT10, @SWR      ;; BELL ON ERROR?
6622 044710 001402          BEQ      1$          ;; NO - SKIP
6623 044712 104401 001204          TYPE      $BELL          ;; RING BELL
6624 044716 005237 001112      1$:      INC      $ERTTL          ;; COUNT THE NUMBER OF ERRORS
6625 044722 011637 001116          MOV      (SP), $ERRPC      ;; GET ADDRESS OF ERROR INSTRUCTION
6626 044726 162737 000002 001116          SUB      #2, $ERRPC
6627 044734 117737 134156 001114          MOV      @ERRPC, $ITEMB    ;; STRIP AND SAVE THE ERROR ITEM CODE
6628 044742 032777 020000 134170      BIT      #BIT13, @SWR      ;; SKIP TYPEOUT IF SET
6629 044750 001004          BNE      20$          ;; SKIP TYPEOUTS
6630 044752 004737 045064          JSR      PC, TYPERR        ;; GO TO USER ERROR ROUTINE
6631 044756 104401 001211          TYPE      $CRLF
6632 044762          20$:
6633 044762 122737 000001 001234          CMPB     #APTENV, $ENV      ;; RUNNING IN APT MODE
6634 044770 001007          BNE      2$          ;; NO SKIP APT ERROR REPORT
6635 044772 113737 001114 045004          MOV      $ITEMB, 21$      ;; SET ITEM NUMBER AS ERROR NUMBER
6636 045000 004737 044434          JSR      PC, $ATY4        ;; REPORT FATAL ERROR TO APT
6637 045004          000      21$:      .BYTE 0
6638 045005          000      .BYTE 0
6639 045006 000777          22$:      BR      22$          ;; APT ERROR LOOP
6640 045010 005777 134124      2$:      TST      @SWR          ;; HALT ON ERROR
6641 045014 100002          BPL          ;; SKIP IF CONTINUE
6642 045016 000000          HALT          ;; HALT ON ERROR!
6643 045020 104407          CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
6644 045022 032777 001000 134110      3$:      BIT      #BIT09, @SWR      ;; LOOP ON ERROR SWITCH SET?
6645 045030 001402          BEQ      4$          ;; BR IF NO
6646 045032 013716 001110          MOV      $LPERR, (SP)     ;; FUDGE RETURN FOR LOOPING
6647 045036 005737 001202      4$:      TST      $ESCAPE          ;; CHECK FOR AN ESCAPE ADDRESS
6648 045042 001402          BEQ      5$          ;; BR IF NONE

```

```

6649 045044 013716 001202          MOV    $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
6650 045050          5$:    CMP    #SENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
6651 045050 022737 043502 000042      BNE   6$              ;;BRANCH IF NO
6652 045056 001001          HALT                  ;;YES
6653 045060 000000          6$:    RTI              ;;RETURN
6654 045062          6$:    RTI              ;;RETURN
6655 045062 000002          RTI                  ;;RETURN
6656
6657 ;:*****
6658 ;SBTTL TYPE ERROR ROUTINE
6659 ;*ENTRY JSR PC,TYPERR
6660 ;*RETURN RTS PC
6661 ;*
6662 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
6663 ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
6664 ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
6665 ;*THE ERROR.
6666 ;:*****
6667 045064 104413          TYPERR: SAVREG
6668 045066 113700 001114      MOVB  $ITEMB,R0      ;ENTER ERROR NUMBER
6669 045072 042700 177400      BIC   #177400,R0    ;CLEAR UNUSED BITS
6670 045076 005300          DEC   R0             ;FORM INDEX FOR ERROR TABLE
6671 045100 006300          ASL  R0
6672 045102 006300          ASL  R0
6673 045104 006300          ASL  R0
6674 045106 062700 001300      1$:  ADD   #ERRTB,R0    ;FORM ADDRESS OF ERROR ENTRY
6675 045112 012037 045126      MOV   (R0)+,2$      ;GET EM POINTER
6676 045116 001404          BEQ  3$              ;BRANCH IF THERE ISN'T ONE
6677 045120 104401 001211      TYPE , $CRLF        ;TYPE CARRIAGE RETURN LINE FEED
6678 045124 104401          TYPE ;TYPE ERROR MESSAGE (EM)
6679 045126 000000          .WORD 0             ;EM POINTER GOES HERE
6680 045130 012037 045144      3$:  MOV   (R0)+,4$      ;GET DH POINTER
6681 045134 001404          BEQ  5$              ;BRANCH IF THERE ISN'T ONE
6682 045136 104401 001211      TYPE , $CRLF        ;TYPE CR-LF
6683 045142 104401          TYPE ;TYPE DATA HEADER
6684 045144 000000          .WORD 0             ;DH POINTER GOES HERE
6685 045146 012001 5$:    MOV   (R0)+,R1      ;GET DT POINTER
6686 045150 001445          BEQ  20$             ;BRANCH IF THERE ARE NONE
6687 045152 005004          CLR  R4              ;RESET INDENT SWITCH
6688 045154 012000          MOV  (R0)+,R0      ;GET DF POINTER
6689 045156 012002          MOV  (R0)+,R2      ;STORE NUMBER OF DH'S
6690 045160 104401 001211      TYPE , $CRLF        ;TYPE <CR><LF>
6691 045164 112003 10$:   MOVB  (R0)+,R3      ;GET & STORE NUMBER OF DATA WORDS
6692 045166 105720          TSTB (R0)+          ;BUMP PAST FORMAT WORD
6693 045170 005703          TST  R3              ;TEST IF ANY DATA FOR THIS HEADER
6694 045172 001416          BEQ  14$             ;NO - SKIP DATA PRINT
6695 045174 005704          TST  R4              ;CHECK FOR INDENT
6696 045176 001004          BNE  12$            ;YES, GO INDENT
6697 045200 013146 11$:   MOV   @ (R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
6698 045202 104402          TYPOC ;TYPE IT
6699 045204 005303          DEC  R3              ;MORE DATA WORDS
6700 045206 001403          BEQ  13$            ;NO-BRANCH
6701 045210 104401 051265      12$: TYPE , SPACE2    ;TYPE SEPARATORS
6702 045214 000771          BR   11$             ;LOOP
6703 045216 104401 001211      13$: TYPE , $CRLF        ;TYPE <CR><LF>
6704 045222 005710          TST  (R0)           ;CHECK IF NEXT HEADER AVAILBLE
    
```

```

6705 045224 001401 BEQ 14$ :NO, DO NOT CHANGE INDENT
6706 045226 005104 COM R4 :CHANGE INDENT
6707 045230 005302 14$: DEC R2 :MORE DH'S?
6708 045232 003414 BLE 20$ :NO-BRANCH
6709 045234 012037 045254 15$: MOV (R0)+,18$ :GET NEXT DH POINTER
6710 045240 001751 BEQ 10$ :IF NO HEADER GO GET DATA
6711 045242 005704 TST R4 :INDENT?
6712 045244 001402 BEQ 17$ :NO-BRANCH
6713 045246 104401 051265 TYPE ,SPACE2 :YES-TYPE SPACES
6714 045252 104401 17$: TYPE :TYPE DH
6715 045254 000000 18$: .WORD 0 :DH POINTER GOES HERE
6716 045256 104401 001211 TYPE ,SCRLF
6717 045262 000740 BR 10$ ;GO TYPE OUT DATA
6718 045264 104414 20$: RESREG
6719 045266 005237 004242 INC ERRCNT :INCREMENT THE ERROR COUNT
6720 045272 032777 010000 133640 BIT #SW12,SWR :CHECK IF SWITCH 12 SET
6721 045300 001421 BEQ 25$ :NO, RETURN
6722 045302 022737 000024 004242 CMP #20.,ERRCNT :CHECK IF ERROR THRESHOLD EXCEEDED
6723 045310 103015 BEQ 25$ :NO, RETURN
6724 045312 104401 051270 TYPE ,ABORT :TYPE "PROGRAM ABORTED BECAUSE ERROR
6725 : THRESHOLD EXCEEDED"
6726 045316 005737 000042 TST 42 :CHECK IF IN CHAIN MODE
6727 045322 001407 BEQ 22$ :NO, HALT PROCESSOR
6728 045324 012737 000001 043340 MOV #1,SEOPCT :FOR PASS COUNT FOR ABORT
6729 045332 012706 001100 MOV #STACK,SP :INITIALIZE STACK
6730 045336 000137 043312 JMP SEOP :BRING IN NEXT PROGRAM
6731
6732 045342 000000 22$: HALT
6733 045344 000207 25$: RTS PC
6734 :.SBTTL TYPE ROUTINE
6735
6736 :*****
6737 :*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
6738 :*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
6739 :*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
6740 :*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
6741 :*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
6742 :*
6743 :*CALL:
6744 :*1) USING A TRAP INSTRUCTION
6745 :* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
6746 :*OR
6747 :* TYPE
6748 :* MESADR
6749 :*
6750
6751 045346 105737 001157 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
6752 045352 100002 BPL 1$ ;; BR IF YES
6753 045354 000000 HALT ;; HALT HERE IF NO TERMINAL
6754 045356 000430 BR 3$ ;; LEAVE
6755 045360 010046 1$: MOV R0,-(SP) ;; SAVE R0
6756 045362 017600 000002 MOV 32(SP),R0 ;; GET ADDRESS OF ASCIZ STRING
6757 045366 122737 000001 001234 CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
6758 045374 001011 BNE 62$ ;; NO,GO CHECK FOR APT CONSOLE
6759 045376 132737 000100 001235 BITB #APTPOOL,$ENVM ;; SPOOL MESSAGE TO APT
6760 045404 001405 BEQ 62$ ;; NO,GO CHECK FOR CONSOLE

```

```

6761 045406 010037 045416      MOV      RD,61$          ;; SETUP MESSAGE ADDRESS FOR APT
6762 045412 004737 044424      JSR      PC,$ATY3       ;; SPOOL MESSAGE TO APT
6763 045416 000000                .WORD      0           ;; MESSAGE ADDRESS
6764 045420 132737 000040 001235 61$:  BITB     #APTC SUP,$ENVM ;; APT CONSOLE SUPPRESSED
6765 045426 001003                BNE      60$           ;; YES, SKIP TYPE OUT
6766 045430 112046                2$:  MOVB     (RD)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
6767 045432 001005                BNE      4$           ;; BR IF IT ISN'T THE TERMINATOR
6768 045434 005726                TST     (SP)+         ;; IF TERMINATOR POP IT OFF THE STACK
6769 045436 012600                60$:  MOV     (SP)+,RD   ;; RESTORE RD
6770 045440 062716 000002      3$:  ADD     #2,(SP)    ;; ADJUST RETURN PC
6771 045444 000002                RTI                      ;; RETURN
6772 045446 122716 000011      4$:  CMPB     #HT,(SP)    ;; BRANCH IF <HT>
6773 045452 001430                BEQ     8$           ;;
6774 045454 122716 000200      CMPB     #CRLF,(SP)    ;; BRANCH IF NOT <CRLF>
6775 045460 001006                BNE     5$           ;;
6776 045462 005726                TST     (SP)+         ;; POP <CR><LF> EQUIV
6777 045464 104401                TYPE     $CRLF        ;; TYPE A CR AND LF
6778 045466 001211                $CRLF
6779 045470 105037 045624      CLRB     $CHARCNT     ;; CLEAR CHARACTER COUNT
6780 045474 000755                BR      2$           ;; GET NEXT CHARACTER
6781 045476 004737 045560      5$:  JSR      PC,$TYPEC   ;; GO TYPE THIS CHARACTER
6782 045502 123726 001156      6$:  CMPB     $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
6783 045506 001350                BNE     2$           ;; IF NO GO GET NEXT CHAR.
6784 045510 013746 001154      MOV     $NULL,-(SP)  ;; GET # OF FILLER CHARS. NEEDED
6785                                AND THE NULL CHAR.
6786 045514 105366 000001      7$:  DECB     1(SP)      ;; DOES A NULL NEED TO BE TYPED?
6787 045520 002770                BLT     6$           ;; BR IF NO--GO POP THE NULL OFF OF STACK
6788 045522 004737 045560      JSR      PC,$TYPEC   ;; GO TYPE A NULL
6789 045526 105337 045624      DECB     $CHARCNT    ;; DO NOT COUNT AS A COUNT
6790 045532 000770                BR      7$           ;; LOOP
6791
6792                                ;HORIZONTAL TAB PROCESSOR
6793
6794 045534 112716 000040      8$:  MOVB     #' (SP)    ;; REPLACE TAB WITH SPACE
6795 045540 004737 045560      9$:  JSR      PC,$TYPEC   ;; TYPE A SPACE
6796 045544 132737 000007 045624  BITB     #7,$CHARCNT  ;; BRANCH IF NOT AT
6797 045552 001372                BNE     9$           ;; TAB STOP
6798 045554 005726                TST     (SP)+         ;; POP SPACE OFF STACK
6799 045556 000724                BR      2$           ;; GET NEXT CHARACTER
6800 045560 105777 133364      $TYPEC: TSTB     $STPS    ;; WAIT UNTIL PRINTER IS READY
6801 045564 100375                BPL     $TYPEC
6802 045566 116677 000002 133356  MOVB     2(SP),$STPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.
6803 045574 122766 000015 000002  CMPB     #CR,2(SP)   ;; IS CHARACTER A CARRIAGE RETURN?
6804 045602 001003                BNE     1$           ;; BRANCH IF NO
6805 045604 105037 045624      CLRB     $CHARCNT    ;; YES--CLEAR CHARACTER COUNT
6806 045610 000406                BR      $TYPEX
6807 045612 122766 000012 000002 1$:  CMPB     #LF,2(SP)  ;; IS CHARACTER A LINE FEED?
6808 045620 001402                BEQ     $TYPEX       ;; BRANCH IF YES
6809 045622 105227                INCB     (PC)+        ;; COUNT THE CHARACTER
6810 045624 000000      $CHARCNT: .WORD      0 ;; CHARACTER COUNT STORAGE
6811 045626 000207      $TYPEX:  RTS         PC
6812
6813                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
6814
6815                                ;*****
6816                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
    
```

6817  
6818  
6819  
6820  
6821  
6822  
6823  
6824  
6825  
6826  
6827  
6828  
6829  
6830  
6831  
6832  
6833  
6834  
6835  
6836  
6837  
6838  
6839  
6840  
6841  
6842  
6843  
6844  
6845  
6846  
6847  
6848  
6849  
6850  
6851  
6852  
6853  
6854  
6855  
6856  
6857  
6858  
6859  
6860  
6861  
6862  
6863  
6864  
6865  
6866  
6867  
6868  
6869  
6870  
6871  
6872

045630 017646 000000  
045634 116637 000001 046053  
045642 112637 046055  
045646 062716 000002  
045652 000406  
045654 112737 000001 046053  
045662 112737 000006 046055  
045670 112737 000005 046052  
045676 010346  
045700 010446  
045702 010546  
045704 113704 046055  
045710 005404  
045712 062704 000006  
045716 110437 046054  
045722 113704 046053  
045726 016605 000012  
045732 005003  
045734 006105  
045736 000404  
045740 006105  
045742 006105  
045744 006105  
045746 010503  
045750 006103  
045752 105337 046054  
045756 100016  
045760 042703 177770  
045764 001002  
045766 005704  
045770 001403  
045772 005204  
045774 052703 000060  
046000 052703 000040  
046004 110337 046050

```
;;OCTAL (ASCII) NUMBER AND TYPE IT.
;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
;*CALL:
;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;*      TYPOS    ;;CALL FOR TYPEOUT
;*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;*      .BYTE   M              ;;M=1 OR 0
;*                                  ;;1=TYPE LEADING ZEROS
;*                                  ;;0=SUPPRESS LEADING ZEROS
;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
;*STYPOS OR $TYPOC
;*CALL:
;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;*      TYPON    ;;CALL FOR TYPEOUT
;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
;*CALL:
;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;*      TYPOC    ;;CALL FOR TYPEOUT
STYPOS: MOV      3(SP),-(SP)      ;;PICKUP THE MODE
        MOV      1(SP),SOFILL    ;;LOAD ZERO FILL SWITCH
        MOV      (SP)+,SOMODE+1  ;;NUMBER OF DIGITS TO TYPE
        ADD      #2,(SP)         ;;ADJUST RETURN ADDRESS
        BR       $TYPON
STYPOC: MOV      #1,SOFILL      ;;SET THE ZERO FILL SWITCH
        MOV      #6,SOMODE+1    ;;SET FOR SIX(6) DIGITS
STYPON: MOV      #5,SOCNT      ;;SET THE ITERATION COUNT
        MOV      R3,-(SP)       ;;SAVE R3
        MOV      R4,-(SP)       ;;SAVE R4
        MOV      R5,-(SP)       ;;SAVE R5
        MOV      SOMODE+1,R4    ;;GET THE NUMBER OF DIGITS TO TYPE
        NEG      R4
        ADD      #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
        MOV      R4,SOMODE      ;;SAVE IT FOR USE
        MOV      SOFILL,R4     ;;GET THE ZERO FILL SWITCH
        MOV      12(SP),R5     ;;PICKUP THE INPUT NUMBER
        CLR      R3            ;;CLEAR THE OUTPUT WORD
1$:    ROL      R5              ;;ROTATE MSB INTO "C"
        BR       3$           ;;GO DO MSB
2$:    ROL      R5              ;;FORM THIS DIGIT
        ROL      R5
        ROL      R5
        MOV      R5,R3
3$:    ROL      R3              ;;GET LSB OF THIS DIGIT
        DECB    SOMODE         ;;TYPE THIS DIGIT?
        BPL     7$            ;;BR IF NO
        BIC     #177770,R3    ;;GET RID OF JUNK
        BNE     4$            ;;TEST FOR 0
        TST     R4            ;;SUPPRESS THIS 0?
        BEQ     5$            ;;BR IF YES
4$:    INC      R4            ;;DON'T SUPPRESS ANYMORE 0'S
        BIS     #'0,R3        ;;MAKE THIS DIGIT ASCII
5$:    BIS     #' ,R3        ;;MAKE ASCII IF NOT ALREADY
        MOV     R3,#$        ;;SAVE FOR TYPING
```

```

6873 046010 104401 046050
6874 046014 105337 046052
6875 046020 003347
6876 046022 002402
6877 046024 005204
6878 046026 000744
6879 046030 012605
6880 046032 012604
6881 046034 012603
6882 046036 016666 000002 000004
6883 046044 012616
6884 046046 000002
6885 046050 000
6886 046051 000
6887 046052 000
6888 046053 000
6889 046054 000000
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902 046056
6903 046056 010046
6904 046060 010146
6905 046062 010246
6906 046064 010346
6907 046066 010546
6908 046070 012746 020200
6909 046074 016605 000020
6910 046100 100004
6911 046102 005405
6912 046104 112766 000055 000001
6913 046112 005000
6914 046114 012703 046272
6915 046120 112723 000040
6916 046124 005002
6917 046126 016001 046262
6918 046132 160105
6919 046134 002402
6920 046136 005202
6921 046140 000774
6922 046142 060105
6923 046144 005702
6924 046146 001002
6925 046150 105716
6926 046152 100407
6927 046154 106316
6928 046156 103003

```

```

TYPE 8$
DECB $OCNT
BGT 2$
BLT 6$
INC R4
BR 2$
MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV 2(SP),4(SP)
MOV (SP)+,(SP)
RTI
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.WORD 0
.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

```

7$:
6$:
8$:
SOCNT:
SOFILL:
SOMODE:
.SBTTL

```

```

:GO TYPE THIS DIGIT
:COUNT BY 1
:BR IF MORE TO DO
:BR IF DONE
:INSURE LAST DIGIT ISN'T A BLANK
:GO DO THE LAST DIGIT
:RESTORE R5
:RESTORE R4
:RESTORE R3
:SET THE STACK FOR RETURNING
:RETURN
:STORAGE FOR ASCII DIGIT
:TERMINATOR FOR TYPE ROUTINE
:OCTAL DIGIT COUNTER
:ZERO FILL SWITCH
:NUMBER OF DIGITS TO TYPE

```

```

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

```

```

$TYPDS:
MOV R0,-(SP) ;PUSH R0 ON STACK
MOV R1,-(SP) ;PUSH R1 ON STACK
MOV R2,-(SP) ;PUSH R2 ON STACK
MOV R3,-(SP) ;PUSH R3 ON STACK
MOV R5,-(SP) ;PUSH R5 ON STACK
MOV #20200,-(SP) ;SET BLANK SWITCH AND SIGN
MOV 20(SP),R5 ;GET THE INPUT NUMBER
BPL 1$ ;BR IF INPUT IS POS.
NEG R5 ;MAKE THE BINARY NUMBER POS.
MOVB #'-,1(SP) ;MAKE THE ASCII NUMBER NEG.
CLR R0 ;ZERO THE CONSTANTS INDEX
MOV #SDBLK,R3 ;SETUP THE OUTPUT POINTER
MOVB #' ,(R3)+ ;SET THE FIRST CHARACTER TO A BLANK
CLR R2 ;CLEAR THE BCD NUMBER
MOV $DTBL(R0),R1 ;GET THE CONSTANT
SUB R1,R5 ;FORM THIS BCD DIGIT
BLT 4$ ;BR IF DONE
INC R2 ;INCREASE THE BCD DIGIT BY 1
BR 3$
ADD R1,R5 ;ADD BACK THE CONSTANT
TST R2 ;CHECK IF BCD DIGIT=0
BNE 5$ ;FALL THROUGH IF 0
TSTB (SP) ;STILL DOING LEADING 0'S?
BMI 7$ ;BR IF YES
ASLB (SP) ;MSD?
BCC 6$ ;BR IF NO

```

```

1$:
2$:
3$:
4$:
5$:

```

```

6929 046160 116663 000001 177777      MOVB    1(SP),-1(R3)      ;;YES--SET THE SIGN
6930 046166 052702 000060          BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
6931 046172 052702 000040          BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
6932 046176 110223          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
6933 046200 005720          TST     (R0)+           ;;JUST INCREMENTING
6934 046202 020027 000010          CMP     R0,#10         ;;CHECK THE TABLE INDEX
6935 046206 002746          BLT     2$              ;;GO DO THE NEXT DIGIT
6936 046210 003002          BGT     8$              ;;GO TO EXIT
6937 046212 010502          MOV     R5,R2          ;;GET THE LSD
6938 046214 000764          BR     6$              ;;GO CHANGE TO ASCII
6939 046216 105726          TSTB   (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
6940 046220 100003          BPL     9$              ;;BR IF NO
6941 046222 116663 177777 177776      MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
6942 046230 105013          CLRB   (R3)            ;;SET THE TERMINATOR
6943 046232 012605          MOV     (SP)+,R5       ;;POP STACK INTO R5
6944 046234 012603          MOV     (SP)+,R3       ;;POP STACK INTO R3
6945 046236 012602          MOV     (SP)+,R2       ;;POP STACK INTO R2
6946 046240 012601          MOV     (SP)+,R1       ;;POP STACK INTO R1
6947 046242 012600          MOV     (SP)+,R0       ;;POP STACK INTO R0
6948 046244 104401 046272          TYPE   $DBLK           ;;NOW TYPE THE NUMBER
6949 046250 016666 000002 000004      MOV     2(SP),4(SP)    ;;ADJUST THE STACK
6950 046256 012616          MOV     (SP)+,(SP)
6951 046260 000002          RTI                                ;;RETURN TO USER
6952 046262 023420          SDTBL: 10000.
6953 046264 001750          1000.
6954 046266 000144          100.
6955 046270 000012          10.
6956 046272 000004          SDBLK: .BLKW 4
6957          .SBTTL TTY INPUT ROUTINE
6958
6959          ;;*****
6960          .ENABL LSB
6961
6962          ;;*****
6963          ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
6964          ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
6965          ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
6966          ;;*WHEN OPERATING IN TTY FLAG MODE.
6967 046302 022737 000176 001140      SCKSWR: CMP     $SWREG,SWR  ;; IS THE SOFT-SWR SELECTED?
6968 046310 001074          BNE     15$            ;; BRANCH IF NO
6969 046312 105777 132626          TSTB   $STKS          ;; CHAR THERE?
6970 046316 100071          BPL     15$            ;; IF NO, DON'T WAIT AROUND
6971 046320 117746 132622          MOVB   $STKB,-(SP)    ;; SAVE THE CHAR
6972 046324 042716 177600          BIC     #'C177,(SP)   ;; STRIP-OFF THE ASCII
6973 046330 022726 000007          CMP     #'7,(SP)+     ;; IS IT A CONTROL G?
6974 046334 001062          BNE     15$            ;; NO, RETURN TO USER
6975 046336 123727 001134 000001          CMPB   $AUTOB,#1     ;; ARE WE RUNNING IN AUTO-MODE?
6976 046344 001456          BEQ     15$            ;; BRANCH IF YES
6977
6978 046346 104401 047155          TYPE   ,SCNTLG        ;; ECHO THE CONTROL-G (†G)
6979 046352 104401 047162          SGTSWR: TYPE   ,SMSWR  ;; TYPE CURRENT CONTENTS
6980 046356 013746 000176          MOV     $SWREG,-(SP)  ;; SAVE SWREG FOR TYPEOUT
6981 046362 104402          TYPOC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
6982 046364 104401 047173          TYPE   ,SMNEW         ;; PROMPT FOR NEW SWR
6983 046370 005046          19$:  CLR     -(SP)    ;; CLEAR COUNTER
6984 046372 005046          CLR     -(SP)         ;; THE NEW SWR
    
```

```

6985 046374 105777 132544      7$:  TSTB  2$TKS      ;;CHAR THERE?
6986 046400 100375              BPL  7$          ;;IF NOT TRY AGAIN
6987
6988 046402 117746 132540      MOVB  2$TKB,-(SP)  ;;PICK UP CHAR
6989 046406 042716 177600      BIC  #1C177,(SP)  ;;MAKE IT 7-BIT ASCII
6990
6991
6992
6993 046412 021627 000025      9$:  CMP  (SP),#25    ;;IS IT A CONTROL-U?
6994 046416 001005              BNE  10$         ;;BRANCH IF NOT
6995 046420 104401 047150      TYPE  ,SCNTLU     ;;YES, ECHO CONTROL-U (↑U)
6996 046424 062706 000006      20$: ADD  #6,SP      ;;IGNORE PREVIOUS INPUT
6997 046430 000757              BR   19$         ;;LET'S TRY IT AGAIN
6998
6999
7000 046432 021627 000015      10$: CMP  (SP),#15   ;;IS IT A <CR>?
7001 046436 001022              BNE  16$         ;;BRANCH IF NO
7002 046440 005766 000004      TST  4(SP)        ;;YES, IS IT THE FIRST CHAR?
7003 046444 001403              BEQ  11$         ;;BRANCH IF YES
7004 046446 016677 000002 132464      MOV  2(SP),2$SWR  ;;SAVE NEW SWR
7005 046454 062706 000006      11$: ADD  #6,SP      ;;CLEAR UP STACK
7006 046460 104401 001211      14$: TYPE  ,SCRLF   ;;ECHO <CR> AND <LF>
7007 046464 123727 001135 000001      CMPB $INTAG,#1   ;;RE-ENABLE TTY KBD INTERRUPTS?
7008 046472 001003              BNE  15$         ;;BRANCH IF NOT
7009 046474 012777 000100 132442      MOV  #100,2$TKS  ;;RE-ENABLE TTY KBD INTERRUPTS
7010 046502 000002              RTI                    ;;RETURN
7011 046504 004737 045560      16$: JSR  PC,$TYPEC  ;;ECHO CHAR
7012 046510 021627 000060      CMP  (SP),#60    ;;CHAR < 0?
7013 046514 002420              BLT  18$         ;;BRANCH IF YES
7014 046516 021627 000067      CMP  (SP),#67    ;;CHAR > 7?
7015 046522 003015              BGT  18$         ;;BRANCH IF YES
7016 046524 042726 000060      BIC  #60,(SP)+   ;;STRIP-OFF ASCII
7017 046530 005766 000002      TST  2(SP)        ;;IS THIS THE FIRST CHAR
7018 046534 001403              BEQ  17$         ;;BRANCH IF YES
7019 046536 006316              ASL  (SP)         ;;NO, SHIFT PRESENT
7020 046540 006316              ASL  (SP)         ;;CHAR OVER TO MAKE
7021 046542 006316              ASL  (SP)         ;;ROOM FOR NEW ONE.
7022 046544 005266 000002      17$: INC  2(SP)        ;;KEEP COUNT OF CHAR
7023 046550 056616 177776      BIS  -2(SP),(SP) ;;SET IN NEW CHAR
7024 046554 000737              BR   7$          ;;GET THE NEXT ONE
7025 046556 104401 001210      18$: TYPE  ,SQUES   ;;TYPE ?<CR><LF>
7026 046562 000720              BR   20$         ;;SIMULATE CONTROL-U
7027 .DSABL  LSB
7028
7029
7030 *****
7031 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7032 *CALL:
7033 *      RDCHR          ;;INPUT A SINGLE CHARACTER FROM THE TTY
7034 *      RETURN HERE   ;;CHARACTER IS ON THE STACK
7035 *                   ;;WITH PARITY BIT STRIPPED OFF
7036 *
7037
7038 046564 011646 000004 000002  $RDCHR: MOV  (SP),-(SP)  ;;PUSH DOWN THE PC
7039 046566 016666 000004 000002  MOV  4(SP),2(SP)    ;;SAVE THE PS
7040 046574 105777 132344      1$:  TSTB  2$TKS      ;;WAIT FOR

```

```

7041 046600 100375          BPL      1$          ;; A CHARACTER
7042 046602 117766 132340 000004      MOVB     2$TKB,4(SP) ;; READ THE TTY
7043 046610 042766 177600 000004      BIC      #1C<177>,4(SP) ;; GET RID OF JUNK IF ANY
7044 046616 026627 000004 000023      CMP      4(SP),#23    ;; IS IT A CONTROL-S?
7045 046624 001013          BNE      3$          ;; BRANCH IF NO
7046 046626 105777 132312          2$: TSTB     2$TKS      ;; WAIT FOR A CHARACTER
7047 046632 100375          BPL      2$          ;; LOOP UNTIL ITS THERE
7048 046634 117746 132306      MOVB     2$TKB,-(SP)  ;; GET CHARACTER
7049 046640 042716 177600      BIC      #1C177,(SP) ;; MAKE IT 7-BIT ASCII
7050 046644 022627 000021      CMP      (SP)+,#21   ;; IS IT A CONTROL-Q?
7051 046650 001366          BNE      2$          ;; IF NOT DISCARD IT
7052 046652 000750          BR       1$          ;; YES, RESUME
7053 046654 026627 000004 000140          3$: CMP      4(SP),#140 ;; IS IT UPPER CASE?
7054 046662 002407          BLT      4$          ;; BRANCH IF YES
7055 046664 026627 000004 000175          CMP      4(SP),#175  ;; IS IT A SPECIAL CHAR?
7056 046672 003003          BGT      4$          ;; BRANCH IF YES
7057 046674 042766 000040 000004          BIC      #40,4(SP)   ;; MAKE IT UPPER CASE
7058 046702 000002          4$: RTI          ;; GO BACK TO USER
7059                                     ;; *****
7060                                     ;; *THIS ROUTINE WILL INPUT A STRING FROM THE TTY *
7061                                     ;; *CALL:
7062                                     ;; *
7063                                     ;; *   RDLIN          ;; INPUT A STRING FROM THE TTY
7064                                     ;; *   RETURN HERE   ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
7065                                     ;; *
7066 046704 010346          $RDLIN: MOV      R3,-(SP)   ;; SAVE R3
7067 046706 005046          CLR      -(SP)      ;; CLEAR THE RUBOUT KEY
7068 046710 012703 047140          1$: MOV      #TTYIN,R3 ;; GET ADDRESS
7069 046714 022703 047150          2$: CMP      #TTYIN+8.,R3 ;; BUFFER FULL?
7070 046720 101456          BLOS     4$          ;; BR IF YES
7071 046722 104410          RDCHR   ;; GO READ ONE CHARACTER FROM THE TTY
7072 046724 112613          MOVB     (SP)+,(R3) ;; GET CHARACTER
7073 046726 122713 000177          10$: CMPB     #177,(R3) ;; IS IT A RUBOUT
7074 046732 001022          BNE      5$          ;; BR IF NO
7075 046734 005716          TST      (SP)        ;; IS THIS THE FIRST RUBOUT?
7076 046736 001007          BNE      6$          ;; BR IF NO
7077 046740 112737 000134 047136          MOVB     #' \,9$    ;; TYPE A BACK SLASH
7078 046746 104401 047136          TYPE     9$
7079 046752 012716 177777          MOV      #-1,(SP)   ;; SET THE RUBOUT KEY
7080 046756 005303          6$: DEC      R3        ;; BACKUP BY ONE
7081 046760 020327 047140          CMP      R3,#TTYIN  ;; STACK EMPTY?
7082 046764 103434          BLO      4$          ;; BR IF YES
7083 046766 111337 047136          MOVB     (R3),9$    ;; SETUP TO TYPEOUT THE DELETED CHAR.
7084 046772 104401 047136          TYPE     9$
7085 046776 000746          BR       2$          ;; GO TYPE
7086 047000 005716          5$: TST      (SP)        ;; GO READ ANOTHER CHAR.
7087 047002 001406          BEQ      7$          ;; RUBOUT KEY SET?
7088 047004 112737 000134 047136          MOVB     #' \,9$    ;; BR IF NO
7089 047012 104401 047136          TYPE     9$
7090 047016 005016          CLR      (SP)        ;; CLEAR THE RUBOUT KEY
7091 047020 122713 000025          7$: CMPB     #25,(R3) ;; IS CHARACTER A CTRL U?
7092 047024 001003          BNE      8$          ;; BR IF NO
7093 047026 104401 047150          TYPE     ,SCNTLU    ;; TYPE A CONTROL "U"
7094 047032 000726          BR       1$          ;; GO START OVER
7095 047034 122713 000022          8$: CMPB     #22,(R3) ;; IS CHARACTER A "r"?
7096 047040 001011          BNE      3$          ;; BRANCH IF NO
  
```

```

7097 047042 105013          CLRB      (R3)          ;; CLEAR THE CHARACTER
7098 047044 104401 001211    TYPE      , $CRLF      ;; TYPE A "CR" & "LF"
7099 047050 104401 047140    TYPE      , $TTYIN     ;; TYPE THE INPUT STRING
7100 047054 000717          BR        2$           ;; GO PICKUP ANOTHER CHAFTER
7101 047056 104401 001210    4$:      TYPE      , $QUES  ;; TYPE A '?'
7102 047062 000712          BR        1$           ;; CLEAR THE BUFFER AND LOOP
7103 047064 111337 047136    3$:      MOVVB     (R3), 9$  ;; ECHO THE CHARACTER
7104 047070 104401 047136    TYPE      , 9$
7105 047074 122723 000015    CMPB     #15, (R3)+    ;; CHECK FOR RETURN
7106 047100 001305          BNE      2$           ;; LOOP IF NOT RETURN
7107 047102 105063 177777    CLRB     -1(R3)       ;; CLEAR RETURN (THE 15)
7108 047106 104401 001212    TYPE      , $LF        ;; TYPE A LINE FEED
7109 047112 005726          TST      (SP)+        ;; CLEAN RUBOUT KEY FROM THE STACK
7110 047114 012603          MOV      (SP)+, R3    ;; RESTORE R3
7111 047116 011646          MOV      (SP), -(SP)  ;; ADJUST THE STACK AND PUT ADDRESS OF THE
7112 047120 016666 000004 000002  MOV      4(SP), 2(SP)  ;; FIRST ASCII CHARACTER ON IT
7113 047126 012766 047140 000004  MOV      # $TTYIN, 4(SP)
7114 047134 000002          RTI                    ;; RETURN
7115 047136          000          9$:      .BYTE      0          ;; STORAGE FOR ASCII CHAR. TO TYPE
7116 047137          000          .BYTE      0          ;; TERMINATOR
7117 047140 000010          $TTYIN: .BLKB      8.    ;; RESERVE 8 BYTES FOR TTY INPUT
7118 047150 052536 005015 000          $CNTLU: .ASCIZ    /TU/<15><12>  ;; CONTROL "U"
7119 047155          136 006507 000012  $CNTLG: .ASCIZ    /TG/<15><12>  ;; CONTROL "G"
7120 047162 005015 053523 020122  $MSWR:  .ASCIZ    <15><12>/SWR = /
7121 047170 020075          000
7122 047173          040 047040 053505  $MNEW:  .ASCIZ    / NEW = /
7123 047200 036440 000040
7124          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
7125
7126          ;; *****
7127          ;; *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
7128          ;; *CHANGE IT TO BINARY.
7129          ;; *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
7130          ;; *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
7131          ;; *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
7132          ;; *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
7133          ;; *CALL:
7134          ;; *      RDOCT          ;; READ AN OCTAL NUMBER
7135          ;; *      RETURN HERE    ;; LOW ORDER BITS ARE ON TOP OF THE STACK
7136          ;; *                    ;; HIGH ORDER BITS ARE IN $HIOCT
7137
7138 047204 011646          $RDOCT: MOV      (SP), -(SP)  ;; PROVIDE SPACE FOR THE
7139 047206 016666 000004 000002  MOV      4(SP), 2(SP)  ;; INPUT NUMBER
7140 047214 010046          MOV      R0, -(SP)    ;; PUSH R0 ON STACK
7141 047216 010146          MOV      R1, -(SP)    ;; PUSH R1 ON STACK
7142 047220 010246          MOV      R2, -(SP)    ;; PUSH R2 ON STACK
7143 047222 104411          1$:      RDLIN     ;; READ AN ASCIZ LINE
7144 047224 012600          MOV      (SP)+, R0    ;; GET ADDRESS OF 1ST CHARACTER
7145 047226 010037 047332    MOV      R0, 5$       ;; AND SAVE IT
7146 047232 005001          CLR      R1          ;; CLEAR DATA WORD
7147 047234 005002          CLR      R2
7148 047236 112046          2$:      MOVVB     (R0)+, -(SP)  ;; PICKUP THIS CHARACTER
7149 047240 001420          BEQ      3$           ;; IF ZERO GET OUT
7150 047242 122716 000060    CMPB     #'0, (SP)    ;; MAKE SURE THIS CHARACTER
7151 047246 003026          BGT      4$           ;; IS AN OCTAL DIGIT
7152 047250 122716 000067    CMPB     #'7, (SP)

```

```

7153 047254 002423          BLT      4$
7154 047256 006301          ASL     R1          ;;*2
7155 047260 006102          ROL    R2
7156 047262 006301          ASL     R1          ;;*4
7157 047264 006102          ROL    R2
7158 047266 006301          ASL     R1          ;;*8
7159 047270 006102          ROL    R2
7160 047272 042716 177770  BIC     #1C7,(SP)    ;;STRIP THE ASCII JUNK
7161 047276 062601          ADD    (SP)+,R1    ;;ADD IN THIS DIGIT
7162 047300 000756          BR     2$          ;;LOOP
7163 047302 005726          3$:    TST    (SP)+  ;;CLEAN TERMINATOR FROM STACK
7164 047304 010166 000012  MOV    R1,12(SP)   ;;SAVE THE RESULT
7165 047310 010237 047342  MOV    R2,$HIOCT
7166 047314 012602          MOV    (SP)+,R2   ;;POP STACK INTO R2
7167 047316 012601          MOV    (SP)+,R1   ;;POP STACK INTO R1
7168 047320 012600          MOV    (SP)+,R0   ;;POP STACK INTO R0
7169 047322 000002          RTI
7170 047324 005726          4$:    TST    (SP)+  ;;CLEAN PARTIAL FROM STACK
7171 047326 105010          CLRB   (R0)       ;;SET A TERMINATOR
7172 047330 104401          TYPE   ;;TYPE UP THRU THE BAD CHAR.
7173 047332 000000          5$:    .WORD  0
7174 047334 104401 001210  TYPE   $QUES      ;;?" "CR" & "LF"
7175 047340 000730          BR     1$          ;;TRY AGAIN
7176 047342 000000          $HIOCT: .WORD  0  ;;HIGH ORDER BITS GO HERE
7177          .SBTTL SAVE AND RESTORE RO-R5 ROUTINES
7178
7179          ;;*****
7180          ;;*SAVE RO-R5
7181          ;;*CALL:
7182          ;;* SAVREG
7183          ;;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
7184          ;;*
7185          ;;*TOP---(+16)
7186          ;;* +2---(+18)
7187          ;;* +4---R5
7188          ;;* +6---R4
7189          ;;* +8---R3
7190          ;;*+10---R2
7191          ;;*+12---R1
7192          ;;*+14---R0
7193
7194          $SAVREG:
7195          MOV    RO,-(SP)    ;;PUSH RO ON STACK
7196          MOV    R1,-(SP)    ;;PUSH R1 ON STACK
7197          MOV    R2,-(SP)    ;;PUSH R2 ON STACK
7198          MOV    R3,-(SP)    ;;PUSH R3 ON STACK
7199          MOV    R4,-(SP)    ;;PUSH R4 ON STACK
7200          MOV    R5,-(SP)    ;;PUSH R5 ON STACK
7201          MOV    22(SP),-(SP) ;;SAVE PS OF MAIN FLOW
7202          MOV    22(SP),-(SP) ;;SAVE PC OF MAIN FLOW
7203          MOV    22(SP),-(SP) ;;SAVE PS OF CALL
7204          MOV    22(SP),-(SP) ;;SAVE PC OF CALL
7205          RTI
7206
7207          ;;*RESTORE RO-R5
7208          ;;*CALL:

```

```

7209
7210 047402
7211 047402 012666 000022
7212 047406 012666 000022
7213 047412 012666 000022
7214 047416 012666 000022
7215 047422 012605
7216 047424 012604
7217 047426 012603
7218 047430 012602
7219 047432 012601
7220 047434 012600
7221 047436 000002
7222
7223
7224
7225
7226
7227 047440 017737 131474 004274
7228 047446 012737 047466 000024
7229 047454 012737 000340 000026
7230 047462 000000
7231 047464 000776
7232
7233
7234
7235 047466 005037 047556
7236 047472 012737 000144 047560
7237 047500 005237 047556
7238 047504 001375
7239 047506 005337 047560
7240 047512 001372
7241 047514 012737 047440 000024
7242 047522 012737 000340 000026
7243 047530 012706 001100
7244 047534 104401 047562
7245 047540 004737 043522
7246 047544 013777 004274 131366
7247 047552 000177 131330
7248
7249 047556 000000 000000
7250 047562 005015 047520 042527
7251 047570 000122
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261 047572 010046
7262 047574 016600 000002
7263 047600 005740
7264 047602 111000

; * RESREG
$RESREG:
MOV (SP)+, 22(SP) ;; RESTORE PC OF CALL
MOV (SP)+, 22(SP) ;; RESTORE PS OF CALL
MOV (SP)+, 22(SP) ;; RESTORE PC OF MAIN FLOW
MOV (SP)+, 22(SP) ;; RESTORE PS OF MAIN FLOW
MOV (SP)+, R5 ;; POP STACK INTO R5
MOV (SP)+, R4 ;; POP STACK INTO R4
MOV (SP)+, R3 ;; POP STACK INTO R3
MOV (SP)+, R2 ;; POP STACK INTO R2
MOV (SP)+, R1 ;; POP STACK INTO R1
MOV (SP)+, R0 ;; POP STACK INTO R0
RTI

.SBTTL POWER DOWN AND UP ROUTINES

; *****
; POWER DOWN ROUTINE
$PWRDN: MOV @SWR, SAVSWR ;; SAVE SWITCH REG
MOV # $PWRUP, PWRVEC ;; SET UP VECTOR
MOV # PR7, PWRVEC+2
HALT
BR -2 ;; HANG UP

; *****
; POWER UP ROUTINE
$PWRUP: CLR $PWRCT ;; LOOP LOOP TIMER
MOV #100, $PWRCT+2
1$: INC $PWRCT ;; WAIT FOR TELETYPE
BNE 1$
DEC $PWRCT+2
BNE 1$
MOV # $PWRDN, PWRVEC ;; SET UP THE POWER DOWN VECTOR
MOV # PR7, PWRVEC+2
MOV # STACK, SP ;; FORCE STACK POINTER
TYPE $POWER ;; TYPE POWER
JSR PC, CHKPAR ;; CHECK FOR MEMORY CHECK ENABLE OPTION
MOV SAVSWR, @SWR ;; RESTORE SWITCH REG
JMP @SLPADR ;; START TEST AGAIN

$PWRCT: .WORD 0,0 ;; COUNTER FOR TELETYPE
$POWER: .ASCIZ <15><12>/POWER/

.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.
$TRAP: MOV RO, -(SP) ;; SAVE RO
MOV 2(SP), RO ;; GET TRAP ADDRESS
TST -(RO) ;; BACKUP BY 2
MOVB (RO), RO ;; GET RIGHT BYTE OF TRAP

```

```

7265 047604 006300           ASL      RO           ;;POSITION FOR INDEXING
7266 047606 016000 047626   MOV      $TRPAD(RO),RO ;;INDEX TO TABLE
7267 047612 000200           RTS      RO           ;;GO TO ROUTINE
7268
7269
7270           ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
7271
7272 047614 011646           $TRAP2: MOV      (SP),-(SP) ;;MOVE THE PC DOWN
7273 047616 016666 000004 000002  MOV      4(SP),2(SP) ;;MOVE THE PSW DOWN
7274 047624 000002           RTI           ;;RESTORE THE PSW
7275
7276           .SBTTL TRAP TABLE
7277
7278           ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
7279           ;*BY THE "TRAP" INSTRUCTION.
7280
7281           ; ROUTINE
7282           ; -----
7283 047626 047614           $TRPAD: .WORD  $TRAP2
7284 047630 045346           $TYPE      ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
7285 047632 045654           $TYPOC     ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
7286 047634 045630           $TYPOS     ;;CALL=TYPOS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
7287 047636 045670           $TYPON     ;;CALL=TYPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
7288 047640 046056           $TYPDS     ;;CALL=TYPDS     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
7289
7290 047642 046352           $GTSWR     ;;CALL=GTSWR     TRAP+6(104406) GET SOFT-SWR SETTING
7291
7292 047644 046302           $CKSWR     ;;CALL=CKSWR     TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
7293 047646 046564           $RDCHR     ;;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
7294 047650 046704           $RDLIN     ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
7295 047652 047204           $RDOCT     ;;CALL=RDOCT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
7296 047654 047344           $$SAVREG   ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
7297 047656 047402           $RESREG    ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
7298 047660 044372           $SCOPI$   ;;CALL=SCOPI   TRAP+15(104415) INTERNAL LOOP ON ERROR
    
```

```

7299      .SBTTL DATA PRINTED BY ERROR ROUTINES
7300
7301 047662 001220 004272      DT000: .WORD $TESTN,TRAPPC
7302 047666 001220 001116 004160 DT001: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.MR2,T.MR2,E.MR3,T.MR3
7303 047674 004120 004206 004146
7304 047702 004210 004150
7305 047706 001220 001116 004160 DT002: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,DRVCOD,E.MR2,T.MR2,E.MR3,T.MR3
7306 047714 004120 004244 004206
7307 047722 004146 004210 004150
7308 047730 001220 001116 004160 DT006: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,HDCODE,E.MR2,T.MR2,E.MR3,T.MR3
7309 047736 004120 004250 004206
7310 047744 004146 004210 004150
7311 047752 001220 001116 004160 DT012: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.MR1,T.MR1,MSGCOD
7312 047760 004120 004204 004144
7313 047766 004246
7314 047770 004206 004146 004210      .WORD E.MR2,T.MR2,E.MR3,T.MR3
7315 047776 004150
7316 050000 001220 001116 004160 DT017: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,CYLIN,E.MR2,T.MR2,E.MR3,T.MR3
7317 050006 004120 004252 004206
7318 050014 004146 004210 004150
7319 050022 001220 001116 004160 DT031: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,OFFVAL,E.MR2,T.MR2,E.MR3,T.MR3
7320 050030 004120 004254 004206
7321 050036 004146 004210 004150
7322 050044 001220 001116 004160 DT035: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,CYLIN,OFFVAL
7323 050052 004120 004252 004254
7324 050060 004206 004146 004210      .WORD E.MR2,T.MR2,E.MR3,T.MR3
7325 050066 004150
7326 050070 001220 001116 004230 DT050: .WORD $TESTN,$ERRPC,U.MR2,U.MR3,SFTCNT,E.MR2,T.MR2,E.MR3,T.MR3
7327 050076 004232 004256 004206
7328 050104 004146 004210 004150
7329 050112 001220 001116 004210 DT052: .WORD $TESTN,$ERRPC,E.MR3,T.MR3,E.MR2,T.MR2
7330 050120 004150 004206 004146
7331 050126 001220 001116 004160 DT062: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,E.DS,T.DS,E.ER,T.ER
7332 050134 004120 004170 004130
7333 050142 004172 004132 004174
7334 050150 004134
7335 050152 001220 001116 004160 DT065: .WORD $TESTN,$ERRPC,E.CS1,T.CS1
7336 050160 004120
7337 050162 001220 001116 004206 DT067: .WORD $TESTN,$ERRPC,E.MR2,T.MR2,E.MR3,T.MR3
7338 050170 004146 004210 004150
7339 050176 001220 001116      DT100: .WORD $TESTN,$ERRPC
7340 050202 001220 001116 004160 DT126: .WORD $TESTN,$ERRPC,E.CS1,T.CS1
7341 050210 004120
7342 050212 001220 001116 004160 DT224: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,E.DS,T.DS
7343 050220 004120 004170 004130
7344 050226 004172 004132
7345 050232 004174 004134 004220      .WORD E.ER,T.ER,P.CS1,P.CS2,P.DS,P.ER
7346 050240 004222 004224 004226
7347 050246 001220 001116 004266 DT230: .WORD $TESTN,$ERRPC,DRVTYP,CYLIN,HDCODE,E.CS1,T.CS1,E.CS2,T.CS2
7348 050254 004252 004250 004160
7349 050262 004120 004170 004130
7350 050270 004172 004132 004174      .WORD E.DS,T.DS,E.ER,T.ER
7351 050276 004134
7352 050300 001220 001116 004160 DT256: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.ER,T.ER,ILLFUN
7353 050306 004120 004174 004134
7354 050314 004270

```

Address	Offset	Value	Mask	Label	Format	Notes
7355				.SBTTL	DATA FORMATS	
7356						
7357	050316	000001		DF000:	.WORD 1	
7358	050320	002	000		.BYTE 2,0	
7359	050322	000007		DF001:	.WORD 7	;ERROR 1
7360	050324	000	000		.BYTE 0,0	
7361	050326	051355			.WORD DH000A	
7362	050330	000	000		.BYTE 0,0	
7363	050332	051373			.WORD DH000B	
7364	050334	002	000		.BYTE 2,0	
7365	050336	051437			.WORD DH001A	
7366	050340	000	000		.BYTE 0,0	
7367	050342	051456			.WORD DH001B	
7368	050344	002	000		.BYTE 2,0	
7369	050346	051474			.WORD DH001C	
7370	050350	000	000		.BYTE 0,0	
7371	050352	051533			.WORD DH001D	
7372	050354	004	000		.BYTE 4,0	
7373	050356	000007		DF002:	.WORD 7	;ERRORS 2-5
7374	050360	000	000		.BYTE 0,0	
7375	050362	051355			.WORD DH000A	
7376	050364	000	000		.BYTE 0,0	
7377	050366	051373			.WORD DH000B	
7378	050370	002	000		.BYTE 2,0	
7379	050372	051572			.WORD DH002A	
7380	050374	000	000		.BYTE 0,0	
7381	050376	051620			.WORD DH002B	
7382	050400	003	000		.BYTE 3,0	
7383	050402	051474			.WORD DH001C	
7384	050404	000	000		.BYTE 0,0	
7385	050406	051533			.WORD DH001D	
7386	050410	004	000		.BYTE 4,0	
7387	050412	000007		DF006:	.WORD 7	;ERRORS 6-11
7388	050414	000	000		.BYTE 0,0	
7389	050416	051355			.WORD DH000A	
7390	050420	000	000		.BYTE 0,0	
7391	050422	051373			.WORD DH000B	
7392	050424	002	000		.BYTE 2,0	
7393	050426	051647			.WORD DH006A	
7394	050430	000	000		.BYTE 0,0	
7395	050432	051674			.WORD DH006B	
7396	050434	003	000		.BYTE 3,0	
7397	050436	051474			.WORD DH001C	
7398	050440	000	000		.BYTE 0,0	
7399	050442	051533			.WORD DH001D	
7400	050444	004	000		.BYTE 4,0	
7401	050446	000007		DF012:	.WORD 7	;ERRORS12-16
7402	050450	000	000		.BYTE 0,0	
7403	050452	051355			.WORD DH000A	
7404	050454	000	000		.BYTE 0,0	
7405	050456	051373			.WORD DH000B	
7406	050460	002	000		.BYTE 2,0	
7407	050462	051720			.WORD DH012A	
7408	050464	000	000		.BYTE 0,0	
7409	050466	051765			.WORD DH012B	
7410	050470	005	000		.BYTE 5,0	

7411	050472	051474		.WORD	DH001C
7412	050474	000	000	.BYTE	0,0
7413	050476	051533		.WORD	DH001D
7414	050500	004	000	.BYTE	4,0
7415	050502	000007		.WORD	7
7416	050504	000	000	.BYTE	0,0
7417	050506	051355		.WORD	DH000A
7418	050510	000	000	.BYTE	0,0
7419	050512	051373		.WORD	DH000B
7420	050514	002	000	.BYTE	2,0
7421	050516	052033		.WORD	DH017A
7422	050520	000	000	.BYTE	0,0
7423	050522	052061		.WORD	DH017B
7424	050524	003	000	.BYTE	3,0
7425	050526	051474		.WORD	DH001C
7426	050530	000	000	.BYTE	0,0
7427	050532	051533		.WORD	DH001D
7428	050534	004	000	.BYTE	4,0
7429	050536	000007		.WORD	7
7430	050540	000	000	.BYTE	0,0
7431	050542	051355		.WORD	DH000A
7432	050544	000	000	.BYTE	0,0
7433	050546	051373		.WORD	DH000B
7434	050550	002	000	.BYTE	2,0
7435	050552	052105		.WORD	DH031A
7436	050554	000	000	.BYTE	0,0
7437	050556	052134		.WORD	DH031B
7438	050560	003	000	.BYTE	3,0
7439	050562	051474		.WORD	DH001C
7440	050564	000	000	.BYTE	0,0
7441	050566	051533		.WORD	DH001D
7442	050570	004	000	.BYTE	4,0
7443	050572	000007		.WORD	7
7444	050574	000	000	.BYTE	0,0
7445	050576	051355		.WORD	DH000A
7446	050600	000	000	.BYTE	0,0
7447	050602	051373		.WORD	DH000B
7448	050604	002	000	.BYTE	2,0
7449	050606	052162		.WORD	DH035A
7450	050610	000	000	.BYTE	0,0
7451	050612	052221		.WORD	DH035B
7452	050614	004	000	.BYTE	4,0
7453	050616	051474		.WORD	DH001C
7454	050620	000	000	.BYTE	0,0
7455	050622	051533		.WORD	DH001D
7456	050624	004	000	.BYTE	4,0
7457	050626	000007		.WORD	7
7458	050630	000	000	.BYTE	0,0
7459	050632	051355		.WORD	DH000A
7460	050634	000	000	.BYTE	0,0
7461	050636	051373		.WORD	DH000B
7462	050640	002	000	.BYTE	2,0
7463	050642	052257		.WORD	DH050A
7464	050644	000	000	.BYTE	0,0
7465	050646	052305		.WORD	DH050B
7466	050650	003	000	.BYTE	3,0

DF017:

;ERRORS 17-30

DF031:

;ERRORS 31-34

DF035:

;ERROR 35-41

DF050:

;ERRORS 50 & 51

7467	050652	051474		.WORD	DH001C	
7468	050654	000	000	.BYTE	0,0	
7469	050656	051533		.WORD	DH001D	
7470	050660	004	000	.BYTE	4,0	
7471	050662	000005		.WORD	5	:ERRORS 52-61
7472	050664	000	000	.BYTE	0,0	
7473	050666	051355		.WORD	DH000A	
7474	050670	000	000	.BYTE	0,0	
7475	050672	051373		.WORD	DH000B	
7476	050674	002	000	.BYTE	2,0	
7477	050676	051474		.WORD	DH001C	
7478	050700	000	000	.BYTE	0,0	
7479	050702	051533		.WORD	DH001D	
7480	050704	004	000	.BYTE	4,0	
7481	050706	000005		.WORD	5	:ERRORS 62-64
7482	050710	000	000	.BYTE	0,0	
7483	050712	051355		.WORD	DH000A	
7484	050714	000	000	.BYTE	0,0	
7485	050716	051373		.WORD	DH000B	
7486	050720	002	000	.BYTE	2,0	
7487	050722	052333		.WORD	DH062A	
7488	050724	000	000	.BYTE	0,0	
7489	050726	052432		.WORD	DH062B	
7490	050730	010	000	.BYTE	8.,0	
7491	050732	000005		.WORD	5	:ERRORS-65-66
7492	050734	000	000	.BYTE	0,0	
7493	050736	051355		.WORD	DH000A	
7494	050740	000	000	.BYTE	0,0	
7495	050742	051373		.WORD	DH000B	
7496	050744	002	000	.BYTE	2,0	
7497	050746	051437		.WORD	DH001A	
7498	050750	000	000	.BYTE	0,0	
7499	050752	051456		.WORD	DH001B	
7500	050754	002	000	.BYTE	2,0	
7501	050756	000005		.WORD	5	:ERRORS 67-70
7502	050760	000	000	.BYTE	0,0	
7503	050762	051355		.WORD	DH000A	
7504	050764	000	000	.BYTE	0,0	
7505	050766	051373		.WORD	DH000B	
7506	050770	002	000	.BYTE	2,0	
7507	050772	052527		.WORD	DH067A	
7508	050774	000	000	.BYTE	0,0	
7509	050776	052566		.WORD	DH067B	
7510	051000	004	000	.BYTE	4,0	
7511	051002	000003		.WORD	3	:ERROR 100
7512	051004	000	000	.BYTE	0,0	
7513	051006	051355		.WORD	DH000A	
7514	051010	000	000	.BYTE	0,0	
7515	051012	051373		.WORD	DH000B	
7516	051014	002	000	.BYTE	2,0	
7517	051016	000005		.WORD	5	:ERROR 126
7518	051020	000	000	.BYTE	0,0	
7519	051022	051355		.WORD	DH000A	
7520	051024	000	000	.BYTE	0,0	
7521	051026	051373		.WORD	DH000B	
7522	051030	002	000	.BYTE	2,0	

DZR6BB.P11 DATA FORMATS

7523	051032	052624	
7524	051034	000	000
7525	051036	052643	
7526	051040	002	000
7527	051042	000007	
7528	051044	000	000
7529	051046	051355	
7530	051050	000	000
7531	051052	051373	
7532	051054	002	000
7533	051056	052333	
7534	051060	000	000
7535	051062	052432	
7536	051064	010	000
7537	051066	052661	
7538	051070	000	000
7539	051072	052714	
7540	051074	004	000
7541	051076	000007	
7542	051100	000	000
7543	051102	051355	
7544	051104	000	000
7545	051106	051373	
7546	051110	002	000
7547	051112	052751	
7548	051114	000	000
7549	051116	052776	
7550	051120	003	000
7551	051122	052333	
7552	051124	000	000
7553	051126	052432	
7554	051130	010	000
7555	051132	000005	
7556	051134	000	000
7557	051136	051355	
7558	051140	000	000
7559	051142	051373	
7560	051144	002	000
7561	051146	053022	
7562	051150	000	000
7563	051152	053066	
7564	051154	005	000

	.WORD	DH126A	
	.BYTE	0,0	
	.WORD	DH126B	
	.BYTE	2,0	
DF224:	.WORD	7	:ERRORS 224-227
	.BYTE	0,0	
	.WORD	DH000A	
	.BYTE	0,0	
	.WORD	DH000B	
	.BYTE	2,0	
	.WORD	DH062A	
	.BYTE	0,0	
	.WORD	DH062B	
	.BYTE	8.,0	
	.WORD	DH224A	
	.BYTE	0,0	
	.WORD	DH224B	
	.BYTE	4,0	
DF230:	.WORD	7	:ERRORS 230-233
	.BYTE	0,0	
	.WORD	DH000A	
	.BYTE	0,0	
	.WORD	DH000B	
	.BYTE	2,0	
	.WORD	DH230A	
	.BYTE	0,0	
	.WORD	DH230B	
	.BYTE	3,0	
	.WORD	DH062A	
	.BYTE	0,0	
	.WORD	DH062B	
	.BYTE	8.,0	
DF256:	.WORD	5	:ERROR 256
	.BYTE	0,0	
	.WORD	DH000A	
	.BYTE	0,0	
	.WORD	DH000B	
	.BYTE	2,0	
	.WORD	DH256A	
	.BYTE	0,0	
	.WORD	DH256B	
	.BYTE	5,0	

```

7565 .SBTTL ASCII MESSAGES
7566
7567 051156 005015 045522 030466 OPRO01: .ASCIZ <15><12>/RK611 BUS ADDRESS ( /
7568 051164 020061 052502 020123
7569 051172 042101 051104 051505
7570 051200 020123 020050 000
7571 051205 040 020051 020075 OPRO02: .ASCIZ / ) = /
7572 051212 000
7573 051213 122 033113 030461 OPRO03: .ASCIZ /RK611 VECTOR ADDRESS ( /
7574 051220 053040 041505 047524
7575 051226 020122 042101 051104
7576 051234 051505 020123 020050
7577 051242 000
7578 051243 122 033113 030461 OPRO04: .ASCIZ /RK611 PRIORITY ( /
7579 051250 050040 044522 051117
7580 051256 052111 020131 020050
7581 051264 000
7582 051265 040 000040 SPACE2: .ASCIZ / /
7583 051270 005015 051120 043517 ABORT: .ASCIZ <15><12>/PROGRAM ABORTED BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
7584 051276 040522 020115 041101
7585 051304 051117 042524 020104
7586 051312 042502 040503 051525
7587 051320 020105 051105 047522
7588 051326 020122 044124 042522
7589 051334 044123 046117 020104
7590 051342 054105 042503 042105
7591 051350 042105 005015 000

```

```

7592          .SBTTL DATA HEADERS
7593
7594 051355      124 051505 020124 DH000A: .ASCIZ /TEST ERROR/
7595 051362    020040 042440 051122
7596 051370    051117      000
7597 051373      116 046525 020040 DH000B: .ASCIZ /NUM PC/
7598 051400    020040 050040 000103
7599 051406    042524 052123 020040 DH000C: .ASCII /TEST TRAP/<15><12>
7600 051414    020040 051124 050101
7601 051422    005015
7602 051424    052516 020115 020040          .ASCIZ /NUM PC/
7603 051432    020040 041520      000
7604 051437      105 050130 041505 DH001A: .ASCIZ /EXPECT ACTUAL/
7605 051444    020124 040440 052103
7606 051452    040525 000114
7607 051456    045522 051503 020061 DH001B: .ASCIZ /RKCS1 RKCS1/
7608 051464    020040 045522 051503
7609 051472    000061
7610 051474    054105 042520 052103 DH001C: .ASCIZ /EXPECT ACTUAL EXPECT ACTUAL/
7611 051502    020040 041501 052524
7612 051510    046101 020040 054105
7613 051516    042520 052103 020040
7614 051524    041501 052524 046101
7615 051532      000
7616 051533      115 051505 020123 DH001D: .ASCIZ /MESS A MESS A MESS B MESS B/
7617 051540    020101 046440 051505
7618 051546    020123 020101 046440
7619 051554    051505 020123 020102
7620 051562    046440 051505 020123
7621 051570    000102
7622 051572    054105 042520 052103 DH002A: .ASCIZ /EXPECT ACTUAL DRIVE/
7623 051600    020040 041501 052524
7624 051606    046101 020040 051104
7625 051614    053111 000105
7626 051620    045522 051503 020061 DH002B: .ASCIZ /RKCS1 RKCS1 SELECT/
7627 051626    020040 045522 051503
7628 051634    020061 020040 042523
7629 051642    042514 052103      000
7630 051647      105 050130 041505 DH006A: .ASCIZ /EXPECT ACTUAL HEAD/
7631 051654    020124 040440 052103
7632 051662    040525 020114 044040
7633 051670    040505 000104
7634 051674    045522 051503 020061 DH006B: .ASCIZ /RKCS1 RKCS1 ADD/
7635 051702    020040 045522 051503
7636 051710    020061 020040 042101
7637 051716    000104
7638 051720    054105 042520 052103 DH012A: .ASCIZ /EXPECT ACTUAL EXPECT ACTUAL MESS/
7639 051726    020040 041501 052524
7640 051734    046101 020040 054105
7641 05174      042520 052103 020040
7642 051750    041501 052524 046101
7643 051756    020040 042515 051523
7644 051764      000
7645 051765      122 041513 030523 DH012B: .ASCIZ /RKCS1 RKCS1 RKMRI RKMRI SELECT/
7646 051772    020040 051040 041513
7647 052000    030523 020040 051040

```



7704	052470	020040	045522	051504						
7705	052476	020040	020040	045522						
7706	052504	051504	020040	020040						
7707	052512	045522	051105	020040						
7708	052520	020040	045522	051105						
7709	052526	000								
7710	052527	105	050130	041505	DH067A:	.ASCIZ	/EXPECT	ACTUAL	EXPECT	ACTUAL/
7711	052534	020124	040440	052103						
7712	052542	040525	020114	042440						
7713	052550	050130	041505	020124						
7714	052556	040440	052103	040525						
7715	052564	000114								
7716	052566	045522	051115	020062	DH067B:	.ASCIZ	/RKMR2	RKMR2	RKMR3	RKMR3/
7717	052574	020040	045522	051115						
7718	052602	020062	020040	045522						
7719	052610	051115	020063	020040						
7720	052616	045522	051115	000063						
7721	052624	054105	042520	052103	DH126A:	.ASCIZ	/EXPECT	ACTUAL/		
7722	052632	020040	041501	052524						
7723	052640	046101	000							
7724	052643	122	041513	030523	DH126B:	.ASCIZ	/RKCS1	RKCS1/		
7725	052650	020040	051040	041513						
7726	052656	030523	000							
7727	052661	120	042522	044526	DH224A:	.ASCIZ	/PREVIOUS	REGISTER	CONTENTS/	
7728	052666	052517	020123	042522						
7729	052674	044507	052123	051105						
7730	052702	041440	047117	042524						
7731	052710	052116	000123							
7732	052714	045522	051503	020061	DH224B:	.ASCIZ	/RKCS1	RKCS2	RKDS	RKER/
7733	052722	020040	045522	051503						
7734	052730	020062	020040	045522						
7735	052736	051504	020040	020040						
7736	052744	045522	051105	000						
7737	052751	104	044522	042526	DH230A:	.ASCIZ	/DRIVE	CYLIN	HEAD/	
7738	052756	020040	041440	046131						
7739	052764	047111	020040	044040						
7740	052772	040505	000104							
7741	052776	054524	042520	020040	DH230B:	.ASCIZ	/TYPE	ADD	ADD/	
7742	053004	020040	042101	020104						
7743	053012	020040	020040	042101						
7744	053020	000104								
7745	053022	054105	042520	052103	DH256A:	.ASCIZ	/EXPECT	ACTUAL	EXPECT	ACTUAL ILL/
7746	053030	020040	041501	052524						
7747	053036	046101	020040	054105						
7748	053044	042520	052103	020040						
7749	053052	041501	052524	046101						
7750	053060	020040	046111	000114						
7751	053066	045522	051503	020061	DH256B:	.ASCIZ	/RKCS1	RKCS1	RKER	RKER FUNCT/
7752	053074	020040	045522	051503						
7753	053102	020061	020040	045522						
7754	053110	051105	020040	020040						
7755	053116	045522	051105	020040						
7756	053124	020040	052506	041516						
7757	053132	000124								

.SBTTL ERROR MESSAGES

7758					
7759					
7760	053134	047125	054105	042520	EM000: .ASCIZ /UNEXPECTED MEMORY PARITY ENABLE TRAP/
7761	053142	052103	042105	046440	
7762	053150	046505	051117	020131	
7763	053156	040520	044522	054524	
7764	053164	042440	040516	046102	
7765	053172	020105	051124	050101	
7766	053200	000			
7767	053201	101	052124	046505	EM100: .ASCIZ /ATTEMPTING A SELECT IN 24 SECTOR FORMAT IN MAINT MODE/
7768	053206	052120	047111	020107	
7769	053214	020101	042523	042514	
7770	053222	052103	044440	020116	
7771	053230	032062	051440	041505	
7772	053236	047524	020122	047506	
7773	053244	046522	052101	044440	
7774	053252	020116	040515	047111	
7775	053260	020124	047515	042504	
7776	053266	000			
7777	053267	101	052124	046505	EM101: .ASCIZ /ATTEMPTING A DRIVE CLEAR IN MAINT MODE/
7778	053274	052120	047111	020107	
7779	053302	020101	051104	053111	
7780	053310	020105	046103	040505	
7781	053316	020122	047111	046440	
7782	053324	044501	052116	046440	
7783	053332	042117	000105		
7784	053336	052101	042524	050115	EM102: .ASCIZ /ATTEMPTING A UNLOAD IN MAINT MODE/
7785	053344	044524	043516	040440	
7786	053352	052440	046116	040517	
7787	053360	020104	047111	046440	
7788	053366	044501	052116	046440	
7789	053374	042117	000105		
7790	053400	052101	042524	050115	EM103: .ASCIZ /ATTEMPTING A PACK ACKNOWLEDGE IN MAINT MODE/
7791	053406	044524	043516	040440	
7792	053414	050040	041501	020113	
7793	053422	041501	047113	053517	
7794	053430	042514	043504	020105	
7795	053436	047111	046440	044501	
7796	053444	052116	046440	042117	
7797	053452	000105			
7798	053454	052101	042524	050115	EM104: .ASCIZ /ATTEMPTING A RECALIBRATE IN MAINT MODE/
7799	053462	044524	043516	040440	
7800	053470	051040	041505	046101	
7801	053476	041111	040522	042524	
7802	053504	044440	020116	040515	
7803	053512	047111	020124	047515	
7804	053520	042504	000		
7805	053523	101	052124	046505	EM105: .ASCIZ /ATTEMPTING A START SPINDLE/
7806	053530	052120	047111	020107	
7807	053536	020101	052123	051101	
7808	053544	020124	050123	047111	
7809	053552	046104	000105		
7810	053556	052101	042524	050115	EM106: .ASCIZ /ATTEMPTING A SELECT USING ALL DRIVE SELECTION CONFIGS IN MAINT MODE/
7811	053564	044524	043516	040440	
7812	053572	051440	046105	041505	
7813	053600	020124	051525	047111	

7814	053606	020107	046101	020114	
7815	053614	051104	053111	020105	
7816	053622	042523	042514	052103	
7817	053630	047511	020116	047503	
7818	053636	043116	043511	020123	
7819	053644	047111	046440	044501	
7820	053652	052116	046440	042117	
7821	053660	000105			
7822	053662	052101	042524	050115	EM107: .ASCIZ /ATTEMPTING A SELECT USING ALL HEAD ADD CONFIGS IN MAINT MODE/
7823	053670	044524	043516	040440	
7824	053676	051440	046105	041505	
7825	053704	020124	051525	047111	
7826	053712	020107	046101	020114	
7827	053720	042510	042101	040440	
7828	053726	042104	041440	047117	
7829	053734	044506	051507	044440	
7830	053742	020116	040515	047111	
7831	053750	020124	047515	042504	
7832	053756	000			
7833	053757	101	052124	046505	EM108: .ASCIZ /ATTEMPTING A SELECT USING ALL MESS SELECT CONFIGS IN MAINT MODE/
7834	053764	052120	047111	020107	
7835	053772	020101	042523	042514	
7836	054000	052103	052440	044523	
7837	054006	043516	040440	046114	
7838	054014	046440	051505	020123	
7839	054022	042523	042514	052103	
7840	054030	041440	047117	044506	
7841	054036	051507	044440	020116	
7842	054044	040515	047111	020124	
7843	054052	047515	042504	000	
7844	054057	101	052124	046505	EM109: .ASCIZ /ATTEMPTING A SEEK TO AN RK06 IN MAINT MODE/
7845	054064	052120	047111	020107	
7846	054072	020101	042523	045505	
7847	054100	052040	020117	047101	
7848	054106	051040	030113	020066	
7849	054114	047111	046440	044501	
7850	054122	052116	046440	042117	
7851	054130	000105			
7852	054132	052101	042524	050115	EM110: .ASCIZ /ATTEMPTING A SEEK WITH CDT SET IN MAINT MODE/
7853	054140	044524	043516	040440	
7854	054146	051440	042505	020113	
7855	054154	044527	044124	041440	
7856	054162	052104	051440	052105	
7857	054170	044440	020116	040515	
7858	054176	047111	020124	047515	
7859	054204	042504	000		
7860	054207	101	052124	046505	EM111: .ASCIZ /ATTEMPTING AN OFFSET IN MAINT MODE/
7861	054214	052120	047111	020107	
7862	054222	047101	047440	043106	
7863	054230	042523	020124	047111	
7864	054236	046440	044501	052116	
7865	054244	046440	042117	000105	
7866	054252	052101	042524	050115	EM112: .ASCII /ATTEMPTING COMMAND WITH NON-ZERO CYLINDER ADDRESS AND/<15><12>
7867	054260	044524	043516	041440	
7868	054266	046517	040515	042116	
7869	054274	053440	052111	020110	

7870	054302	047516	026516	042532	
7871	054310	047522	041440	046131	
7872	054316	047111	042504	020122	
7873	054324	042101	051104	051505	
7874	054332	020123	047101	006504	
7875	054340	012			
7876	054341	116	047117	055055	.ASCIZ /NON-ZERO OFFSET IN MAINTENANCE MODE/
7877	054346	051105	020117	043117	
7878	054354	051506	052105	044440	
7879	054362	020116	040515	047111	
7880	054370	042524	040516	041516	
7881	054376	020105	047515	042504	
7882	054404	000			
7883	054405	101	052124	046505	EM113: .ASCII /ATTEMPTING COMMAND WITH NON-ZERO MESSAGE SELECT CODE/<15><12>
7884	054412	052120	047111	020107	
7885	054420	047503	046515	047101	
7886	054426	020104	044527	044124	
7887	054434	047040	047117	055055	
7888	054442	051105	020117	042515	
7889	054450	051523	043501	020105	
7890	054456	042523	042514	052103	
7891	054464	041440	042117	006505	
7892	054472	012			
7893	054473	111	020116	040515	.ASCIZ /IN MAINTENANCE MODE/
7894	054500	047111	042524	040516	
7895	054506	041516	020105	047515	
7896	054514	042504	000		
7897	054517	101	052124	046505	EM114: .ASCIZ /ATTEMPTING TO SHIFT DRIVE MESSAGES/
7898	054524	052120	047111	020107	
7899	054532	047524	051440	044510	
7900	054540	052106	042040	044522	
7901	054546	042526	046440	051505	
7902	054554	040523	042507	000123	
7903	054562	052101	042524	050115	EM115: .ASCIZ /ATTEMPTING TO GENERATE ODD PARITY ON SELECT DRIVE MESSAGE/
7904	054570	044524	043516	052040	
7905	054576	020117	042507	042516	
7906	054604	040522	042524	047440	
7907	054612	042104	050040	051101	
7908	054620	052111	020131	047117	
7909	054626	051440	046105	041505	
7910	054634	020124	051104	053111	
7911	054642	020105	042515	051523	
7912	054650	043501	000105		
7913	054654	052101	042524	050115	EM116: .ASCIZ /ATTEMPTING TO GENERATE EVEN PARITY ON SELECT DRIVE MESSAGE/
7914	054662	044524	043516	052040	
7915	054670	020117	042507	042516	
7916	054676	040522	042524	042440	
7917	054704	042526	020116	040520	
7918	054712	044522	054524	047440	
7919	054720	020116	042523	042514	
7920	054726	052103	042040	044522	
7921	054734	042526	046440	051505	
7922	054742	040523	042507	000	
7923	054747	101	052124	046505	EM117: .ASCII /ATTEMPTING COMPLETE EXECUTION OF DESELECT DRIVE COMMAND/
7924	054754	052120	047111	020107	
7925	054762	047503	050115	042514	

7926	054770	042524	042440	042530	
7927	054776	052503	044524	047117	
7928	055004	047440	020106	042504	
7929	055012	042523	042514	052103	
7930	055020	042040	044522	042526	
7931	055026	041440	046517	040515	
7932	055034	042116			
7933	055036	005015	047111	046440	.ASCIZ <15><12>/IN MAINTENANCE MODE/
7934	055044	044501	052116	047105	
7935	055052	047101	042503	046440	
7936	055060	042117	000105		
7937	055064	052101	042524	050115	EM118: .ASCII /ATTEMPTING COMPLETE EXECUTION OF SELECT DRIVE COMMAND/
7938	055072	044524	043516	041440	
7939	055100	046517	046120	052105	
7940	055106	020105	054105	041505	
7941	055114	052125	047511	020116	
7942	055122	043117	051440	046105	
7943	055130	041505	020124	051104	
7944	055136	053111	020105	047503	
7945	055144	046515	047101	104	
7946	055151	015	044412	020116	.ASCIZ <15><12>/IN MAINTENANCE MODE/
7947	055156	040515	047111	042524	
7948	055164	040516	041516	020105	
7949	055172	047515	042504	000	
7950	055177	101	052124	046505	EM119: .ASCIZ /ATTEMPTING EXECUTION OF DESELECT DRIVE AT NORMAL SPEED/
7951	055204	052120	047111	020107	
7952	055212	054105	041505	052125	
7953	055220	047511	020116	043117	
7954	055226	042040	051505	046105	
7955	055234	041505	020124	051104	
7956	055242	053111	020105	052101	
7957	055250	047040	051117	040515	
7958	055256	020114	050123	042505	
7959	055264	000104			
7960	055266	052101	042524	050115	EM120: .ASCIZ /ATTEMPTING TO WRITE COMMAND AND STATUS REG. 1 IN MAINT MODE/
7961	055274	044524	043516	052040	
7962	055302	020117	051127	052111	
7963	055310	020105	047503	046515	
7964	055316	047101	020104	047101	
7965	055324	020104	052123	052101	
7966	055332	051525	051040	043505	
7967	055340	020056	020061	047111	
7968	055346	046440	044501	052116	
7969	055354	046440	042117	000105	
7970	055362	052101	042524	050115	EM121: .ASCIZ /ATTEMPTING EXECUTION OF DESELECT DRIVE WITH INTERRUPT ENABLE SET/
7971	055370	044524	043516	042440	
7972	055376	042530	052503	044524	
7973	055404	047117	047440	020106	
7974	055412	042504	042523	042514	
7975	055420	052103	042040	044522	
7976	055426	042526	053440	052111	
7977	055434	020110	047111	042524	
7978	055442	051122	050125	020124	
7979	055450	047105	041101	042514	
7980	055456	051440	052105	000	
7981	055463	101	052124	046505	EM122: .ASCII /ATTEMPTING DESELECT COMMAND AFTER WRITING SILO /

DZR6BB.P11

ERROR MESSAGES

7982	055470	052120	047111	020107	
7983	055476	042504	042523	042514	
7984	055504	052103	041440	046517	
7985	055512	040515	042116	040440	
7986	055520	052106	051105	053440	
7987	055526	044522	044524	043516	
7988	055534	051440	046111	020117	
7989	055542	047524	041440	042510	.ASCIZ /TO CHECK GO CLEAR/
7990	055550	045503	043440	020117	
7991	055556	046103	040505	000122	
7992	055564	052101	042524	050115	EM123: .ASCIZ /ATTEMPTING COMPLETE EXECUTION OF SEEK IN MAINT MODE/
7993	055572	044524	043516	041440	
7994	055600	046517	046120	052105	
7995	055606	020105	054105	041505	
7996	055614	052125	047511	020116	
7997	055622	043117	051440	042505	
7998	055630	020113	047111	046440	
7999	055636	044501	052116	046440	
8000	055644	042117	000105		
8001	055650	052101	042524	050115	EM124: .ASCIZ /ATTEMPTING SELECT DRIVE IN MAINT MODE/
8002	055656	044524	043516	051440	
8003	055664	046105	041505	020124	
8004	055672	051104	053111	020105	
8005	055700	047111	046440	044501	
8006	055706	052116	046440	042117	
8007	055714	000105			
8008	055716	052101	042524	050115	EM125: .ASCII /ATTEMPTING CHECK "LOAD STATUS" BY FORCING/<15><12>
8009	055724	044524	043516	041440	
8010	055732	042510	045503	021040	
8011	055740	047514	042101	051440	
8012	055746	040524	052524	021123	
8013	055754	041040	020131	047506	
8014	055762	041522	047111	006507	
8015	055770	012			
8016	055771	104	044522	042526	.ASCII /DRIVE AVAILIABLE, SPEED LOSS, VOLUME VALID,<15><12>
8017	055776	040440	040526	046111	
8018	056004	040511	046102	026105	
8019	056012	051440	042520	042105	
8020	056020	046040	051517	026123	
8021	056026	053040	046117	046525	
8022	056034	020105	040526	044514	
8023	056042	026104	005015		
8024	056046	043117	051506	052105	.ASCII /OFFSET, DRIVE READY, AND WRITE LOCK/<15><12>
8025	056054	020054	051104	053111	
8026	056062	020105	042522	042101	
8027	056070	026131	040440	042116	
8028	056076	053440	044522	042524	
8029	056104	046040	041517	006513	
8030	056112	012			
8031	056113	104	044522	042526	.ASCIZ /DRIVE STATUS REG./
8032	056120	051440	040524	052524	
8033	056126	020123	042522	027107	
8034	056134	000			
8035	056135	101	052124	046505	EM126: .ASCIZ /ATTEMPTING TO FORCE DRIVE AVAILIABLE/
8036	056142	052120	047111	020107	
8037	056150	047524	043040	051117	

8038	056156	042503	042040	044522	
8039	056164	042526	040440	040526	
8040	056172	046111	040511	046102	
8041	056200	000105			
8042	056202	052101	042524	050115	EM127: .ASCII /ATTEMPTING TO FORCE DRIVE BUS PARITY ERROR/<15><12>
8043	056210	044524	043516	052040	
8044	056216	020117	047506	041522	
8045	056224	020105	051104	053111	
8046	056232	020105	052502	020123	
8047	056240	040520	044522	054524	
8048	056246	042440	051122	051117	
8049	056254	005015			
8050	056256	042504	042524	052103	.ASCIZ /DETECTED BY RK611/
8051	056264	042105	041040	020131	
8052	056272	045522	030466	000061	
8053	056300	052101	042524	050115	EM128: .ASCIZ /ATTEMPTING TO FORCE DRIVE AVAILABLE RESET ERROR/
8054	056306	044524	043516	052040	
8055	056314	020117	047506	041522	
8056	056322	020105	051104	053111	
8057	056330	020105	053101	044501	
8058	056336	044514	041101	042514	
8059	056344	051040	051505	052105	
8060	056352	042440	051122	051117	
8061	056360	000			
8062	056361	124	051505	044524	EM129: .ASCIZ /TESTING CDT SET DRIVE TYPE DETECTION/
8063	056366	043516	041440	052104	
8064	056374	051440	052105	042040	
8065	056402	044522	042526	052040	
8066	056410	050131	020105	042504	
8067	056416	042524	052103	047511	
8068	056424	000116			
8069	056426	052101	042524	050115	EM130: .ASCIZ /ATTEMPTING TO FORCE DRIVE TYPE ERROR WITH CDT SET/
8070	056434	044524	043516	052040	
8071	056442	020117	047506	041522	
8072	056450	020105	051104	053111	
8073	056456	020105	054524	042520	
8074	056464	042440	051122	051117	
8075	056472	053440	052111	020110	
8076	056500	042103	020124	042523	
8077	056506	000124			
8078	056510	052101	042524	050115	EM131: .ASCIZ /ATTEMPTING TO FORCE DRIVE TYPE ERROR ADDRESSING RK06/
8079	056516	044524	043516	052040	
8080	056524	020117	047506	041522	
8081	056532	020105	051104	053111	
8082	056540	020105	054524	042520	
8083	056546	042440	051122	051117	
8084	056554	040440	042104	042522	
8085	056562	051523	047111	020107	
8086	056570	045522	033060	000	
8087	056575	101	052124	046505	EM132: .ASCIZ /ATTEMPTING TO FORCE SPEED LOSS/
8088	056602	052120	047111	020107	
8089	056610	047524	043040	051117	
8090	056616	042503	051440	042520	
8091	056624	042105	046040	051517	
8092	056632	000123			
8093	056634	052101	042524	050115	EM133: .ASCIZ /ATTEMPTING TO FORCE DRIVE OFF TRACK/

Address	Hex 1	Hex 2	Hex 3	Hex 4	Message
0094	056642	040024	043516	052040	
0095	056650	020117	047506	041522	
0096	056656	020105	051104	053111	
0097	056664	020105	043117	020106	
0098	056672	051124	041501	000113	
0099	056700	052101	042524	050115	EM134: .ASCIZ /ATTEMPTING TO FORCE WRITE LOCK ERROR/
0100	056706	044524	043516	052040	
0101	056714	020117	047506	041522	
0102	056722	020105	051127	052111	
0103	056730	020105	047514	045503	
0104	056736	042440	051122	051117	
0105	056744	000			
0106	056745	101	052124	046505	EM135: .ASCIZ /ATTEMPTING TO FORCE SEEK INCOMPLETE/
0107	056752	052120	047111	020107	
0108	056760	047524	043040	051117	
0109	056766	042503	051440	042505	
0110	056774	020113	047111	047503	
0111	057002	050115	042514	042524	
0112	057010	000			
0113	057011	101	052124	046505	EM136: .ASCIZ /ATTEMPTING TO FORCE NON-EXECUTABLE FUNCTION/
0114	057016	052120	047111	020107	
0115	057024	047524	043040	051117	
0116	057032	042503	047040	047117	
0117	057040	042455	042530	052503	
0118	057046	040524	046102	020105	
0119	057054	052506	041516	044524	
0120	057062	047117	000		
0121	057065	101	052124	046505	EM137: .ASCIZ /ATTEMPTING TO FORCE AC LOW AND C-D PARITY ERROR/
0122	057072	052120	047111	020107	
0123	057100	047524	043040	051117	
0124	057106	042503	040440	020103	
0125	057114	047514	020127	047101	
0126	057122	020104	026503	020104	
0127	057130	040520	044522	054524	
0128	057136	042440	051122	051117	
0129	057144	000			
0130	057145	101	052124	046505	EM138: .ASCII /ATTEMPTING TO FORCE ILLEGAL DISK ADDRESS ERROR/
0131	057152	052120	047111	020107	
0132	057160	047524	043040	051117	
0133	057166	042503	044440	046114	
0134	057174	043505	046101	042040	
0135	057202	051511	020113	042101	
0136	057210	051104	051505	020123	
0137	057216	051105	047522	122	
0138	057223	015	043012	047522	.ASCIZ <15><12>/FROM DRIVE MESSAGE BITS/
0139	057230	020115	051104	053111	
0140	057236	020105	042515	051523	
0141	057244	043501	020105	044502	
0142	057252	051524	000		
0143	057255	101	052124	046505	EM139: .ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR/
0144	057262	052120	047111	020107	
0145	057270	047524	041440	042514	
0146	057276	051101	051040	033113	
0147	057304	030461	053440	052111	
0148	057312	020110	020101	047503	
0149	057320	052116	047522	046114	

DZR68B.P11

ERROR MESSAGES

0150	057326	051105	041440	042514	
0151	057334	051101	000		
0152	057337	124	051505	044524	EM140: .ASCIZ /TESTING ILLEGAL DISK ADDRESS ERROR LOGIC IN RK611/
0153	057344	043516	044440	046114	
0154	057352	043505	046101	042040	
0155	057360	051511	020113	042101	
0156	057366	051104	051505	020123	
0157	057374	051105	047522	020122	
0158	057402	047514	044507	020103	
0159	057410	047111	051040	033113	
0160	057416	030461	000		
0161	057421	101	052124	046505	EM141: .ASCIZ /ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES/
0162	057426	052120	047111	020107	
0163	057434	047524	051040	041505	
0164	057442	044505	042526	047040	
0165	057450	047117	051455	040524	
0166	057456	042116	051101	020104	
0167	057464	042515	051523	043501	
0168	057472	051505	000		
0169	057475	101	052124	046505	EM142: .ASCII /ATTEMPTING TO RECEIVE NON-STANDARD MESSAGES/
0170	057502	052120	047111	020107	
0171	057510	047524	051040	041505	
0172	057516	044505	042526	047040	
0173	057524	047117	051455	040524	
0174	057532	042116	051101	020104	
0175	057540	042515	051523	043501	
0176	057546	051505			
0177	057550	053440	052111	020110	.ASCIZ / WITH PARITY ERROR/
0178	057556	040520	044522	054524	
0179	057564	042440	051122	051117	
0180	057572	000			
0181	057573	101	052124	046505	EM143: .ASCIZ /ATTEMPTING TO FORCE NON-EXISTENT DRIVE (DRIVE BUS TIMEOUT)/
0182	057600	052120	047111	020107	
0183	057606	047524	043040	051117	
0184	057614	042503	047040	047117	
0185	057622	042455	044530	052123	
0186	057630	047105	020124	051104	
0187	057636	053111	020105	042050	
0188	057644	044522	042526	041040	
0189	057652	051525	052040	046511	
0190	057660	047505	052125	000051	
0191	057666	052101	042524	050115	EM144: .ASCIZ /ATTEMPTING TO FORCE NON-EXISTENT DRIVE (NO SACK)/
0192	057674	044524	043516	052040	
0193	057702	020117	047506	041522	
0194	057710	020105	047516	026516	
0195	057716	054105	051511	042524	
0196	057724	052116	042040	044522	
0197	057732	042526	024040	047516	
0198	057740	051440	041501	024513	
0199	057746	000			
0200	057747	101	052124	046505	EM145: .ASCIZ /ATTEMPTING EXECUTION OF DESELECT DRIVE WITH IE RESET/
0201	057754	052120	047111	020107	
0202	057762	054105	041505	052125	
0203	057770	047511	020116	043117	
0204	057776	042040	051505	046105	
0205	060004	041505	020124	051104	

8206	060012	053111	020105	044527	
8207	060020	044124	044440	020105	
8208	060026	042522	042523	000124	
8209	060034	052101	042524	050115	EM146: .ASCIZ /ATTEMPTING TO EXECUTE AN ILLEGAL FUNCTION/
8210	060042	044524	043516	052040	
8211	060050	020117	054105	041505	
8212	060056	052125	020105	047101	
8213	060064	044440	046114	043505	
8214	060072	046101	043040	047125	
8215	060100	052103	047511	000116	
8216	060106	052101	042524	050115	EM147: .ASCIZ /ATTEMPTING TO CLEAR ILLEGAL FUNCTION/
8217	060114	044524	043516	052040	
8218	060122	020117	046103	040505	
8219	060130	020122	046111	042514	
8220	060136	040507	020114	052506	
8221	060144	041516	044524	047117	
8222	060152	000			
8223	060153	104	044522	042526	EM2000: .ASCIZ /DRIVE COMMAND BIT DID NOT SET IN DRIVE MESS A/
8224	060160	041440	046517	040515	
8225	060166	042116	041040	052111	
8226	060174	042040	042111	047040	
8227	060202	052117	051440	052105	
8228	060210	044440	020116	051104	
8229	060216	053111	020105	042515	
8230	060224	051523	040440	000	
8231	060231	104	044522	042526	EM2001: .ASCIZ /DRIVE MESS A INCORRECT/
8232	060236	046440	051505	020123	
8233	060244	020101	047111	047503	
8234	060252	051122	041505	000124	
8235	060260	051104	053111	020105	EM2002: .ASCIZ /DRIVE MESS B INCORRECT/
8236	060266	042515	051523	041040	
8237	060274	044440	041516	051117	
8238	060302	042522	052103	000	
8239	060307	103	046517	040515	EM2003: .ASCIZ /COMMAND AND STAES REG. 1 INCORRECT/
8240	060314	042116	040440	042116	
8241	060322	051440	040524	051525	
8242	060330	051040	043505	020056	
8243	060336	020061	047111	047503	
8244	060344	051122	041505	000124	
8245	060352	051104	053111	020105	EM2004: .ASCIZ /DRIVE SELECT CODE IN MESSAGE A INCORRECT/
8246	060360	042523	042514	052103	
8247	060366	041440	042117	020105	
8248	060374	047111	046440	051505	
8249	060402	040523	042507	040440	
8250	060410	044440	041516	051117	
8251	060416	042522	052103	000	
8252	060423	110	040505	020104	EM2005: .ASCIZ /HEAD ADD CODE IN MESSAGE A INCORRECT/
8253	060430	042101	020104	047503	
8254	060436	042504	044440	020116	
8255	060444	042515	051523	043501	
8256	060452	020105	020101	047111	
8257	060460	047503	051122	041505	
8258	060466	000124			
8259	060470	040515	047111	020124	EM2006: .ASCIZ /MAINT REG. 1 INCORRECT/
8260	060476	042522	027107	030440	
8261	060504	044440	041516	051117	

8262	060512	042522	052103	000	
8263	060517	115	051505	020123	EM2007: .ASCIZ /MESS SELECT CODE IN MESSAGE B INCORRECT/
8264	060524	042523	042514	052103	
8265	060532	041440	042117	020105	
8266	060540	047111	046440	051505	
8267	060546	040523	042507	041040	
8268	060554	044440	041516	051117	
8269	060562	042522	052103	000	
8270	060567	103	046131	047111	EM2008: .ASCIZ /CYLINDER ADD BITS IN MESSAGE B INCORRECT/
8271	060574	042504	020122	042101	
8272	060602	020104	044502	051524	
8273	060610	044440	020116	042515	
8274	060616	051523	043501	020105	
8275	060624	020102	047111	047503	
8276	060632	051122	041505	000124	
8277	060640	043117	051506	052105	EM2009: .ASCIZ /OFFSET VALUE BITS IN MESSAGE B INCORRECT/
8278	060646	053040	046101	042525	
8279	060654	041040	052111	020123	
8280	060662	047111	046440	051505	
8281	060670	040523	042507	041040	
8282	060676	044440	041516	051117	
8283	060704	042522	052103	000	
8284	060711	120	051101	052111	EM2010: .ASCIZ /PARITY BIT IN MESSAGE A INCORRECT/
8285	060716	020131	044502	020124	
8286	060724	047111	046440	051505	
8287	060732	040523	042507	040440	
8288	060740	044440	041516	051117	
8289	060746	042522	052103	000	
8290	060753	120	051101	052111	EM2011: .ASCIZ /PARITY BIT IN MESSAGE B INCORRECT/
8291	060760	020131	044502	020124	
8292	060766	047111	046440	051505	
8293	060774	040523	042507	041040	
8294	061002	044440	041516	051117	
8295	061010	042522	052103	000	
8296	061015	103	046517	040515	EM2012: .ASCIZ /COMMAND AND STATUS REG 2 INCORRECT/
8297	061022	042116	040440	042116	
8298	061030	051440	040524	052524	
8299	061036	020123	042522	020107	
8300	061044	020062	047111	047503	
8301	061052	051122	041505	000124	
8302	061060	051105	047522	020122	EM2013: .ASCIZ /ERROR REG INCORRECT/
8303	061066	042522	020107	047111	
8304	061074	047503	051122	041505	
8305	061102	000124			
8306	061104	047503	046515	047101	EM2014: .ASCIZ /COMMAND AND STATUS REG 1 INCORRECT AT PHASE ADDRESS 4/
8307	061112	020104	047101	020104	
8308	061120	052123	052101	051525	
8309	061126	051040	043505	030440	
8310	061134	044440	041516	051117	
8311	061142	042522	052103	040440	
8312	061150	020124	044120	051501	
8313	061156	020105	042101	051104	
8314	061164	051505	020123	000064	
8315	061172	047503	046515	047101	EM2015: .ASCIZ /COMMAND AND STATUS REG 1 INVALID DURING COMMAND EXECUTION/
8316	061200	020104	047101	020104	
8317	061206	052123	052101	051525	

8318	061214	051040	043505	030440
8319	061222	044440	053116	046101
8320	061230	042111	042040	051125
8321	061236	047111	020107	047503
8322	061244	046515	047101	020104
8323	061252	054105	041505	052125
8324	061260	047511	000116	
8325	061264	040515	047111	042524
8326	061272	040516	041516	020105
8327	061300	042522	020107	020062
8328	061306	047125	054105	042520
8329	061314	052103	042105	054514
8330	061322	041440	040510	043516
8331	061330	042105	042040	051125
8332	061336	047111	020107	047503
8333	061344	046515	047101	020104
8334	061352	054105	041505	052125
8335	061360	047511	000116	
8336	061364	040515	047111	042524
8337	061372	040516	041516	020105
8338	061400	042522	020107	020063
8339	061406	047125	054105	042520
8340	061414	052103	042105	054514
8341	061422	041440	040510	043516
8342	061430	042105	042040	051125
8343	061436	047111	020107	047503
8344	061444	046515	047101	020104
8345	061452	054105	041505	052125
8346	061460	047511	000116	
8347	061464	047111	042524	051122
8348	061472	050125	020124	044504
8349	061500	020104	047516	020124
8350	061506	041517	052503	000122
8351	061514	047503	046515	047101
8352	061522	020104	047101	020104
8353	061530	052123	052101	051525
8354	061536	051040	043505	030440
8355	061544	044440	041516	051117
8356	061552	042522	052103	040440
8357	061560	052106	051105	044440
8358	061566	052116	051105	052522
8359	061574	052120	000	
8360	061577	103	046517	040515
8361	061604	042116	040440	042116
8362	061612	051440	040524	052524
8363	061620	020123	042522	020107
8364	061626	020062	047111	047503
8365	061634	051122	041505	020124
8366	061642	043101	042524	020122
8367	061650	047111	042524	051122
8368	061656	050125	000124	
8369	061662	051105	047522	020122
8370	061670	042522	044507	052123
8371	061676	051105	044440	041516
8372	061704	051117	042522	052103
8373	061712	040440	052106	051105

EM2016: .ASCIZ /MAINTENANCE REG 2 UNEXPECTEDLY CHANGED DURING COMMAND EXECUTION/

EM2017: .ASCIZ /MAINTENANCE REG 3 UNEXPECTEDLY CHANGED DURING COMMAND EXECUTION/

EM2018: .ASCIZ /INTERRUPT DID NOT OCCUR/

EM2019: .ASCIZ /COMMAND AND STATUS REG 1 INCORRECT AFTER INTERRUPT/

EM2020: .ASCIZ /COMMAND AND STATUS REG 2 INCORRECT AFTER INTERRUPT/

EM2021: .ASCIZ /ERROR REGISTER INCORRECT AFTER INTERRUPT/

8374	061720	044440	052116	051105	
8375	061726	052522	052120	000	
8376	061733	111	052116	051105	EM2022: .ASCIZ /INTERRUPT DID NOT CLEAR IN RK611/
8377	061740	052522	052120	042040	
8378	061746	042111	047040	052117	
8379	061754	041440	042514	051101	
8380	061762	044440	020116	045522	
8381	061770	030466	000061		
8382	061774	040504	040524	046040	EM2023: .ASCIZ /DATA LATE DID NOT OCCUR WHEN LEAVING SILO/
8383	062002	052101	020105	044504	
8384	062010	020104	047516	020124	
8385	062016	041517	052503	020122	
8386	062024	044127	047105	046040	
8387	062032	040505	044526	043516	
8388	062040	051440	046111	000117	
8389	062046	051104	053111	020105	EM2024: .ASCIZ /DRIVE COMMAND BITS IN MESSAGE INCORRECT/
8390	062054	047503	046515	047101	
8391	062062	020104	044502	051524	
8392	062070	044440	020116	042515	
8393	062076	051523	043501	020105	
8394	062104	047111	047503	051122	
8395	062112	041505	000124		
8396	062116	051104	053111	020105	EM2025: .ASCIZ /DRIVE STATUS REGISTER INCORRECT/
8397	062124	052123	052101	051525	
8398	062132	051040	043505	051511	
8399	062140	042524	020122	047111	
8400	062146	047503	051122	041505	
8401	062154	000124			
8402	062156	047503	052116	047522	EM2026: .ASCIZ /CONTROLLER READY DID NOT SET/
8403	062164	046114	051105	051040	
8404	062172	040505	054504	042040	
8405	062200	042111	047040	052117	
8406	062206	051440	052105	000	
8407	062213	114	040517	020104	EM2027: .ASCIZ /LOAD STATUS DID NOT LOAD DRIVE STATUS REG./
8408	062220	052123	052101	051525	
8409	062226	042040	042111	047040	
8410	062234	052117	046040	040517	
8411	062242	020104	051104	053111	
8412	062250	020105	052123	052101	
8413	062256	051525	051040	043505	
8414	062264	000056			
8415	062266	047125	054105	042520	EM2028: .ASCIZ /UNEXPECTED INTERRUPT OCCURRED/
8416	062274	052103	042105	044440	
8417	062302	052116	051105	052522	
8418	062310	052120	047440	041503	
8419	062316	051125	042522	000104	
8420	062324	047111	042524	051122	EM2029: .ASCIZ /INTERRUPT OCCURRED WHEN INTERRUPT ENABLE SET/
8421	062332	050125	020124	041517	
8422	062340	052503	051122	042105	
8423	062346	053440	042510	020116	
8424	062354	047111	042524	051122	
8425	062362	050125	020124	047105	
8426	062370	041101	042514	051440	
8427	062376	052105	000		
8428		000001			.END

ABASE = 177440	138#	373	414
ABORT = 051270	6724	7583#	
ACDW1 = 000000	373	416	
ACDW2 = 000000	373	417	
ACLO = 000C10	233#	5173	
ACPUOP = 000000	373	388	
ADDW0 = 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	
AENV = 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	
APTCSU = 000040	6601#	6764	
APTENV = 000001	6557	6599#	6633 6757
APTSIZ = 000200	1602	6598#	
APTSPO = 000100	6559	6600#	6759
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#		







DZR6BB.P11		CROSS REFERENCE TABLE -- USER SYMBOLS														
DT065	050152	725	732	787	837	843	7335#									
DT067	050162	739	746	850	856	7337#										
DT100	050176	793	817	952	1444	1450	7339#									
DT126	050202	7340#														
DT224	050212	1300	1306	1312	1318	7342#										
DT230	050246	1324	1330	1336	1342	7347#										
DT256	050300	1456	1462	1467	1472	7352#										
ECCW =	020000	257#	1957	1959	2930	2932	2997	2999	3064	3066	3131	3133	3198	3200		
ECH =	000100	3265	3267	3325	3327											
EMTVEC =	000030	217#	1573*	1574*												
EMC00	053134	128#	7760#													
EM1N	001300	1475	7760#	1838*	1845*	1851*	1855*	2005*	2026*	2036*	2042*	2046*	2068*	2089*		
		436#	1821*	2109*	2131*	2152*	2162*	2168*	2172*	2193*	2210*	2217*	2223*	2227*		
		2099*	2105*	2266*	2272*	2276*										
		2242*	2259*													
EM100	053201	1821	7767#													
EM101	053267	2005	7777#													
EM102	053336	2068	7784#													
EM103	053400	2131	7790#													
EM104	053454	2193	7798#													
EM105	053523	2242	7805#													
EM106	053556	441	446	451	456	7810#										
EM107	053662	461	466	471	476	7822#										
EM108	053757	481	486	491	496	501	7833#									
EM109	054057	506	511	516	521	526	7844#									
EM110	054132	531	536	542	547	552	7852#									
EM111	054207	557	562	567	572	7860#										
EM112	054252	578	584	590	596	602	7866#									
EM113	054405	608	614	620	626	632	638	7883#								
EM114	054517	644	650	7897#												
EM115	054562	656	662	668	674	7903#										
EM116	054654	680	686	692	698	7913#										
EM117	054747	704	710	716	926	7923#										
EM118	055064	723	730	737	744	750	756	762	932	7937#						
EM119	055177	768	774	780	938	7950#										
EM120	055266	785	7960#													
EM121	055362	791	797	803	809	815	7970#									
EM122	055463	822	829	7981#												
EM123	055564	835	841	848	854	860	866	872	944	7992#						
EM124	055650	878	884	890	896	902	908	914	920	8001#						
EM125	055716	950	956	962	968	974	980	8008#								
EM126	056135	986	992	998	1004	8035#										
EM127	056202	1010	1016	1022	1028	8042#										
EM128	056300	1034	1040	1046	1052	8053#										
EM129	056361	1058	1064	1070	1076	8062#										
EM130	056426	1082	1088	1094	1100	8069#										
EM131	056510	1106	1112	1118	1124	8078#										
EM132	056575	1130	1136	1142	1148	8087#										
EM133	056634	1154	1160	1166	1172	8093#										
EM134	056700	1178	1184	1190	1196	8099#										
EM135	056745	1202	1208	1214	1220	8106#										
EM136	057011	1226	1232	1238	1244	8113#										
EM137	057065	1250	1256	1262	1268	8121#										
EM138	057145	1274	1280	1286	1292	8130#										
EM139	057255	1298	1304	1310	1316	8143#										
EM140	057337	1322	1328	1334	1340	8152#										

EM141	057421	1346	1352	1358	1364	8161#									
EM142	057475	1370	1376	1382	1388	8169#									
EM143	057573	1394	1400	1406	1412	8181#									
EM144	057666	1418	1424	1430	1436	8191#									
EM145	057747	1442	1448	8200#											
EM146	060034	1454	1460	8209#											
EM147	060106	1465	1470	8216#											
EM2000	060153	512	537	585	621	1845	2036	2099	2162	2217	2266	8223#			
EM2001	060231	452	472	497	522	548	568	573	597	633	645	669	693	769	
EM2002	060260	1851	2042	2105	2168	2223	2272	8231#							
EM2003	060307	457	477	502	527	553	603	639	651	675	699	1855	2046	2109	
		2172	2227	2276	8235#										
		442	462	482	507	532	558	579	609	705	751	786	861	879	
		963	987	1011	1035	1059	1083	1107	1131	1155	1179	1203	1227	1251	
		1275	1299	1323	1347	1371	1395	1419	1455	1466	1838	2026	2089	2152	
		2210	2259	8239#											
EM2004	060352	447	885	8245#											
EM2005	060423	467	897	8252#											
EM2006	060470	487	615	8259#											
EM2007	060517	492	627	909	8263#										
EM2008	060567	517	543	591	915	8270#									
EM2009	060640	563	8277#												
EM2010	060711	657	681	903	8284#										
EM2011	060753	663	687	921	8290#										
EM2012	061015	711	757	775	823	867	969	993	1017	1041	1065	1089	1113	1137	
		1161	1185	1209	1233	1257	1281	1305	1329	1353	1377	1401	1425	8296#	
EM2013	061060	717	763	781	873	975	1005	1029	1053	1077	1101	1125	1149	1173	
		1197	1221	1245	1269	1293	1317	1341	1365	1389	1413	1437	1461	1471	
		8302#													
EM2014	061104	724	836	8306#											
EM2015	061172	731	842	8315#											
EM2016	061264	738	849	8325#											
EM2017	061364	745	855	8336#											
EM2018	061464	792	8347#												
EM2019	061514	798	8351#												
EM2020	061577	804	8360#												
EM2021	061662	810	8369#												
EM2022	061733	816	8376#												
EM2023	061774	830	8382#												
EM2024	062046	891	8389#												
EM2025	062116	927	933	939	945	981	999	1023	1047	1071	1095	1119	1143	1167	
		1191	1215	1239	1263	1287	1311	1335	1359	1383	1407	1431	8396#		
EM2026	062156	951	8402#												
EM2027	062213	957	8407#												
EM2028	062266	1443	8415#												
EM2029	062324	1449	8420#												
ERRCNT	004242	1533#	1606#	6719*	6722										
ERRVEC	000004	121#	1587	1588*	1599*	6365*	6366*	6378*	6379*	6416	6417*	6419*	6422*		
E.ASOF	004176	1507#													
E.BA	004164	1502#													
E.CS1	004160	1500#	1706*	1707	1720*	1723	1766*	1785	1825*	1836	1873*	1899	1937*	1962	
		2006*	2014	2024	2049	2051*	2069*	2077	2087	2112	2114*	2132*	2140	2150	
		2175	2177*	2197*	2208	2246*	2257	2293*	2316	2361*	2379	2424*	2447	2494*	
		2524	2565*	2581	2618*	2634	2678*	2694	2738*	2754	2798*	2814	2858*	2874	
		2915*	2935	2982*	3002	3049*	3069	3116*	3136	3183*	3203	3250*	3270	3310*	
		3330	3798*	3803	3850*	3851	3878	3904*	3909	3957*	3962	4024*	4027	4097*	



MPF. SPAR = 004216 . WC = 004162 MTE = 000020 GNS = ***** U  GO = 000001  GTSWR = 104406 HCODE = 004250  HT = 000011 HVRC = 000400 IDAE = 002000 IE = 000100 ILF = 000001 ILLFUN = 004270 INTR = 000300 TOTVEC = 000020 TR = 000100  LF = 000012 MCLK = 000400  MDS = 001000 MEMBAS = 172100 MEMERR = 043612 MEMVEC = 000114 MERD = 001000 MESMSK = 000017 MEWD = 002000 MIND = 000200 MSGCOD = 004246  MSP = 000100 NED = 010000 NEM = 004030 NEWPAS = 005270 NXF = 000004 OFFSET = 000015	2107 2337 2513* 2533* 2631 2640* 7208 1515* 1501* 215* 277 7295 178* 5017 6062 1621 1536* 5566 31* 219* 221* 179* 211* 1544* 174* 126* 198* 4426 4940 5444 6008 32* 252* 2371 3055 3448 4134 5003 6048 201* 134* 6371 133* 253* 246* 254* 251* 1535* 3249* 250* 204* 203* 1680* 213* 167*	2149* 2360* 2514* 2657 3068* 3415* 3672* 7314 1626 7296 4183 5095 6284 7290* 1872* 5648* 6772 5252 4007 6286 6277* 1571* 3800 4475 4966 5503 6063 6807 1713 2434 3056 3494 4149 5081 6124 6367 6384* 6371* 1956 1936* 3309* 6145 6359 5099 2494	2170 2392 2515* 2693* 3098 3439* 3684* 7316 6336 7297 4234 5171 6284 1881 5650 6813 5336 4024 6283 6288 1572* 3906 4501 5018 5529 6088 6813 1776 2505 3122 3524 4175 5157 6209 6372* 2929 1943 3638* 6224 2503	2207* 2400 2516* 2717 3135* 3458 3730* 7319 6343 7298 4242 5249 6288 1893 5732* 6288 3959 4552 5044 5587 6145 1827 2572 3123 3576 4223 5235 1885 2625 3189 3577 4308 5318 3063 1955 3647 3250 3254	2225 2407* 2517* 2753* 3165 3463 3736* 7324 7284 7285 4322 5332 1919* 5734 5757 6309* 4025 4578 5096 5613 6170 1947 2685 3190 3577 4385 5402 3130 1976 7311 3254	2256* 2442* 2518* 2777 3202* 3471* 3751 7326 7286 4474 4399 5416 5501 5816* 5818 5841 6310 4098 4630 5122 5671 6224 1948 2745 3190 3652 4385 5487 3197 1988* 1989 3264	2311* 2443* 2519* 2813* 3269* 3501* 3864* 7329 7287 4551 5585 5311* 5819 5925 7352 4184 4656 5172 5697 6249 2016 2805 3257 3715 4537 5571 3264 1989 3324	2312* 2445* 2523* 2873* 3292 3532* 3888 7337 7288 4629 5669 5395* 5902 5313 5900* 5925 4243 4785 5250 5781 6249 2079 2865 3316 3716 4615 5655 3324	2313* 2446* 2532* 2837 3292 3532* 3888 7337 7290 4707 5753 5397 7308 5397 7308 4272 4785 5276 5839 6249 2142 2921 3317 3790 4693 5739 3324	2314* 2450 2540 2934* 3359 3591* 4166 7337 7292 4784 5837 5480* 7347 5482 4323 4811 5334 5865 6249 2199 2922 3391 3845 4770 5823 3324	2315* 2468 2580* 2964 3384* 3597* 7302 7302 7293 4861 5921 5482 5482 4349 4862 5360 5923 6249 2248 2988 3392 3871 4847 5907 3115*	2329 2512* 2597 3001* 3402 3612 7305 7294 4939 6006 5564* 4400 4888 5418 5949 2303 2989 3447 3896 4925 5992 3182*
---	---	---	---	---	--	---	--	---	--	--	---	--

OFFVAL = 004254	1538#	2493*	2502	2513	2521	2544*	2545	2564*	2617*	2677*	2737*	2797*	2857*
OFST = 000004	7319	7322											
OPI = 020000	232#	4244	4246										
OPR001 = 051156	224#												
OPR002 = 051205	1630	7567#											
OPR003 = 051213	1633	1644	1661	7571#									
OPR004 = 051243	1640	7573#											
OR = 000200	1652	7578#											
PACK = 000003	199#												
PARBIT = 004260	162#	2132	2618	2623	2915	2919	4923	4939					
PARM = 004276	1540#	3592*	3595*	3600	3607	3641*	3660	3667	3685*	3731*	3734*	3739	3746
PAR.EN = 000001	284	1549#											
PAT = 000020	135#	6370											
PC = %000007	248#	3520	3524	3525	3710								
	52#	1680*	6328*	6331*	6353*	6358	6380*	6576*	6593*	6630*	6636*	6733*	6762*
	6781*	6788*	6795*	6809*	6811*	7011*	7245*						
PCA = 004000	255#												
PCD = 010000	256#												
PGE = 002000	202#												
PIP = 020000	240#	5419											
PIRQ = 177772	38#												
PIRQVE = 000240	132#												
PR0 = 000000	55#												
PR1 = 000040	56#												
PR2 = 000100	57#												
PR3 = 000140	58#												
PR4 = 000200	59#												
PR5 = 000240	60#	1528											
PR6 = 000300	61#												
PR7 = 000340	62#	1558	1681	4001	4013	4050	6366	6372	7229	7242			
PS = 177776	35#	36											
PSW = 177776	36#												
PWRVEC = 000024	127#	1577*	1578*	7228*	7229*	7241*	7242*						
P.CS1 = 004220	1519#	4261*	4338*	4415*	4490*	4567*	4645*	4723*	4800*	4877*	4955*	5033*	5111*
	5187*	5265*	5349*	5433*	5518*	5602*	5686*	5770*	5854*	5938*	6078*	6160*	6239*
	7345												
P.CS2 = 004222	1520#	4262*	4339*	4416*	4491*	4568*	4646*	4724*	4801*	4878*	4956*	5034*	5112*
	5188#	5266*	5350*	5434*	5519*	5603*	5687*	5771*	5855*	5939*	6079*	6161*	6240*
	7345												
P.DS = 004224	1521#	4263*	4340*	4417*	4492*	4569*	4647*	4725*	4802*	4879*	4957*	5035*	5113*
	5189#	5267*	5351*	5435*	5520*	5604*	5688*	5772*	5856*	5940*	6080*	6162*	6241*
	7345												
P.ER = 004226	1522#	4264*	4341*	4418*	4493*	4570*	4648*	4726*	4803*	4880*	4958*	5036*	5114*
	5190#	5268*	5352*	5436*	5521*	5605*	5689*	5773*	5857*	5941*	6081*	6163*	6242*
	7345												
RDCHR = 104410	7071	7293#											
RDDATA = 000021	169#												
RDGATE = 100000	259#												
RDHEAD = 000025	171#												
RDLIN = 104411	7143	7294#											
RDOCT = 104412	1634	1645	1662	7295#									
RDY = 000200	180#	3798	3904	3957	4024	4097	4108	4183	4234	4242	4271	4322	4348
	4399	4425	4474	4500	4551	4577	4629	4655	4707	4733	4784	4810	4861
	4887	4939	4965	5017	5043	5095	5121	5171	5197	5249	5275	5332	5359
	5416	5443	5501	5528	5585	5612	5669	5696	5753	5780	5837	5864	5921
	5948	6006	6062	6087	6141	6169	6223	6248	6285	6300			

# D15

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P2 MD-11-DZR68-B  
 DZR688.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 01-OCT-76 10:23 PAGE 166

SEQ 0167

RECAL = 000013  
 RESREG = 104414  
 RESTRT = 004306  
 RESVEC = 000010  
 RKASOF = 000016  
 RKBA = 000004  
 RKCS1 = 000000

166#	2196	2197	2858	2863	3183	3187	5233	5249					
6718	7297#												
282	1552#												
122#													
149#	2502*	2569*	2622*	2682*	2742*	2802*	2862*						
144#													
142#	1699*	1701*	1705	1717	1771*	1774*	1780	1788*	1822*	1824*	1831	1840*	
1878*	1883*	1889	1902*	1942*	1945*	1951	1965*	1971*	2012*	2014*	2020	2028*	
2075*	2077*	2083	2091*	2138*	2140*	2146	2154*	2194*	2196*	2203	2212*	2243*	
2245*	2252	2261*	2298*	2301*	2307	2319*	2366*	2369*	2375	2382*	2429*	2432*	
2438	2450*	2500*	2503*	2509	2527*	2566*	2570*	2576	2584*	2619*	2623*	2629	
2637*	2679*	2683*	2689	2697*	2739*	2743*	2749	2757*	2799*	2803*	2809	2817*	
2859*	2863*	2869	2877*	2916*	2919*	2925	2938*	2944*	2983*	2986*	2992	3005*	
3011*	3050*	3053*	3059	3072*	3078*	3117*	3120*	3126	3139*	3145*	3184*	3187*	
3193	3206*	3212*	3251*	3254*	3260	3273*	3279*	3311*	3314*	3320	3333*	3339*	
3380*	3389*	3435*	3444*	3490*	3492*	3519*	3522*	3570*	3574*	3646*	3650*	3708*	
3713*	3788*	3794	3843*	3849	3875	3900	3947*	3949	3953	4007*	4009	4021	
4045*	4048*	4087*	4089	4093	4105	4113*	4132*	4138	4153	4179	4221*	4229	
4233	4238	4266*	4267	4306*	4314	4318	4343*	4344	4383*	4391	4395	4420*	
4421	4458*	4466	4470	4495*	4496	4535*	4543	4547	4572*	4573	4613*	4621	
4625	4650*	4651	4691*	4699	4703	4728*	4729	4768*	4776	4780	4805*	4806	
4845*	4853	4857	4882*	4883	4923*	4931	4935	4960*	4961	5001*	5009	5013	
5038*	5039	5079*	5087	5091	5116*	5117	5155*	5163	5167	5192*	5193	5233*	
5241	5245	5270*	5271	5316*	5324	5328	5354*	5355	5400*	5408	5412	5438*	
5439	5485*	5493	5497	5523*	5524	5569*	5577	5581	5607*	5608	5653*	5661	
5665	5691*	5692	5737*	5745	5749	5775*	5776	5821*	5829	5833	5859*	5860	
5905*	5913	5917	5943*	5944	5990*	5998	6002	6046*	6054	6058	6082*	6083	
6122*	6130	6137	6164*	6165	6207*	6215	6219	6243*	6244	6282*	6288*	6289	
6297*	6298												

RKCS2 = 000010

146#	1773*	3388*	3573*	3649*	3712*	3785*	3787*	3795	3840*	3842*	3901	3945*	
3946*	3954	3997*	3998*	4022	4044*	4065*	4084*	4086*	4094	4106	4128*	4180	
4217*	4239	4268	4302*	4319	4345	4379*	4396	4422	4456*	4471	4497	4531*	
4548	4574	4609*	4626	4652	4687*	4704	4730	4764*	4781	4807	4841*	4858	
4884	4919*	4936	4962	4997*	5014	5040	5075*	5092	5118	5153*	5168	5194	
5229*	5246	5272	5307*	5329	5356	5391*	5413	5440	5476*	5498	5525	5560*	
5582	5609	5644*	5666	5693	5728*	5750	5777	5812*	5834	5861	5896*	5918	
5945	5985*	5989*	6003	6042*	6045*	6059	6084	6120*	6138	6166	6205*	6220	
6245													

RKDA = 000006

145#	1882*	3443*	4131*	4220*	4305*	4382*	4534*	4612*	4690*	4767*	4844*	4922*	
5000*	5078*	5232*	5314*	5398*	5483*	5567*	5651*	5735*	5819*	5903*	5988*		

RKDB = 000024

RKDCYL = 000020

151#	4085*	4104											
150#	2300*	2368*	2431*	2568*	2621*	2681*	2741*	2801*	2861*	3442*	4130*	4219*	
4304*	4381*	4533*	4611*	4689*	4766*	4843*	4921*	4999*	5077*	5231*	5309*	5393*	
5478*	5562*	5646*	5730*	5814*	5898*	5987*	6044*						

RKDS = 000012

147#	3796	3902	3955	4095	4181	4240	4269	4320	4346	4397	4423	4472	
4498	4549	4575	4627	4653	4705	4731	4782	4808	4859	4885	4937	4963	
5015	5041	5093	5119	5169	5195	5247	5273	5330	5357	5414	5441	5499	
5526	5583	5610	5667	5694	5751	5778	5835	5862	5919	5946	6004	6060	
6085	6139	6167	6221	6246									

RKECP5 = 000030

RKECPT = 000032

RKER = 000014

155#													
156#													
148#	3797	3903	3956	4023	4096	4107	4182	4241	4270	4321	4347	4398	
4424	4473	4499	4550	4576	4628	4654	4706	4732	4783	4809	4860	4886	
4938	4964	5016	5042	5094	5120	5170	5196	5248	5274	5331	5358	5415	
5442	5500	5527	5584	5611	5668	5695	5752	5779	5836	5863	5920	5947	
6005	6061	6086	6140	6168	6222	6247	6290	6299					



F15

1889	1890	1891	1902*	1935*	1942*	1943*	1944*	1945*	1947*	1948*	1951	1952
1953	1954	1965*	1971*	2004*	2012*	2013*	2014*	2016*	2017*	2020	2021	2022
2028*	2067*	2075*	2076*	2077*	2079*	2080*	2083	2084	2085	2091*	2130*	2138*
2139*	2140*	2142*	2143*	2146	2147	2148	2154*	2192*	2194*	2195*	2196*	2199*
2200*	2203	2204	2205	2212*	2241*	2243*	2244*	2245*	2248*	2249*	2252	2253
2254	2261*	2291*	2298*	2299*	2300*	2301*	2303*	2304*	2307	2308	2309	2319*
2358*	2366*	2367*	2368*	2369*	2371*	2372*	2375	2376	2377	2382*	2422*	2429*
2430*	2431*	2432*	2434*	2435*	2438	2439	2440	2450*	2492*	2500*	2501*	2502*
2503*	2505*	2506*	2509	2510	2511	2527*	2562*	2566*	2567*	2568*	2569*	2570*
2572*	2573*	2576	2577	2578	2584*	2615*	2619*	2620*	2621*	2622*	2623*	2625*
2626*	2629	2630	2631	2637*	2675*	2679*	2680*	2681*	2682*	2683*	2685*	2686*
2689	2690	2691	2697*	2735*	2739*	2740*	2741*	2742*	2743*	2745*	2746*	2749
2750	2751	2757*	2795*	2799*	2800*	2801*	2802*	2803*	2805*	2806*	2809	2810
2811	2817*	2855*	2859*	2860*	2861*	2862*	2863*	2865*	2866*	2869	2870	2871
2877*	2913*	2916*	2917*	2919*	2921*	2922*	2925	2926	2927	2928	2938*	2944*
2980*	2983*	2984*	2986*	2988*	2989*	2992	2993	2994	2995	3005*	3011*	3047*
3050*	3051*	3053*	3055*	3056*	3059	3060	3061	3062	3072*	3078*	3114*	3117*
3118*	3120*	3122*	3123*	3126	3127	3128	3129	3139*	3145*	3181*	3184*	3185*
3187*	3189*	3190*	3193	3194	3195	3196	3206*	3212*	3248*	3251*	3252*	3254*
3256*	3257*	3260	3261	3262	3263	3273*	3279*	3308*	3311*	3312*	3314*	3316*
3317*	3320	3321	3322	3323	3333*	3339*	3379*	3380*	3386*	3388*	3389*	3391*
3392*	3395	3396	3434*	3435*	3441*	3442*	3443*	3444*	3447*	3448*	3451	3452
3489*	3490*	3491*	3492*	3494*	3495*	3498	3499	3519*	3520*	3522*	3524*	3525*
3529	3530	3564*	3570*	3571*	3572*	3573*	3574*	3576*	3577*	3580	3581	3636*
3646*	3647*	3648*	3649*	3650*	3652*	3653*	3656	3657	3702*	3708*	3709*	3710*
3712*	3713*	3715*	3716*	3719	3720	3779*	3785*	3786*	3787*	3788*	3790*	3791*
3794	3795	3796	3797	3834*	3840*	3841*	3842*	3843*	3845*	3846*	3849	3871*
3872*	3875	3876	3877	3896*	3897*	3900	3901	3902	3903	3939*	3945*	3946*
3947*	3949	3953	3954	3955	3956	3996*	3997*	3998*	4007*	4009	4021	4022
4023	4044*	4045*	4048*	4065*	4083*	4084*	4085*	4086*	4087*	4089	4093	4094
4095	4096	4104	4105	4106	4107	4113*	4127*	4128*	4129*	4130*	4131*	4132*
4134*	4135*	4138	4149*	4150*	4153	4154	4155	4175*	4176*	4179	4180	4181
4182	4216*	4217*	4218*	4219*	4220*	4221*	4223*	4224*	4227*	4229	4233	4238
4239	4240	4241	4266*	4267	4268	4269	4270	4301*	4302*	4303*	4304*	4305*
4306*	4308*	4309*	4312*	4314	4318	4319	4320	4321	4343*	4344	4345	4346
4347	4378*	4379*	4380*	4381*	4382*	4383*	4385*	4386*	4389*	4391	4395	4396
4397	4398	4420*	4421	4422	4423	4424	4455*	4456*	4457*	4458*	4460*	4461*
4464*	4466	4470	4471	4472	4473	4495*	4496	4497	4498	4499	4530*	4531*
4532*	4533*	4534*	4535*	4537*	4538*	4541*	4543	4547	4548	4549	4550	4572*
4573	4574	4575	4576	4608*	4609*	4610*	4611*	4612*	4613*	4615*	4616*	4619*
4621	4625	4626	4627	4628	4650*	4651	4652	4653	4654	4686*	4687*	4688*
4689*	4690*	4691*	4693*	4694*	4697*	4699	4703	4704	4705	4706	4728*	4729
4730	4731	4732	4763*	4764*	4765*	4766*	4767*	4768*	4770*	4771*	4774*	4776
4780	4781	4782	4783	4805*	4806	4807	4808	4809	4840*	4841*	4842*	4843*
4844*	4845*	4847*	4848*	4851*	4853	4857	4858	4859	4860	4882*	4883	4884
4885	4886	4918*	4919*	4920*	4921*	4922*	4923*	4925*	4926*	4929*	4931	4935
4936	4937	4938	4960*	4961	4962	4963	4964	4996*	4997*	4998*	4999*	5000*
5001*	5003*	5004*	5007*	5009	5013	5014	5015	5016	5038*	5039	5040	5041
5042	5074*	5075*	5076*	5077*	5078*	5079*	5081*	5082*	5085*	5087	5091	5092
5093	5094	5116*	5117	5118	5119	5120	5152*	5153*	5154*	5155*	5157*	5158*
5161*	5163	5167	5168	5169	5170	5192*	5193	5194	5195	5196	5228*	5229*
5230*	5231*	5232*	5233*	5235*	5236*	5239*	5241	5245	5246	5247	5248	5270*
5271	5272	5273	5274	5306*	5307*	5308*	5309*	5314*	5316*	5318*	5319*	5322*
5324	5328	5329	5330	5331	5354*	5355	5356	5357	5358	5390*	5391*	5392*
5393*	5398*	5400*	5402*	5403*	5406*	5408	5412	5413	5414	5415	5438*	5439
5440	5441	5442	5475*	5476*	5477*	5478*	5483*	5485*	5487*	5488*	5491*	5493

R3 =%000003

R4 =%000004

R5 =%000005

R6 =%000006

R7 =%000007

SAVREG= 104413

SAVSWR 004274

SCLR = 000040

SCOPI = 104415

SCOPI\$ 044372

SEEK = 000017

SELDRV= 000001

SFTCNT 004256

SKI = 000002

SP =%000006

5497	5498	5499	5500	5523*	5524	5525	5526	5527	5559*	5560*	5561*	5562*
5567*	5569*	5571*	5572*	5575*	5577	5581	5582	5583	5584	5607*	5608	5609
5610	5611	5643*	5644*	5645*	5646*	5651*	5653*	5655*	5656*	5659*	5661	5665
5666	5667	5668	5691*	5692	5693	5694	5695	5727*	5728*	5729*	5730*	5735*
5737*	5739*	5740*	5743*	5745	5749	5750	5751	5752	5775*	5776	5777	5778
5779	5811*	5812*	5813*	5814*	5819*	5821*	5823*	5824*	5827*	5829	5833	5834
5835	5836	5859*	5860	5861	5862	5863	5835*	5896*	5897*	5898*	5903*	5905*
5907*	5908*	5911*	5913	5917	5918	5919	5920	5943*	5944	5945	5946	5947
5979*	5985*	5986*	5987*	5988*	5989*	5990*	5992*	5993*	5996*	5998	6002	6003
6004	6005	6041*	6042*	6043*	6044*	6045*	6046*	6048*	6049*	6052*	6054	6058
6059	6060	6061	6082*	6083	6084	6085	6086	6119*	6120*	6121*	6122*	6124*
6125*	6128*	6130	6137	6138	6139	6140	6164*	6165	6166	6167	6168	6204*
6205*	6206*	6207*	6209*	6210*	6213*	6215	6219	6220	6221	6222	6243*	6244
6245	6246	6247	6282*	6287*	6288*	6289	6290	6297*	6298	6299	6689*	6707*
6905	6916*	6920*	6923	6930*	6931*	6932	6937*	6945*	7142	7147*	7155*	7157*
7159*	7165	7166*	7197	7218*								
46#	3583*	3588*	3722*	3727*	3856*	3859*	6367*	6370*	6691*	6693	6699*	6846
6855*	6861*	6862*	6865*	6870*	6871*	6872	6881*	6906	6914*	6915*	6929*	6932*
6941*	6942*	6944*	7066	7068*	7069	7072*	7073	7080*	7081	7083	7091	7095
7097*	7103	7105	7107*	7110*	7198	7217*						
47#	3584*	3587*	3593	3723*	3726*	3732	6369*	6373*	6687*	6695	6706*	6711
6847	6849*	6850*	6851*	6852	6853*	6867	6869*	6877*	6880*	7199	7216*	
48#	6848	6854*	6856*	6858*	6859*	6860*	6861	6879*	6907	6909*	6911*	6918*
6922*	6937	6943*	7200	7215*								
49#	51	1565*	1566*	1567								
50#	52											
6667	7296#											
1546#	7227*	7246										
197#	3785	3840	3945	3997	4065	4084	4128	4217	4302	4379	4476	4531
4609	4687	4764	4841	4919	4997	5075	5153	5229	5307	5391	5476	5560
5644	5728	5812	5896	5985	6042	6120	6205					
1804	1918	1987	2048	2111	2174	2340	2403	2471	2543	3618	3678	3757
3815	3921	3974	6023	6308	7298#							
6539#	7298											
168#	2293	2301	2361	2369	2424	2432	3310	3314	3444	4132	4139	4183
4221	4234	4242	4306	4322	4383	4399	4535	4551	4613	4629	4691	4707
4768	4784	4845	4861	5316	5332	5400	5416	5485	5501	5569	5585	5653
5669	5737	5753	5821	5837	5905	5921	5990	6006	6046	6062		
161#	1701	1706	1720	1766	1774	1824	1825	1873	1883	1937	1945	2565
2570	3389	3492	3522	3574	3650	3713	3788	3843	3850	3947	4007	4045
4087	4458	4474	6122	6207								
1539#	3381*	3417*	3418	3436*	3473*	3474	7326					
212#	5020											
51#	1557*	1558*	1559*	1569*	1587*	1595*	1599	1631*	1635	1641*	1642*	1646
1653*	1654*	1655*	1656*	1657*	1658*	1659*	1663	1681*	1682*	1880*	1881*	1882
4002*	4003*	4013*	4014*	4020*	4037*	4038*	4050*	4051*	4054*	4058*	4062*	5312*
5313*	5314	5396*	5397*	5398	5481*	5482*	5483	5565*	5566*	5567	5649*	5650*
5651	5733*	5734*	5735	5817*	5818*	5819	5901*	5902*	5903	6338*	6345*	6377
6385	6392*	6416*	6419	6421	6422	6427*	6428*	6430	6432	6433*	6434*	6435*
6436	6438	6461	6462	6466*	6543*	6553*	6554*	6561	6562*	6573	6574*	6575*
6585	6586*	6591	6592	6625	6646*	6649*	6697*	6729*	6755*	6756	6766*	6768
6769	6770*	6772	6774	6776	6782	6784*	6786*	6794*	6798	6802	6803	6807
6838*	6839	6840	6841*	6846*	6847*	6848*	6854	6879	6880	6881	6882*	6883*
6903*	6904*	6905*	6906*	6907*	6908*	6909	6912*	6925	6927*	6929	6939	6941
6943	6944	6945	6946	6947	6949*	6950*	6971*	6972*	6973	6980*	6983*	6984*
6988*	6989*	6993	6996*	7000	7002	7004	7005*	7012	7014	7016*	7017	7019*



TKVEC =	000060	130#							
TPVEC =	000064	131#							
TRAPPC	004272	1545#	6385*	7301					
TRAPVE=	000034	129#	1575*	1576*					
TRTVEC=	000014	124#							
TST1	005306	1682	1696#	6470					
TST10	010046	2113	2128#	6477					
TST100	042000	6101	6117#	6533					
TST101	042442	6183	6202#	6534					
TST102	043054	6262	6275#	6535					
TST11	010364	2176	2190#	6478					
TST12	010630	2213	2219	2226	2239#	6479			
TST13	011074	2262	2268	2275	2289#	6480			
TST14	011412	2356#	6481						
TST15	011722	2405	2420#	6482					
TST16	012254	2490#	6483						
TST17	012616	2560#	6484						
TST2	005612	1710	1726	1748	1762#	6471			
TST20	013062	2585	2592	2598	2613#	6485			
TST21	013342	2638	2645	2652	2658	2673#	6486		
TST22	013622	2698	2705	2712	2718	2733#	6487		
TST23	014102	2758	2765	2772	2778	2793#	6488		
TST24	014362	2818	2825	2832	2838	2853#	6489		
TST25	014642	2878	2885	2892	2898	2911#	6490		
TST26	015154	2939	2945	2952	2959	2965	2978#	6491	
TST27	015466	3006	3012	3019	3026	3032	3045#	6492	
TST3	006072	1818#	6472						
TST30	016000	3073	3079	3086	3093	3099	3112#	6493	
TST31	016312	3140	3146	3153	3160	3166	3179#	6494	
TST32	016624	3207	3213	3220	3227	3233	3246#	6495	
TST33	017122	3274	3280	3287	3293	3306#	6496		
TST34	017434	3334	3340	3347	3354	3360	3377#	6497	
TST35	017672	3400	3405	3432#	6498				
TST36	020136	3456	3461	3487#	6499				
TST37	020472	3536	3541	3547	3562#	6500			
TST4	006336	1841	1847	1854	1869#	6473			
TST40	021036	3634#	6501						
TST41	021362	3680	3700#	6502					
TST42	021726	3777#	6503						
TST43	022202	3832#	6504						
TST44	022732	3924	3937#	6505					
TST45	023172	3994#	6506						
TST46	023560	4081#	6507						
TST47	024010	4125#	6508						
TST5	006650	1933#	6474						
TST50	024444	4143	4159	4164	4169	4197	4214#	6509	
TST51	025126	4285	4299#	6510					
TST52	025560	4362	4376#	6511					
TST53	026212	4439	4453#	6512					
TST54	026630	4514	4528#	6513					
TST55	027262	4591	4606#	6514					
TST56	027714	4669	4684#	6515					
TST57	030346	4747	4761#	6516					
TST6	007212	2002#	6475						
TST60	031000	4824	4838#	6517					
TST61	031432	4901	4916#	6518					





L15

CROSS REFERENCE TABLE -- USER SYMBOLS		236#	4244	4246											
VV = 000100		236#	4244	4246											
WAITIM 004262		1541#	3948	4008	4088	4228	4313	4390	4465	4542	4620	4698	4775	4852	
		4930	5008	5086	5162	5240	5323	5407	5492	5576	5660	5744	5828	5912	
		5997	6053	6129	6214										
WCE = 040000		206#													
WLE = 004000		222#	4942												
WRDATA= 000023		170#													
WRHEAD= 000027		172#													
WRL = 004000		239#	4244	4246											
WRTCHK= 000031		173#													
WRTGAT= 040000		258#													
\$APTHD 001000		305	311#												
\$ASTAT= ***** U		6579	6594												
\$ATYC 044442		6550	6552#												
\$ATY1 044416		6548#													
\$ATY3 044424		6549#	6762												
\$ATY4 044434		6551#	6636												
\$AUTOB 001134		342#	1623*	6975	7124										
\$BASE 001270		414#	1631	1639*	1698	1764	1820	1871	1935	2004	2067	2130	2192	2241	
		2291	2358	2422	2492	2562	2615	2675	2735	2795	2855	2913	2980	3047	
		3114	3181	3248	3308	3379	3434	3489	3564	3636	3702	3779	3834	3939	
		3996	4083	4127	4216	4301	4378	4455	4530	4608	4686	4763	4840	4918	
		4996	5074	5152	5228	5306	5390	5475	5559	5643	5727	5811	5895	5979	
		6041	6119	6204											
\$BDADR 001122		337#													
\$BDAT 001126		339#													
\$BELL 001204		365#	6623	6656											
\$CDW1 001274		416#													
\$CDW2 001276		417#													
\$CHARC 045624		6779*	6789*	6796	6805*	6810#									
\$CKSWR 046302		6967#	7292												
\$CMTAG 001100		325#	1564	1565	1573	1579	1580	1581							
\$CM3 = 000000		355#													
\$CM4 = 000010		355#	356#	357#	358#	359#	360#	361#	362#	363#					
\$CNTLG 047155		6978	7119#												
\$CNTLU 047150		6995	7093	7118#											
\$CPUOP 001242		388#													
\$CRLF 001211		367#	6348	6631	6656	6677	6682	6690	6703	6716	6778	6813	7006	7098	
		7118	7177												
\$DBLK 046272		6914	6948	6956#											
\$DEVCT 001224		379#													
\$DEVN 001272		415#													
\$DOAGN 043512		6330	6351	6357#											
\$DTBL 046262		6917	6952#												
\$ENDAD 043502		291	1611	6353#	6651										
\$ENDCT 043346		1579	6332#												
\$ENULL 043516		6360#													
\$ENV 001234		384#	1617	6557	6581	6633	6757								
\$ENVN 001235		385#	1602	6559	6759	6764									
\$EOP 043312		6322#	6730												
\$EOPCT 043340		1579*	6329#	6333	6728*										
\$ERFLG 001103		328#	6400	6439	6441	6447*	6469	6541	6618*	6656					
\$ERMAX 001115		334#	1582*	6441	6464*	6469									
\$ERROR 044664		1573	6616#												
\$ERRPC 001116		335#	6625*	6626*	6627	6656	7302	7305	7308	7311	7316	7319	7322	7326	
		7329	7331	7335	7337	7339	7340	7342	7347	7352					

\$ERRTB	001300	434#	6674																		
\$ERTTL	001112	332#	6345	6349*	6624*	6656															
\$ESCAP	001202	364#	1581*	6384*	6387*	6463*	6647	6649	6656												
\$ETABL	001234	383#																			
\$ETEND	001200	317	418#																		
\$FATAL	001216	376#	6585*																		
\$FFLG	044662	6548*	6551*	6579	6588*	6596#															
\$FILLC	001156	353#	6782	6813																	
\$FILLS	001155	352#	6813																		
\$GDADR	001120	336#																			
\$GDADR	001124	338#																			
\$GET42	043472	6350#																			
\$GTSWR	046352	6979#	7290																		
\$HD	= 000000	11																			
\$HIBTS	001000	312#																			
\$HIOCT	047342	7165*	7176#																		
\$ICNT	001104	329#	6454*	6455	6457*	6468															
\$INTAG	001135	343#	7007	7124																	
\$ITEMB	001114	333#	6627*	6635	6656	6668															
\$LF	001212	368#	6656	6813	7108	7118	7177														
\$LFLG	044661	6589*	6595#																		
\$LPADR	001106	330#	1583*	6436*	6445*	6461*	6466	6468	7247												
\$LPERR	001110	331#	1584*	1767*	1874*	1938*	2008*	2071*	2134*	2294*	2362*	2425*	2496*	3566*							
		3642*	3704*	3781*	3836*	3941*	5981*	6278*	6393	6445	6462*	6468	6543	6646							
\$MADR1	001246	401#																			
\$MADR2	001252	405#																			
\$MADR3	001256	408#																			
\$MADR4	001262	411#																			
\$MAIL	001214	313	317	374#	1601	1617	6460	6633	6757												
\$MAMS1	001244	395#																			
\$MAMS2	001250	403#																			
\$MAMS3	001254	406#																			
\$MAMS4	001260	409#																			
\$MBADR	001002	313#																			
\$MFLG	044660	6549*	6555	6590*	6594#																
\$MNEW	047173	6982	7122#																		
\$MSGAD	001230	381#	6565*	6568																	
\$MSGLG	001232	382#	6570*																		
\$MSGTY	001214	375#	6563	6571*	6583	6587*															
\$MSWR	047162	6979	7120#																		
\$MTYP1	001245	396#																			
\$MTYP2	001251	404#																			
\$MTYP3	001255	407#																			
\$MTYP4	001261	410#																			
\$MXCNT	044164	6458	6468#																		
\$NULL	001154	351#	6784	6813																	
\$NWTST=	000001	1686#	1688	1751#	1753	1809#	1811	1858#	1860	1923#	1925	1992#	1994	2055#							
		2057	2118#	2120	2181#	2183	2230#	2232	2279#	2281	2345#	2347	2410#	2412							
		2479#	2481	2548#	2550	2601#	2603	2661#	2663	2721#	2723	2781#	2783	2841#							
		2843	2901#	2903	2968#	2970	3035#	3037	3102#	3104	3169#	3171	3236#	3238							
		3296#	3298	3366#	3368	3421#	3423	3477#	3479	3550#	3552	3623#	3625	3688#							
		3690	3765#	3767	3820#	3822	3927#	3929	3979#	3981	4071#	4073	4115#	4117							
		4202#	4204	4288#	4290	4365#	4367	4442#	4444	4517#	4519	4594#	4596	4672#							
		4674	4750#	4752	4827#	4829	4904#	4906	4982#	4984	5060#	5062	5138#	5140							
		5214#	5216	5292#	5294	5376#	5378	5460#	5462	5545#	5547	5629#	5631	5713#							
		5715	5797#	5799	5881#	5883	5965#	5967	6028#	6030	6104#	6106	6186#	6188							



STMP3	001164	357#													
STMP3	001166	358#													
STMP4	001170	359#													
STMP5	001172	360#													
STMP6	001174	361#													
STMP7	001176	362#													
STN	= 000103	1#													
		1854	11	1686	1697#	1710	1726	1748	1751	1763#	1809	1819#	1841	1847	
		2176	1858	1870#	1923	1934#	1992	2003#	2050	2055#	2066#	2113	2118	2129#	
		2248	2181	2191#	2212	2219#	2226	2230#	2240#	2262	2268#	2275	2279	2290#	
		2614	2357#	2405	2410	2421#	2479	2491#	2548	2561#	2585	2592	2598	2601#	
		2758	2638#	2645	2652	2658#	2661	2674#	2698	2705#	2712	2718	2721	2734#	
		2885	2765#	2772	2778	2781#	2794	2818	2825	2832#	2838	2841	2854#	2878	
		3012	2892#	2898	2901	2912#	2939	2945	2952	2959#	2965	2968	2979#	3006	
		3140	3019#	3026	3032	3035#	3046	3073	3079	3086#	3093	3099	3102	3113#	
		3247	3146#	3153	3160	3166#	3169	3180#	3207	3213	3220#	3227	3233	3236#	
		3378	3274#	3280	3287	3293#	3296	3307#	3334	3340	3347#	3354	3360	3366#	
		3563	3400#	3405	3421	3433#	3456	3461	3477	3488#	3536	3541	3547	3550#	
		3979	3623#	3635#	3680	3688#	3701#	3765	3778#	3820	3833#	3924	3927	3938#	
		4285	3995#	4071	4082#	4115	4126#	4143	4159	4164#	4169	4197	4202	4215#	
		4594	4289#	4300#	4362	4365#	4377#	4439	4442	4454#	4514	4517	4529#	4591	
		4917	4607#	4669	4672	4685#	4747	4750	4762#	4824	4827	4839#	4901	4904	
		5289	4979#	4982	4995#	5057	5060	5073#	5135	5138#	5151#	5211	5214	5227#	
		5629	5292#	5305#	5373	5376#	5389#	5457	5460	5474#	5542	5545	5558#	5626	
		5978	5642#	5710	5713	5726#	5794	5797	5810#	5878	5891	5894#	5962	5965	
		6470	6028#	6040#	6101	6104#	6118#	6183	6186	6203#	6262	6267	6276#	6430	
STPB	001152	350#	6802*	6813											
STPFLG	001157	354#	6751	6813											
STPS	001150	349#	6800	6813											
STRAP	047572	1575	7261#												
STRAP2	047614	7272#	7283												
STRP	= 000016	7276#	7285#	7286#	7287#	7288#	7289#	7290	7291#	7292	7293#	7294#	7295#	7296#	
		7297#	7298#	7299#											
STRPAD	047626	7266	7283#												
STSTM	001004	314#													
STSTNM	001102	327#	6324*	6400	6432*	6459*	6460	6465	6469	6620	6656				
STTYIN	047140	7068	7069	7081	7099	7113	7117#								
STYPBN	= ***** U	7289													
STYPDS	046056	6902#	7288												
STYPE	045346	6576	6751#	7276	7284										
STYPEC	045560	6781	6788	6795	6800#	6801	7011								
STYPEX	045626	6806	6808	6811#											
STYPOC	045654	6843#	7285												
STYPON	045670	6842	6845#	7287											
STYPOS	045630	6838#	7286												
SUNIT	001226	380#													
SUNITM	001010	316#													
SUSMR	001240	387#													
SVECT1	001264	412#	1641	1650*	1651*	1654	1674*	1675*	1676	1678					
SVECT2	001266	413#													
SXTSTR	043666	6414#													
SSGET4	= 000000	6352#													
SSSW08	= 000103	6469#	6470	6471#	6472#	6473#	6474#	6475#	6476#	6477#	6478#	6479#	6480#	6481#	
		6482#	6483#	6484#	6485#	6486#	6487#	6488#	6489#	6490#	6491#	6492#	6493#	6494#	
		6495#	6496#	6497#	6498#	6499#	6500#	6501#	6502#	6503#	6504#	6505#	6506#	6507#	
		6508#	6509#	6510#	6511#	6512#	6513#	6514#	6515#	6516#	6517#	6518#	6519#	6520#	



DZR6B8.P11 CROSS REFERENCE TABLE -- MACRO NAMES		4002	4037													
CLAPSW	318#	4002	4037													
COMMEN	1#	133#														
CYLWRT	318#	2562	2615	2675	2735	2795	2855									
ENDCOM	1#	133#														
ERROR	27#	1709	1725	1730	1734	1737	1740	1743	1746	1749	1787	1795	1800	1803	1839	
	1846	1852	1856	1901	1909	1914	1917	1964	1970	1978	1983	1996	2027	2037	2043	
	2047	2090	2100	2106	2110	2153	2163	2169	2173	2211	2218	2224	2228	2260	2267	
	2273	2277	2318	2324	2331	2336	2339	2381	2387	2394	2399	2402	2449	2455	2462	
	2467	2470	2526	2534	2539	2542	2583	2591	2596	2599	2636	2644	2651	2656	2659	
	2696	2704	2711	2716	2719	2756	2764	2771	2776	2779	2816	2824	2831	2836	2839	
	2876	2884	2891	2896	2899	2937	2943	2951	2958	2963	2966	3004	3010	3018	3025	
	3030	3033	3071	3077	3085	3092	3097	3100	3138	3144	3152	3159	3164	3167	3205	
	3211	3219	3226	3231	3234	3272	3278	3286	3291	3294	3332	3338	3346	3353	3358	
	3361	3399	3404	3455	3460	3505	3510	3515	3518	3535	3540	3545	3548	3602	3609	
	3614	3617	3662	3669	3674	3677	3741	3748	3753	3756	3805	3808	3811	3814	3853	
	3880	3885	3890	3911	3914	3917	3920	3964	3967	3970	3973	4017	4029	4032	4035	
	4055	4059	4063	4103	4112	4142	4158	4163	4168	4189	4192	4195	4198	4235	4248	
	4251	4254	4257	4260	4277	4280	4283	4286	4328	4331	4334	4337	4354	4357	4360	
	4363	4405	4408	4411	4414	4431	4434	4437	4440	4480	4483	4486	4489	4506	4509	
	4512	4515	4557	4560	4563	4566	4583	4586	4589	4592	4635	4638	4641	4644	4661	
	4664	4667	4670	4713	4716	4719	4722	4739	4742	4745	4748	4790	4793	4796	4799	
	4816	4819	4822	4825	4867	4870	4873	4876	4893	4896	4899	4902	4945	4948	4951	
	4954	4971	4974	4977	4980	5023	5026	5029	5032	5049	5052	5055	5058	5101	5104	
	5107	5110	5127	5130	5133	5136	5177	5180	5183	5186	5203	5206	5209	5212	5255	
	5258	5261	5264	5281	5284	5287	5290	5339	5342	5345	5348	5365	5368	5371	5374	
	5423	5426	5429	5432	5449	5452	5455	5458	5508	5511	5514	5517	5534	5537	5540	
	5543	5592	5595	5598	5601	5618	5621	5624	5627	5676	5679	5682	5685	5702	5705	
	5708	5711	5760	5763	5766	5769	5786	5789	5792	5795	5844	5847	5850	5853	5870	
	5873	5876	5879	5928	5931	5934	5937	5954	5957	5960	5963	6013	6016	6019	6022	
	6068	6071	6074	6077	6093	6096	6099	6102	6150	6153	6156	6159	6175	6178	6181	
	6184	6229	6232	6235	6238	6254	6257	6260	6263	6293	6296	6304	6307	6386		
ESCAPE	1#	133#														
FORERR	318#	4301	4378	4455	4530	4608	4686	4763	4840	4918	4996	5074	5152	5228		
GETPRI	1#	133#														
GETSWR	1#	133#	1614#													
IDAETT	318#	5306	5390	5475	5559	5643	5727	5811	5895							
LDLPER	318#	1767	1874	1938	2008	2071	2134	2294	2362	2425	2496	3566	3642	3704	3781	
MESCMD	3836	3941	5981	6278												
MESSFT	318#	1820	2192	2241												
MSCMD1	318#	3379	3434													
MSG	318#	2004	2067	2130												
	1686#	1688	1751#	1753	1809#	1811	1858#	1860	1923#	1925	1992#	1994	2055#	2057	2118#	
	2120	2181#	2183	2230#	2232	2279#	2281	2345#	2347	2410#	2412	2479#	2481	2548#	2550	
	2601#	2603	2661#	2663	2721#	2723	2781#	2783	2841#	2843	2901#	2903	2968#	2970	3035#	
	3037	3102#	3104	3169#	3171	3236#	3238	3296#	3298	3366#	3368	3421#	3423	3477#	3479	
	3550#	3552	3623#	3625	3688#	3690	3765#	3767	3820#	3822	3927#	3929	3979#	3981	4071#	
	4073	4115#	4117	4202#	4204	4288#	4290	4365#	4367	4442#	4444	4517#	4519	4594#	4596	
	4672#	4674	4750#	4752	4827#	4829	4904#	4906	4982#	4984	5060#	5062	5138#	5140	5214#	
	5216	5292#	5294	5376#	5378	5460#	5462	5545#	5547	5629#	5631	5713#	5715	5797#	5799	
	5881#	5883	5965#	5967	6028#	6030	6104#	6106	6186#	6188	6267#	6269				
MSGSEL	318#	2913	2980	3047	3114	3181	3248	3308								
MULT	1#	133#														
NEWTST	1#	133#	1686	1751	1809	1858	1923	1992	2055	2118	2181	2230	2279	2345	2410	
	2479	2548	2601	2661	2721	2781	2841	2901	2968	3035	3102	3169	3236	3296	3366	
	3421	3477	3550	3623	3688	3765	3820	3927	3979	4071	4115	4202	4288	4365	4442	
	4517	4594	4672	4750	4827	4904	4982	5060	5138	5214	5292	5376	5460	5545	5629	

PARGEN	5713	5797	5881	5965	6029	6104	6186	6267							
POP	318#	3564	3702												
PUSH	1#	133#	6591	6592	6943	7166	7215								
REPORT	1#	133#	6552	6554	6575	6902	7140	7195							
SCOPE	1#	133#													
	28#	1696	1762	1818	1869	1933	2002	2065	2128	2190	2239	2299	2356	2420	2490
	2560	2613	2673	2733	2793	2853	2911	2978	3045	3112	3179	3246	3306	3377	3432
	3487	3562	3634	3700	3777	3832	3937	3994	4081	4125	4214	4299	4376	4453	4528
	4606	4684	4761	4838	4916	4994	5072	5150	5226	5304	5388	5473	5557	5641	5725
	5809	5893	5977	6039	6117	6202	6275	6323							
SETPRI	1#	133#													
SETTRA	7276#	7285	7286	7287	7288	7290	7292	7293	7294	7295	7296	7297	7298		
SETUP	1#	133#	1562												
SKIP	1#	133#	1710	1726	1748	1841	1847	1854	2050	2113	2176	2213	2219	2226	2262
	2268	2275	2405	2585	2592	2598	2638	2645	2652	2658	2698	2705	2712	2718	2758
	2765	2772	2778	2818	2825	2832	2838	2878	2885	2892	2898	2939	2945	2952	2959
	2965	3006	3012	3019	3026	3032	3073	3079	3086	3093	3099	3140	3146	3153	3160
	3166	3207	3213	3220	3227	3233	3274	3280	3287	3293	3334	3340	3347	3354	3360
	3400	3405	3456	3461	3536	3541	3547	3680	3924	4143	4159	4164	4169	4197	4285
	4362	4439	4514	4591	4669	4747	4824	4901	4979	5057	5135	5211	5289	5373	5457
	5542	5626	5710	5794	5878	5962	6101	6183	6262						
SLASH	1#	133#													
SPACE	133#														
STARS	1#	133#	287	298	300	307	320	369	372	1686	1695	1751	1761	1809	1817
	1858	1868	1923	1932	1992	2001	2055	2064	2118	2127	2181	2189	2230	2238	2279
	2288	2345	2355	2410	2419	2479	2489	2548	2559	2601	2612	2661	2672	2721	2732
	2781	2792	2841	2852	2901	2910	2968	2977	3035	3044	3102	3111	3169	3178	3236
	3245	3296	3305	3366	3376	3421	3431	3477	3486	3550	3561	3623	3633	3688	3699
	3765	3776	3820	3831	3927	3936	3979	3993	4071	4080	4115	4124	4202	4213	4288
	4298	4365	4375	4442	4452	4517	4527	4594	4605	4672	4683	4750	4760	4827	4837
	4904	4915	4982	4993	5060	5071	5138	5149	5214	5225	5292	5303	5376	5387	5460
	5472	5545	5556	5629	5640	5713	5724	5797	5808	5881	5892	5965	5976	6028	6038
	6104	6116	6186	6201	6267	6274	6315	6397	6536	6547	6604	6657	6666	6736	6815
	6892	6959	6962	7030	7059	7126	7179	7225	7233	7255					
SWRSU	1#	133#	1585#												
TRMTRP	7276#														
TYPBIN	1#	133#													
TYPDEC	1#	133#	6338	6345											
TYPNAM	1#	133#	1607												
TYPNUM	1#	133#													
TYPOCS	1#	133#													
TYPOCT	1#	133#	1631	6980											
TYPTXT	1#	133#	6334	6341											
SSCMRE	318#														
SSCHTM	318#	355	356	357	358	359	360	361	362						
SSSCA	1#	133#													
SSNEWT	1#	133#	1686	1751	1809	1858	1923	1992	2055	2118	2181	2230	2279	2345	2410
	2479	2548	2601	2661	2721	2781	2841	2901	2968	3035	3102	3169	3236	3296	3366
	3421	3477	3550	3623	3688	3765	3820	3927	3979	4071	4115	4202	4288	4365	4442
	4517	4594	4672	4750	4827	4904	4982	5060	5138	5214	5292	5376	5460	5545	5629
	5713	5797	5881	5965	6028	6104	6186	6267	6397	6536	6547	6604	6657	6736	6815
SSSET	7276#	7285	7286	7287	7288	7290	7292	7293	7294	7295	7296	7297	7298		
SSSETM	1601#														
SSSKIP	1#	133#	1710	1726	1748	1841	1847	1854	2050	2113	2176	2213	2219	2226	2262
	2268	2275	2405	2585	2592	2598	2638	2645	2652	2658	2698	2705	2712	2718	2758
	2765	2772	2778	2818	2825	2832	2838	2878	2885	2892	2898	2939	2945	2952	2959

2965	3006	3012	3019	3026	3032	3073	3079	3086	3093	3099	3140	3146	3153	3160
3166	3207	3213	3220	3227	3233	3274	3290	3287	3293	3334	3340	3347	3354	3360
3400	3405	3456	3461	3536	3541	3547	3680	3924	4143	4159	4164	4169	4197	4285
4362	4439	4514	4591	4669	4747	4824	4901	4979	5057	5135	5211	5289	5373	5457
5542	5626	5710	5794	5878	5962	6101	6183	6262						

.EQUAT	18	
.HEADE	18	
.KT11	18	
.SETUP	18	1549
.SWRHI	18	11
.SWRLO	18	23
.SACT1	18	285
.SAPT8	18	370
.SAPTH	18	296
.SAPTY	18	6545
.SASTA	18	
.SCATC	18	271
.SCMTA	18	318
.SDB20	18	
.SDB20	18	
.SDIV	18	
.SEOP	18	6313
.SERRO	18	6602
.SERRT	18	
.SMULT	18	
.SPOWE	18	
.SRAND	18	
.SRDE	18	
.SRDOC	18	7124
.SREAD	18	6957
.SR2AZ	18	
.SSAVE	18	7177
.SSB20	18	
.SSB20	18	
.SSCOP	18	6395
.SSIZE	18	
.SSUPR	18	
.STRAP	18	7253
.STYPB	18	
.STYPD	18	6890
.STYPE	18	6734
.STYPO	18	6813
.S4OCA	18	
.1170	18	



BLO	1666	3924	7082												
BLOS	1649	7070													
BLT	6431	6787	6876	6919	6935	7013	7054	7153							
BMI	3950	4010	4090	4230	4315	4392	4467	4544	4622	4700	4777	4854	4932	5010	5088
BNE	5164	5242	5325	5409	5494	5578	5662	5746	5830	5914	5999	6055	6131	6216	6296
	1568	1592	1610	1616	1620	1629	1704	1716	1779	1830	1844	1698	1950	2019	2050
	2082	2113	2145	2176	2202	2216	2251	2265	2306	2323	2374	2386	2405	2437	2454
	2473	2508	2522	2575	2628	2643	2688	2703	2748	2763	2808	2823	2868	2883	2924
	2950	2991	3017	3058	3084	3125	3151	3192	3218	3259	3319	3345	3394	3450	3497
	3504	3509	3528	3579	3589	3594	3655	3661	3718	3728	3793	3848	3863	3866	3874
	3894	3899	3952	4012	4092	4137	4152	4172	4178	4226	4232	4247	4311	4317	4388
	4394	4463	4469	4540	4546	4618	4624	4696	4702	4773	4779	4850	4856	4928	4934
	5006	5012	5084	5090	5160	5166	5238	5244	5321	5327	5405	5411	5490	5496	5574
	5580	5658	5664	5742	5748	5826	5832	5910	5916	5995	6001	6051	6057	6127	6133
	6136	6212	6218	6374	6389	6412	6451	6558	6564	6567	6584	6629	6634	6652	6696
	6758	6765	6767	6775	6783	6797	6804	6866	6924	6968	6974	6994	7001	7008	7045
	7051	7074	7076	7092	7096	7106	7238	7240							
BPL	6641	6752	6801	6864	6910	6940	6970	6986	7041	7047					
BR	1550	1553	1594	1622	1625	1710	1726	1789	1796	1841	1847	1903	1910	1966	1972
	1979	2029	2038	2053	2092	2101	2116	2155	2164	2179	2213	2219	2262	2268	2320
	2325	2332	2383	2388	2395	2408	2451	2456	2463	2528	2535	2585	2592	2638	2645
	2652	2698	2705	2712	2758	2765	2772	2818	2825	2832	2878	2885	2892	2939	2945
	2952	2959	3006	3012	3019	3026	3073	3079	3086	3093	3140	3146	3153	3160	3207
	3213	3220	3227	3274	3280	3287	3334	3340	3347	3354	3400	3405	3410	3456	3461
	3466	3506	3511	3536	3541	3603	3610	3663	3670	3686	3742	3749	3854	3881	3886
	3891	4018	4056	4060	4143	4159	4164	4169	4236	6335	6342	6375	6414	6420	6423
	6437	6446	6449	6550	6572	6639	6702	6717	6754	6780	6790	6799	6806	6842	6857
	6878	6921	6938	6997	7024	7026	7052	7085	7094	7100	7102	7162	7175	7231	
CLC	3413	3469													
CLR	1555	1566	1580	1581	1601	1606	1653	1721	1722	1765	1784	1835	1872	1880	1892
	1898	1936	1960	2023	2086	2149	2207	2256	2292	2360	2423	2493	2495	2512	2580
	2633	2693	2753	2813	2873	2934	3001	3068	3135	3202	3268	3329	3381	3436	3531
	3532	3584	3592	3638	3681	3685	3723	3731	3801	3802	3835	3858	3907	3908	3960
	3961	4002	4026	4037	4069	4085	4099	4100	4185	4186	4227	4245	4273	4274	4312
	4350	4351	4389	4427	4428	4464	4502	4503	4541	4579	4580	4619	4657	4658	4697
	4735	4736	4774	4812	4813	4851	4889	4890	4929	4967	4968	5007	5045	5046	5085
	5123	5124	5161	5199	5200	5239	5277	5278	5312	5322	5361	5362	5396	5406	5445
	5446	5481	5491	5530	5531	5565	5575	5614	5615	5649	5659	5698	5699	5733	5743
	5782	5783	5817	5827	5866	5867	5901	5911	5950	5951	5996	6009	6010	6052	6064
	6065	6089	6090	6128	6147	6171	6172	6213	6225	6226	6250	6251	6301	6324	6325
	6349	6379	6387	6427	6448	6463	6687	6855	6913	6916	6983	6984	7067	7090	7146
	7147	7235													
CLRB	6447	6588	6589	6590	6779	6805	6942	7097	7107	7171					
CMP	1567	1591	1611	1619	1628	1637	1648	1665	1667	1707	1723	1785	1793	1798	1801
	1806	1836	1849	1853	1899	1907	1912	1915	1920	1962	1968	1976	1981	1984	1989
	2024	2033	2040	2044	2087	2096	2103	2107	2150	2159	2166	2170	2208	2221	2225
	2257	2270	2274	2316	2329	2334	2337	2342	2379	2392	2397	2400	2404	2447	2460
	2465	2468	2476	2524	2532	2537	2540	2545	2581	2594	2597	2634	2654	2657	2694
	2714	2717	2754	2774	2777	2814	2834	2837	2874	2894	2897	2935	2941	2961	2964
	3002	3008	3028	3031	3069	3075	3095	3098	3136	3142	3162	3165	3203	3209	3229
	3232	3270	3276	3289	3292	3330	3336	3356	3359	3397	3402	3418	3453	3458	3474
	3513	3516	3543	3546	3600	3607	3612	3615	3620	3660	3667	3672	3675	3739	3746
	3751	3754	3759	3803	3806	3809	3812	3817	3851	3878	3883	3888	3909	3912	3915
	3918	3923	3962	3965	3968	3971	3976	4027	4030	4033	4101	4110	4140	4156	4161
	4166	4187	4190	4193	4196	4249	4252	4255	4258	4275	4278	4281	4284	4326	4329
	4332	4335	4352	4355	4358	4361	4403	4406	4409	4412	4429	4432	4435	4438	4478

	4481	4484	4487	4504	4507	4510	4513	4555	4558	4561	4564	4581	4584	4587	4590
	4633	4636	4639	4642	4659	4662	4665	4668	4711	4714	4717	4720	4737	4740	4743
	4746	4788	4791	4794	4797	4814	4817	4820	4823	4865	4868	4871	4874	4891	4894
	4897	4900	4943	4946	4949	4952	4969	4972	4975	4978	5021	5024	5027	5030	5047
	5050	5053	5056	5099	5102	5105	5108	5125	5128	5131	5134	5175	5178	5181	5184
	5201	5204	5207	5210	5253	5256	5259	5262	5279	5282	5285	5298	5337	5340	5343
	5346	5363	5366	5369	5372	5421	5424	5427	5430	5447	5450	5453	5456	5506	5509
	5512	5515	5632	5535	5538	5541	5590	5593	5596	5599	5616	5619	5622	5625	5674
	5677	5680	5683	5700	5703	5706	5709	5758	5761	5764	5767	5784	5787	5790	5793
	5842	5845	5848	5851	5868	5871	5874	5877	5926	5929	5932	5935	5952	5955	5958
	5961	6011	6014	6017	6020	6066	6069	6072	6075	6091	6094	6097	6100	6148	6151
	6154	6157	6173	6176	6179	6182	6227	6230	6233	6236	6252	6255	6258	6261	6291
	6294	6302	6305	6310	6377	6421	6430	6455	6651	6722	6934	6967	6973	6993	7000
CMPB	7012	7014	7044	7050	7053	7055	7069	7081							
	1617	6441	6557	6633	6757	6772	6774	6782	6803	6807	6975	7007	7073	7091	7095
	7105	7150	7152												
COM DEC	2514	6706													
	1703	1715	1778	1829	1887	1949	2018	2081	2144	2201	2250	2305	2373	2436	2507
	2574	2627	2687	2747	2807	2867	2923	2990	3057	3124	3191	3258	3318	3393	3449
	3496	3526	3578	3588	3654	3717	3727	3792	3847	3862	3873	3893	3898	3951	4011
	4091	4136	4151	4171	4177	4225	4231	4310	4316	4387	4393	4462	4468	4539	4545
	4617	4623	4695	4701	4772	4778	4849	4855	4927	4933	5005	5011	5083	5089	5159
	5165	5237	5243	5320	5326	5404	5410	5489	5495	5573	5579	5657	5663	5741	5747
	5825	5831	5909	5915	5994	6000	6050	6056	6126	6132	6135	6211	6217	6328	6373
	6433	6670	6699	6707	7080	7239									
DECB EMT HALT INC	6786	6789	6863	6874											
	27														
	277	6642	6653	6732	6753	7230									
INC	1609	1805	1919	1988	2341	2475	2544	3417	3473	3587	3619	3726	3758	3816	3861
	3922	3975	6326	6454	6587	6624	6719	6869	6877	6920	7022	7237			
INCB IOT JMP JSR MOV	6459	6618	6809												
	28														
	281	282	284	3925	6358	6393	6730	7247							
	1680	6353	6576	6630	6636	6762	6781	6788	6795	7011	7245				
	1549	1552	1557	1558	1559	1565	1569	1571	1572	1573	1574	1575	1576	1577	1578
	1579	1583	1584	1587	1588	1589	1590	1595	1597	1598	1599	1604	1631	1635	1639
	1641	1646	1663	1676	1681	1682	1697	1698	1699	1700	1701	1702	1705	1706	1712
	1713	1714	1717	1718	1719	1720	1763	1764	1766	1767	1771	1772	1773	1774	1775
	1776	1777	1780	1781	1782	1783	1788	1791	1819	1820	1821	1822	1823	1824	1825
	1826	1827	1828	1831	1832	1833	1834	1838	1840	1845	1851	1855	1870	1871	1873
	1874	1878	1879	1882	1883	1884	1885	1886	1889	1890	1891	1902	1905	1934	1935
	1937	1938	1942	1943	1945	1946	1951	1952	1953	1954	1955	1961	1965	1971	1974
	2003	2004	2005	2006	2007	2008	2012	2013	2014	2015	2016	2017	2020	2021	2022
	2026	2028	2031	2036	2042	2046	2066	2067	2068	2069	2070	2071	2075	2076	2077
	2078	2079	2080	2083	2084	2085	2089	2091	2094	2099	2105	2109	2129	2130	2131
	2132	2133	2134	2138	2139	2140	2141	2142	2143	2146	2147	2148	2152	2154	2157
	2162	2168	2172	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2203	2204
	2205	2206	2210	2212	2217	2223	2227	2240	2241	2242	2243	2244	2245	2246	2247
	2248	2249	2252	2253	2254	2255	2259	2261	2266	2272	2276	2290	2291	2293	2294
	2298	2299	2300	2301	2302	2303	2304	2307	2308	2309	2310	2311	2319	2327	2357
	2358	2359	2361	2362	2366	2367	2368	2369	2370	2371	2372	2375	2376	2377	2378
	2382	2390	2406	2407	2421	2422	2424	2425	2429	2430	2431	2432	2433	2434	2435
	2438	2439	2440	2441	2442	2450	2458	2474	2491	2492	2494	2496	2500	2501	2502
	2503	2504	2505	2506	2509	2510	2511	2513	2527	2530	2561	2562	2563	2564	2565
	2566	2567	2568	2569	2570	2571	2572	2573	2576	2577	2578	2579	2584	2588	2614
	2615	2616	2617	2618	2619	2620	2621	2622	2623	2624	2625	2626	2629	2630	2631

2632	2637	2648	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685
2686	2689	2690	2691	2692	2697	2708	2734	2735	2736	2737	2738	2739	2740	2741
2742	2743	2744	2745	2746	2749	2750	2751	2752	2757	2768	2794	2795	2796	2797
2798	2799	2800	2801	2802	2803	2804	2805	2806	2809	2810	2811	2812	2817	2828
2854	2855	2856	2857	2858	2859	2860	2861	2862	2863	2864	2865	2866	2869	2870
2871	2872	2877	2888	2912	2913	2914	2915	2916	2917	2919	2920	2925	2926	2927
2928	2929	2933	2938	2944	2955	2979	2980	2981	2982	2983	2984	2986	2987	2992
2993	2994	2995	2996	3000	3005	3011	3022	3046	3047	3048	3049	3050	3051	3053
3054	3059	3060	3061	3062	3063	3067	3072	3078	3089	3113	3114	3115	3116	3117
3118	3120	3121	3126	3127	3128	3129	3130	3134	3139	3145	3156	3180	3181	3182
3183	3184	3185	3187	3188	3193	3194	3195	3196	3197	3201	3206	3212	3223	3247
3248	3249	3250	3251	3252	3254	3255	3260	3261	3262	3263	3264	3269	3273	3279
3283	3307	3308	3309	3310	3311	3312	3314	3315	3320	3321	3322	3323	3324	3328
3333	3339	3350	3378	3379	3380	3382	3383	3384	3385	3386	3388	3389	3390	3395
3396	3416	3433	3434	3435	3437	3438	3439	3440	3441	3442	3443	3444	3446	3451
3452	3472	3488	3489	3490	3491	3492	3493	3494	3495	3498	3499	3500	3501	3519
3520	3522	3523	3524	3525	3529	3530	3563	3564	3565	3566	3570	3571	3573	3574
3575	3580	3581	3582	3583	3590	3591	3595	3598	3605	3635	3636	3637	3639	3640
3641	3642	3646	3647	3649	3650	3651	3656	3657	3658	3665	3682	3683	3684	3701
3702	3703	3704	3708	3709	3712	3713	3714	3719	3720	3721	3722	3729	3730	3734
3737	3744	3778	3779	3780	3781	3785	3786	3787	3788	3789	3790	3791	3794	3795
3796	3797	3798	3799	3833	3834	3836	3840	3841	3842	3843	3844	3845	3846	3849
3850	3856	3857	3864	3868	3869	3870	3871	3872	3875	3876	3877	3895	3896	3897
3900	3901	3902	3903	3904	3905	3938	3939	3940	3941	3945	3946	3947	3948	3953
3954	3955	3956	3957	3958	3995	3996	3997	3998	3999	4000	4001	4003	4007	4008
4013	4014	4021	4022	4023	4024	4025	4036	4038	4043	4044	4045	4047	4048	4050
4051	4065	4066	4067	4082	4083	4084	4086	4087	4088	4093	4094	4095	4096	4097
4098	4105	4106	4107	4108	4109	4113	4126	4127	4128	4129	4130	4131	4132	4133
4134	4135	4138	4139	4145	4146	4147	4148	4149	4150	4153	4154	4155	4174	4175
4176	4179	4180	4181	4182	4183	4184	4215	4216	4217	4218	4219	4220	4221	4222
4223	4224	4228	4233	4234	4238	4239	4240	4241	4242	4243	4244	4261	4262	4263
4264	4266	4267	4268	4269	4270	4271	4272	4300	4301	4302	4303	4304	4305	4306
4307	4308	4309	4313	4318	4319	4320	4321	4322	4323	4324	4325	4338	4339	4340
4341	4343	4344	4345	4346	4347	4348	4349	4377	4378	4379	4380	4381	4382	4383
4384	4385	4386	4390	4395	4396	4397	4398	4399	4400	4401	4402	4415	4416	4417
4418	4420	4421	4422	4423	4424	4425	4426	4454	4455	4456	4457	4458	4459	4460
4461	4465	4470	4471	4472	4473	4474	4475	4476	4477	4490	4491	4492	4493	4495
4496	4497	4498	4499	4500	4501	4529	4530	4531	4532	4533	4534	4535	4536	4537
4538	4542	4547	4548	4549	4550	4551	4552	4553	4554	4567	4568	4569	4570	4572
4573	4574	4575	4576	4577	4578	4607	4608	4609	4610	4611	4612	4613	4614	4615
4616	4620	4625	4626	4627	4628	4629	4630	4631	4632	4645	4646	4647	4648	4650
4651	4652	4653	4654	4655	4656	4685	4686	4687	4688	4689	4690	4691	4692	4693
4694	4698	4703	4704	4705	4706	4707	4708	4709	4710	4723	4724	4725	4726	4728
4729	4730	4731	4732	4733	4734	4762	4763	4764	4765	4766	4767	4768	4769	4770
4771	4775	4780	4781	4782	4783	4784	4785	4786	4787	4800	4801	4802	4803	4805
4806	4807	4808	4809	4810	4811	4839	4840	4841	4842	4843	4844	4845	4846	4847
4848	4852	4857	4858	4859	4860	4861	4862	4863	4864	4877	4878	4879	4880	4882
4883	4884	4885	4886	4887	4888	4917	4918	4919	4920	4921	4922	4923	4924	4925
4926	4930	4935	4936	4937	4938	4939	4940	4941	4942	4955	4956	4957	4958	4960
4961	4962	4963	4964	4965	4966	4995	4996	4997	4998	4999	5000	5001	5002	5003
5004	5008	5013	5014	5015	5016	5017	5018	5019	5020	5033	5034	5035	5036	5038
5039	5040	5041	5042	5043	5044	5073	5074	5075	5076	5077	5078	5079	5080	5081
5082	5086	5091	5092	5093	5094	5095	5096	5097	5098	5111	5112	5113	5114	5116
5117	5118	5119	5120	5121	5122	5151	5152	5153	5154	5155	5156	5157	5158	5162
5167	5168	5169	5170	5171	5172	5173	5174	5187	5188	5189	5190	5192	5193	5194
5195	5196	5197	5198	5227	5228	5229	5230	5231	5232	5233	5234	5235	5236	5240

	5245	5246	5247	5248	5249	5250	5251	5252	5265	5266	5267	5268	5270	5271	5272
	5273	5274	5275	5276	5305	5306	5307	5308	5309	5310	5311	5314	5315	5316	5317
	5318	5319	5323	5328	5329	5330	5331	5332	5334	5335	5336	5349	5350	5351	5352
	5354	5355	5356	5357	5358	5359	5360	5389	5390	5391	5392	5393	5394	5395	5398
	5399	5400	5401	5402	5403	5407	5412	5413	5414	5415	5416	5418	5419	5420	5433
	5434	5435	5436	5438	5439	5440	5441	5442	5443	5444	5474	5475	5476	5477	5478
	5479	5480	5483	5484	5485	5486	5487	5488	5492	5497	5498	5499	5500	5501	5503
	5504	5505	5518	5519	5520	5521	5523	5524	5525	5526	5527	5528	5529	5558	5559
	5560	5561	5562	5563	5564	5567	5568	5569	5570	5571	5572	5576	5581	5582	5583
	5584	5585	5587	5588	5589	5602	5603	5604	5605	5607	5608	5609	5610	5611	5612
	5613	5642	5643	5644	5645	5646	5647	5648	5651	5652	5653	5654	5655	5656	5660
	5665	5666	5667	5668	5669	5671	5672	5673	5686	5687	5688	5689	5691	5692	5693
	5694	5695	5696	5697	5726	5727	5728	5729	5730	5731	5732	5735	5736	5737	5738
	5739	5740	5744	5749	5750	5751	5752	5753	5755	5756	5757	5770	5771	5772	5773
	5775	5776	5777	5778	5779	5780	5781	5810	5811	5812	5813	5814	5815	5816	5819
	5820	5821	5822	5823	5824	5828	5833	5834	5835	5836	5837	5839	5840	5841	5854
	5855	5856	5857	5859	5860	5861	5862	5863	5864	5865	5894	5895	5896	5897	5898
	5899	5900	5903	5904	5905	5906	5907	5908	5912	5917	5918	5919	5920	5921	5923
	5924	5925	5938	5939	5940	5941	5943	5944	5945	5946	5947	5948	5949	5978	5979
	5980	5981	5985	5986	5987	5988	5989	5990	5991	5992	5993	5997	6002	6003	6004
	6005	6006	6007	6040	6041	6042	6043	6044	6045	6046	6047	6048	6049	6053	6058
	6059	6060	6061	6062	6063	6078	6079	6080	6081	6082	6083	6084	6085	6086	6087
	6088	6118	6119	6120	6121	6122	6123	6124	6125	6129	6134	6137	6138	6139	6140
	6141	6145	6146	6160	6161	6162	6163	6164	6165	6166	6167	6168	6169	6170	6203
	6204	6205	6206	6207	6208	6209	6210	6214	6219	6220	6221	6222	6223	6224	6239
	6240	6241	6242	6243	6244	6245	6246	6247	6248	6249	6276	6277	6278	6282	6283
	6286	6287	6288	6289	6290	6297	6298	6299	6300	6331	6338	6345	6350	6365	6366
	6367	6369	6370	6371	6372	6378	6384	6385	6392	6416	6417	6419	6422	6432	6436
	6445	6457	6458	6461	6462	6465	6466	6543	6553	6554	6561	6565	6570	6571	6573
	6575	6585	6591	6592	6620	6625	6646	6649	6675	6680	6685	6688	6689	6697	6709
	6728	6729	6755	6756	6761	6769	6784	6838	6846	6847	6848	6854	6861	6879	6880
	6881	6882	6883	6903	6904	6905	6906	6907	6908	6909	6914	6917	6937	6943	6944
	6945	6946	6947	6949	6950	6980	7004	7009	7038	7039	7066	7068	7079	7110	7111
	7112	7113	7138	7139	7140	7141	7142	7144	7145	7164	7165	7166	7167	7168	7195
	7196	7197	7198	7199	7200	7201	7202	7203	7204	7211	7212	7213	7214	7215	7216
	7217	7218	7219	7220	7227	7228	7229	7236	7241	7242	7243	7246	7261	7262	7266
	7272	7273													
MOV B	1582	1623	1654	1678	1881	1893	5313	5397	5482	5566	5650	5734	5818	5902	6428
	6460	6464	6548	6549	6551	6627	6635	6668	6691	6766	6794	6802	6839	6840	6843
	6844	6845	6849	6852	6853	6872	6912	6915	6929	6932	6941	6971	6988	7042	7048
	7072	7077	7083	7088	7103	7148	7264								
NEG	6850	6911													
NOP	4042	4046	4049	6354	6355	6356									
RESET	1556	6352													
ROL	6856	6858	6859	6860	6862	7155	7157	7159							
ROR	3414	3415	3470	3471	3585	3724	3859								
RTI	1560	1596	1683	4004	4015	4039	4052	6390	6467	6544	6655	6771	6884	6951	7010
	7058	7114	7169	7205	7221	7274									
RTS	6380	6593	6733	6811	7267										
SEC	3409	3465													
SUB	6568	6626	6918												
TRAP	7276	7285	7286	7287	7288	7290	7292	7293	7294	7295	7296	7297	7298		
TST	1615	2472	3679	4104	6418	6438	6452	6563	6581	6583	6640	6647	6693	6695	6704
	6711	6726	6768	6776	6798	6867	6923	6933	7002	7017	7075	7086	7109	7163	7170
	7263														
TSTB	3949	4009	4089	4229	4314	4391	4466	4543	4621	4699	4776	4853	4931	5009	5087

	5163	5241	5324	5408	5493	5577	5661	5745	5829	5913	5998	6054	6130	6215	6439
	6541	6555	6566	6579	6692	6751	6800	6925	6933	6969	6985	7040	7046		
.ASCII	366	367	7599	7866	7883	7923	7937	7981	8008	8016	8024	8042	8130	8169	
.ASCIZ	365	368	1627	6337	6344	7118	7119	7120	7122	7250	7567	7571	7573	7578	7582
	7583	7594	7597	7602	7604	7607	7610	7616	7622	7626	7630	7634	7638	7645	7652
	7656	7660	7664	7668	7674	7680	7684	7688	7699	7710	7716	7721	7724	7727	7732
	7737	7741	7745	7751	7760	7767	7777	7784	7790	7798	7805	7810	7822	7833	7844
	7852	7860	7876	7893	7897	7903	7913	7933	7946	7950	7960	7970	7989	7992	8001
	8031	8035	8050	8053	8062	8069	8078	8087	8093	8099	8106	8113	8121	8138	8143
	8152	8161	8177	8181	8191	8200	8209	8216	8223	8231	8235	8239	8245	8252	8259
	8263	8270	8277	8284	8290	8296	8302	8306	8315	8325	8336	8347	8351	8360	8369
	8376	8382	8389	8396	8402	8407	8415	8420							
.BLKB	7117														
.BLKW	6956														
.BYTE	327	328	333	334	342	343	351	352	353	354	384	385	395	396	403
	404	406	407	409	410	6360	6594	6595	6596	6637	6638	6885	6886	6887	6888
	7115	7116	7358	7360	7362	7364	7366	7368	7370	7372	7374	7376	7378	7380	7382
	7384	7386	7388	7390	7392	7394	7396	7398	7400	7402	7404	7406	7408	7410	7412
	7414	7416	7418	7420	7422	7424	7426	7428	7430	7432	7434	7436	7438	7440	7442
	7444	7446	7448	7450	7452	7454	7456	7458	7460	7462	7464	7466	7468	7470	7472
	7474	7476	7478	7480	7482	7484	7486	7488	7490	7492	7494	7496	7498	7500	7502
	7504	7506	7508	7510	7512	7514	7516	7518	7520	7522	7524	7526	7528	7530	7532
	7534	7536	7538	7540	7542	7544	7546	7548	7550	7552	7554	7556	7558	7560	7562
	7564														
.DSABL	7027														
.ENABL	1	6960													
.END	8428														
.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
	1583	1585	415	416	417	420	1549	1569	1570	1573	1575	1577	1579	1580	1581
	1727	1749	1606	1611	1613	1619	1625	1627	1687	1688	1695	1696	1697	1698	1711
	1848	1855	1752	1753	1761	1762	1763	1764	1810	1811	1817	1818	1819	1820	1842
	1994	2001	1859	1860	1868	1869	1870	1871	1924	1925	1932	1933	1934	1935	1993
	2127	2002	1860	1860	1868	1869	1870	1871	2064	2065	2066	2067	2114	2119	2120
	2232	2003	2002	2003	2004	2051	2056	2057	2064	2065	2066	2067	2114	2119	2231
	2347	2128	2129	2130	2177	2182	2183	2189	2190	2191	2192	2214	2220	2227	2231
	2490	2238	2239	2240	2241	2263	2269	2276	2280	2281	2288	2289	2290	2291	2346
	2612	2355	2356	2357	2358	2406	2411	2412	2419	2420	2421	2422	2480	2481	2489
	2699	2491	2492	2549	2550	2559	2560	2561	2562	2586	2588	2593	2599	2602	2603
	2779	2613	2614	2615	2639	2646	2648	2653	2659	2662	2663	2672	2673	2674	2675
	2853	2706	2708	2713	2719	2722	2723	2732	2733	2734	2735	2759	2766	2768	2773
	2940	2782	2783	2792	2793	2794	2795	2819	2826	2828	2833	2839	2842	2843	2852
	3020	2854	2855	2879	2886	2888	2893	2899	2902	2903	2910	2911	2912	2913	2935
	3094	2946	2953	2955	2960	2966	2969	2970	2977	2978	2979	2980	3002	3007	3013
	3170	3022	3027	3033	3036	3037	3044	3045	3046	3047	3069	3074	3080	3087	3089
	3245	3100	3103	3104	3111	3112	3113	3114	3136	3141	3147	3154	3156	3161	3167
	3308	3171	3178	3179	3180	3181	3203	3208	3214	3221	3223	3228	3234	3237	3238
	3401	3246	3247	3248	3270	3275	3281	3283	3288	3294	3297	3298	3305	3306	3307
	3488	3330	3335	3341	3348	3350	3355	3361	3367	3368	3376	3377	3378	3379	3390
	3633	3406	3422	3423	3431	3432	3433	3434	3446	3457	3462	3478	3479	3486	3487
	3776	3489	3537	3542	3548	3551	3552	3561	3562	3563	3564	3573	3595	3624	3625
	3938	3634	3635	3636	3681	3689	3690	3699	3700	3701	3702	3712	3734	3766	3767
	4117	3777	3778	3779	3821	3822	3831	3832	3833	3834	3925	3928	3929	3936	3937
	4216	3939	3980	3981	3993	3994	3995	3996	4072	4073	4080	4081	4082	4083	4116
	4377	4124	4125	4126	4127	4144	4160	4165	4170	4198	4203	4204	4213	4214	4215
		4286	4289	4290	4298	4299	4300	4301	4306	4323	4363	4366	4367	4375	4376
		4378	4383	4400	4440	4443	4444	4445	4452	4453	4454	4455	4475	4515	4518

4519	4527	4528	4529	4530	4535	4552	4592	4595	4596	4605	4606	4607	4608	4613	
4630	4670	4673	4674	4683	4684	4685	4696	4691	4708	4748	4751	4752	4760	4761	
4762	4763	4768	4785	4825	4828	4829	4837	4838	4839	4840	4845	4862	4902	4905	
4906	4915	4916	4917	4918	4923	4940	4980	4983	4984	4993	4994	4995	4996	5001	
5018	5058	5061	5062	5071	5072	5073	5074	5079	5096	5136	5139	5140	5149	5150	
5151	5152	5155	5172	5212	5215	5216	5225	5226	5227	5228	5233	5250	5290	5293	
5294	5303	5304	5305	5306	5317	5374	5377	5378	5387	5388	5389	5390	5401	5458	
5461	5462	5472	5473	5474	5475	5486	5543	5546	5547	5556	5557	5558	5559	5570	
5627	5630	5631	5640	5641	5642	5643	5654	5711	5714	5715	5724	5725	5726	5727	
5738	5795	5798	5799	5808	5809	5810	5811	5822	5879	5882	5883	5892	5893	5894	
5895	5906	5963	5966	5967	5976	5977	5978	5979	6029	6030	6038	6039	6040	6041	
6102	6105	6106	6116	6117	6118	6119	6184	6187	6188	6201	6202	6203	6204	6263	
6268	6269	6274	6275	6276	6277	6316	6317	6319	6321	6324	6330	6333	6334	6337	
6344	6350	6352	6359	6360	6361	6398	6401	6406	6411	6413	6424	6427	6439	6441	
6443	6450	6454	6459	6461	6465	6468	6469	6536	6537	6548	6549	6552	6579	6594	
6605	6608	6618	6625	6630	6631	6632	6640	6651	6655	6656	6658	6667	6737	6766	
6816	6893	6960	6961	6963	6991	7027	7031	7059	7060	7068	7070	7073	7101	7118	
7124	7127	7133	7177	7180	7226	7234	7256	7262	7265	7284	7285	7286	7287	7288	
7289	7290	7291	7292	7293	7294	7295	7296	7297	7298						
.EQUIV	27	28	36	51	52	81	82	83	84	85	86	87	88	89	90
.EVEN	109	110	111	112	113	114	115	116	117	118					
.IF	373	1627	6337	6344	6361	6597	7252								
	2	18	20	21	22	23	25	91	119	280	287	290	292	298	300
	307	320	324	326	355	363	364	365	369	370	372	395	403	406	409
	412	413	414	415	416	417	418	420	1549	1564	1569	1571	1573	1575	1577
	1579	1580	1581	1583	1601	1610	1611	1612	1614	1617	1626	1686	1688	1695	1697
	1698	1710	1726	1748	1751	1753	1761	1763	1764	1809	1811	1817	1819	1820	1841
	1847	1854	1858	1860	1868	1870	1871	1923	1925	1932	1934	1935	1992	1994	2001
	2003	2004	2050	2055	2057	2064	2066	2067	2113	2118	2120	2127	2129	2130	2176
	2181	2183	2189	2191	2192	2213	2219	2226	2230	2232	2238	2240	2241	2262	2268
	2275	2279	2281	2288	2290	2291	2345	2347	2355	2357	2358	2405	2410	2412	2419
	2421	2422	2479	2481	2489	2491	2492	2548	2550	2559	2561	2562	2585	2588	2592
	2598	2601	2603	2612	2614	2615	2638	2641	2645	2652	2658	2661	2663	2672	2674
	2675	2698	2701	2705	2712	2718	2721	2723	2732	2734	2735	2758	2761	2765	2772
	2778	2781	2783	2792	2794	2795	2818	2821	2825	2832	2838	2841	2843	2852	2854
	2855	2878	2881	2885	2892	2898	2901	2903	2910	2912	2913	2933	2939	2945	2948
	2952	2959	2965	2968	2970	2977	2979	2980	3000	3006	3012	3015	3019	3026	3032
	3035	3037	3044	3046	3047	3067	3073	3079	3082	3086	3093	3099	3102	3104	3111
	3113	3114	3134	3140	3146	3149	3153	3160	3166	3169	3171	3178	3180	3181	3201
	3207	3213	3216	3220	3227	3233	3236	3238	3245	3247	3248	3268	3274	3280	3283
	3287	3293	3296	3298	3305	3307	3308	3328	3334	3340	3343	3347	3354	3360	3366
	3368	3376	3378	3379	3382	3400	3405	3421	3423	3431	3433	3434	3437	3456	3461
	3477	3479	3486	3488	3489	3536	3541	3547	3550	3552	3561	3563	3564	3572	3594
	3623	3625	3633	3635	3636	3680	3688	3690	3699	3701	3702	3710	3733	3765	3767
	3776	3778	3779	3820	3822	3831	3833	3834	3924	3927	3929	3936	3938	3939	3979
	3981	3993	3995	3996	4071	4073	4080	4082	4083	4115	4117	4124	4126	4127	4143
	4159	4164	4169	4197	4202	4204	4213	4215	4216	4285	4288	4290	4298	4300	4301
	4304	4322	4362	4365	4367	4375	4377	4378	4381	4399	4439	4442	4444	4452	4454
	4455	4458	4474	4514	4517	4519	4527	4529	4530	4533	4551	4591	4594	4596	4605
	4607	4608	4611	4629	4669	4672	4674	4683	4685	4686	4689	4707	4747	4750	4752
	4760	4762	4763	4766	4784	4824	4827	4829	4837	4839	4840	4843	4861	4901	4904
	4906	4915	4917	4918	4921	4939	4979	4982	4984	4993	4995	4996	4999	5017	5057
	5060	5062	5071	5073	5074	5077	5095	5135	5138	5140	5149	5151	5152	5155	5171
	5211	5214	5216	5225	5227	5228	5231	5249	5289	5292	5294	5303	5305	5306	5316
	5373	5376	5378	5387	5389	5390	5400	5457	5460	5462	5472	5474	5475	5485	5542
	5545	5547	5556	5558	5559	5569	5626	5629	5631	5640	5642	5643	5653	5710	5713

	5715	5724	5726	5727	5737	5794	5797	5799	5808	5810	5811	5821	5878	5881	5883
	5892	5894	5895	5905	5962	5965	5967	5976	5978	5979	6028	6030	6038	6040	6041
	6101	6104	6106	6116	6118	6119	6183	6186	6188	6201	6203	6204	6262	6267	6269
	6274	6276	6277	6315	6316	6317	6319	6320	6321	6323	6329	6332	6334	6336	6343
	6352	6358	6360	6361	6397	6400	6405	6411	6423	6425	6426	6427	6439	6440	6441
	6450	6452	6460	6462	6467	6468	6469	6536	6547	6549	6552	6579	6594	6604	6607
	6618	6621	6628	6630	6631	6633	6640	6644	6651	6655	6656	6657	6666	6736	6757
	6815	6892	6959	6961	6962	6963	6991	7030	7031	7059	7067	7069	7073	7074	7117
	7118	7124	7126	7129	7145	7179	7225	7233	7255	7261	7265	7276	7285	7286	7287
	7288	7289	7290	7292	7293	7294	7295	7296	7297	7298					
.IFF	18	21	22	23	25	288	292	294	299	301	308	321	324	327	355
	370	373	1569	1610	1612	1687	1688	1696	1697	1711	1727	1749	1752	1753	1762
	1763	1810	1811	1818	1819	1842	1848	1855	1859	1860	1869	1870	1924	1925	1933
	1934	1993	1994	2032	2003	2051	2056	2057	2065	2066	2114	2119	2120	2128	2129
	2177	2182	2183	2190	2191	2214	2220	2227	2231	2232	2239	2240	2263	2269	2276
	2280	2281	2289	2290	2346	2347	2356	2357	2406	2411	2412	2420	2421	2480	2481
	2490	2491	2549	2550	2560	2561	2586	2593	2599	2602	2603	2613	2614	2639	2646
	2653	2659	2662	2663	2673	2674	2699	2706	2713	2719	2722	2723	2733	2734	2759
	2766	2773	2779	2782	2783	2793	2794	2819	2826	2833	2839	2842	2843	2853	2854
	2879	2886	2893	2899	2902	2903	2911	2912	2935	2940	2946	2953	2960	2966	2969
	2970	2978	2979	3002	3007	3013	3020	3027	3033	3036	3037	3045	3046	3069	3074
	3080	3087	3094	3100	3103	3104	3112	3113	3136	3141	3147	3154	3161	3167	3170
	3171	3179	3180	3203	3208	3214	3221	3228	3234	3237	3238	3246	3247	3268	3275
	3281	3288	3294	3297	3298	3306	3307	3330	3335	3341	3348	3355	3361	3367	3368
	3377	3378	3390	3401	3406	3422	3423	3432	3433	3437	3457	3462	3478	3479	3487
	3488	3537	3542	3548	3551	3552	3562	3563	3573	3595	3624	3625	3634	3635	3681
	3689	3690	3700	3701	3710	3733	3766	3767	3777	3778	3821	3822	3832	3833	3925
	3928	3929	3937	3938	3980	3981	3994	3995	4072	4073	4081	4082	4116	4117	4125
	4126	4144	4160	4165	4170	4198	4203	4204	4214	4215	4286	4289	4290	4299	4300
	4323	4363	4366	4367	4376	4377	4399	4440	4443	4444	4453	4454	4474	4515	4518
	4519	4528	4529	4551	4592	4595	4596	4606	4607	4629	4670	4673	4674	4684	4685
	4707	4748	4751	4752	4761	4762	4785	4825	4828	4829	4838	4839	4862	4902	4905
	4906	4916	4917	4939	4980	4983	4984	4994	4995	5017	5058	5061	5062	5072	5073
	5095	5136	5139	5140	5150	5151	5171	5212	5215	5216	5226	5227	5249	5290	5293
	5294	5304	5305	5317	5374	5377	5378	5388	5389	5400	5458	5461	5462	5473	5474
	5486	5543	5546	5547	5557	5558	5569	5627	5630	5631	5641	5642	5653	5711	5714
	5715	5725	5726	5738	5795	5798	5799	5809	5810	5821	5879	5882	5883	5893	5894
	5905	5963	5966	5967	5977	5978	6029	6030	6039	6040	6102	6105	6106	6117	6118
	6184	6187	6188	6202	6203	6263	6268	6269	6275	6276	6316	6320	6324	6330	6333
	6360	6398	6424	6427	6439	6441	6468	6469	6537	6548	6605	6607	6621	6651	6656
	6658	6667	6737	6816	6893	6960	6963	7031	7033	7038	7059	7060	7070	7101	7118
	7127	7180	7226	7234	7256	7262									
.IFT	1627	6337	6344	6449	6631	7033	7038	7150	7170	7177					
.IFTF	1627	6337	6344	6447	6630	6978	7031	7034	7146	7154	7176				
.IIF	1	6	11	15	16	17	19	22	23	277	369	373	1570	1573	1579
	1580	1581	1583	1584	1611	1632	6317	6324	6325	6339	6346	6360	6361	6401	6402
	6403	6404	6405	6406	6410	6429	6448	6449	6465	6468	6469	6608	6609	6610	6611
	6612	6617	6643	6651	6656	6813	6960	6981	7109	7118	7124	7177	7284	7285	7286
	7287	7288	7290	7292	7293	7294	7295	7296	7297	7298					
.IRP	1549	1686	1751	1809	1858	1923	1992	2055	2118	2181	2230	2279	2345	2410	2479
	2548	2601	2661	2721	2781	2841	2901	2968	3035	3102	3169	3236	3296	3366	3421
	3477	3550	3623	3688	3765	3820	3927	3979	4071	4115	4202	4288	4365	4442	4517
	4594	4672	4750	4827	4904	4982	5060	5138	5214	5292	5376	5460	5545	5629	5713
	5797	5881	5965	6028	6104	6186	6267	6430	6470	6471	6472	6473	6474	6475	6476
	6477	6478	6479	6480	6481	6482	6483	6484	6485	6486	6487	6488	6489	6490	6491
	6492	6493	6494	6495	6496	6497	6498	6499	6500	6501	6502	6503	6504	6505	6506

.LIST	6507 6522 6554 1 1549 1992 2421 2901 3378 3927 4454 5060 5642 6267 6478 6493 6508 6523 7059 7297	6508 6523 6575 22 1585 2003 2479 2912 3421 3938 4517 5073 5713 6276 6479 6494 6509 6524 7276 7298	6509 6524 6591 133 1611 2055 2491 2968 3433 3979 4529 5138 5726 6324 6480 6495 6510 6525 7284 7299	6510 6525 6592 277 1614 2066 2548 2979 3477 3995 4594 5151 5797 6337 6481 6496 6511 6526 7285	6511 6526 6903 355 1627 2118 2561 3035 3488 4071 4607 5214 5810 6344 6482 6497 6512 6527 7286	6512 6527 6943 356 1686 2129 2601 3046 3550 4082 4672 5227 5881 6352 6483 6498 6513 6528 7287	6513 6528 7140 357 1697 2181 2614 3102 3563 4115 4685 5292 5894 6405 6484 6499 6514 6529 7288	6514 6529 7166 358 1751 2191 2661 3113 3623 4126 4750 5305 5965 6469 6485 6500 6515 6530 7289	6515 6530 7195 359 1763 2230 2674 3169 3635 4202 4762 5376 5978 6471 6486 6501 6516 6531 7290	6516 6531 7215 360 1809 2240 2721 3180 3688 4215 4827 5389 6028 6472 6487 6502 6517 6532 7291	6517 6532 361 1819 2279 2734 3236 3701 4288 4839 5460 6040 6473 6488 6503 6518 6533 7292	6518 6533 362 1858 2290 2791 3247 3765 4300 4904 5474 6104 6474 6489 6504 6519 6534 7293	6519 6534 363 1870 2345 2794 3296 3778 4365 4917 5545 6118 6475 6490 6505 6520 6535 7294	6520 6535 370 1923 2357 2841 3307 3820 4377 4982 5558 6186 6476 6491 6506 6521 6536 7295	6521 6553 373 1934 2410 2854 3366 3833 4442 4995 5629 6203 6477 6492 6507 6522 6651 7296
.MACRO	1 2345 3296 4365 5545	23 2410 3366 4442 5629	318 2479 3421 4517 5713	1601 2548 3477 4594 5797	1686 2601 3550 4672 5881	1751 2661 3623 4750 5965	1809 2721 3688 4827 6028	1858 2781 3765 4904 6104	1923 2841 3820 4982 6267	1992 2901 3927 5060 7276	2055 2968 3979 5138 7276	2118 3035 4071 5214	2181 3102 4115 5292	2230 3169 4202 5376	2279 3236 4288 5460 7296
.MCALL	1	133	370	1585	1614	1751	1809	1858	1923	1992	2055	2118	2181	2230	2279
.MEXIT	419														
.NLIST	1 1549 1992 2421 2901 3378 3927 4454 5060 5642 6267 6478 6493 6508 6523 7059 7297	22 1585 2003 2479 2912 3421 3938 4517 5073 5713 6276 6479 6494 6509 6524 7276 7298	133 1611 2055 2491 2968 3433 3979 4529 5138 5726 6324 6480 6495 6510 6525 7284 7299	277 1614 2066 2548 2979 3477 3995 4594 5151 5797 6337 6481 6496 6511 6526 7285	355 1627 2118 2561 3035 3488 4071 4607 5214 5810 6344 6482 6497 6512 6527 7286	356 1686 2129 2601 3046 3550 4082 4672 5227 5881 6352 6483 6498 6513 6528 7287	357 1697 2181 2614 3102 3563 4115 4685 5292 5894 6405 6484 6499 6514 6529 7288	358 1751 2191 2661 3113 3623 4126 4750 5305 5965 6469 6485 6500 6515 6530 7289	359 1763 2230 2674 3169 3635 4202 4762 5376 5978 6471 6486 6501 6516 6531 7290	360 1809 2240 2721 3180 3688 4215 4827 5389 6028 6472 6487 6502 6517 6532 7291	361 1819 2279 2734 3236 3701 4288 4839 5460 6040 6473 6488 6503 6518 6533 7292	362 1858 2290 2791 3247 3765 4300 4904 5474 6104 6474 6489 6504 6519 6534 7293	363 1870 2345 2794 3296 3778 4365 4917 5545 6118 6475 6490 6505 6520 6535 7294	370 1923 2357 2841 3307 3820 4377 4982 5558 6186 6476 6491 6506 6521 6536 7295	373 1934 2410 2854 3366 3833 4442 4995 5629 6203 6477 6492 6507 6522 6651 7296
.PAGE	318	420	1479	1525	1547	1684	6313	7299	7355	7565	7592	7758			
.REPT	277	355	6470												
.SBTTL	11 370	23 420	140 1479	159 1498	176 1517	192 1525	209 1547	228 1563	244 1607	261 1614	271 1684	280 1686	285 1751	296 1809	318 1858
	1923 2841 3763 4750 5965 6813	1992 2901 3765 4827 6028 6890	2055 2968 3820 4904 6104 6957	2118 3035 3927 4982 6186 7124	2181 3102 3979 5060 6265 7177	2230 3169 4071 5138 6267 7223	2279 3236 4115 5214 6313 7253	2345 3296 4200 5376 6363 7276	2410 3364 4202 5460 6382 7299	2479 3366 4288 5545 6395 7355	2548 3421 4365 4442 5545 7565	2601 3477 4442 5629 6537 7592	2661 3550 4517 5713 6602 7758	2721 3623 4594 5797 6658	2781 3688 4672 5881 6734
.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	1481	1482	1483	1484	1485	1486
1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1500	1501	1502	1503	1504
1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1519	1520	1521	1522
1523	1524	1527	1528	1529	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542
1543	1544	1545	1546	6329	6332	6359	6470	6471	6472	6473	6474	6475	6476	6477
6478	6479	6480	6481	6482	6483	6484	6485	6486	6487	6488	6489	6490	6491	6492
6493	6494	6495	6496	6497	6498	6499	6500	6501	6502	6503	6504	6505	6506	6507
6508	6509	6510	6511	6512	6513	6514	6515	6516	6517	6518	6519	6520	6521	6522
6523	6524	6525	6526	6527	6528	6529	6530	6531	6532	6533	6534	6535	6577	6679
6684	6715	6763	6810	6889	7173	7176	7249	7283	7301	7302	7305	7308	7311	7314
7316	7319	7322	7324	7326	7329	7331	7335	7337	7339	7340	7342	7345	7347	7350
7352	7357	7359	7361	7363	7365	7367	7369	7371	7373	7375	7377	7379	7381	7383
7385	7387	7389	7391	7393	7395	7397	7399	7401	7403	7405	7407	7409	7411	7413
7415	7417	7419	7421	7423	7425	7427	7429	7431	7433	7435	7437	7439	7441	7443
7445	7447	7449	7451	7453	7455	7457	7459	7461	7463	7465	7467	7469	7471	7473
7475	7477	7479	7481	7483	7485	7487	7489	7491	7493	7495	7497	7499	7501	7503
7505	7507	7509	7511	7513	7515	7517	7519	7521	7523	7525	7527	7529	7531	7533
7535	7537	7539	7541	7543	7545	7547	7549	7551	7553	7555	7557	7559	7561	7563

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\* DZR68B/CRF/SOL=SYSMAC.C1,DZR68B.P11  
RUN-TIME: 77 110 17 SECONDS  
RUN-TIME RATIO: 496/206=2.4  
CORE USED: 39K (77 PAGES)

