RK611/RK06

DRIVE DIAGNOSTIC PART 1

MD-11-DZR6H-D

EP-DZR6H-D-DL-A

FICHE 1 OF 2

APR 1977

digital

MADE IN USA

RK611/RK06

DRIVE DIAGNOSTIC PART 1

MD-11-DZR6H-D

EP-DZR6H-D-DL-A

FICHE 2 OF 2

APR 1977

digital

MADE IN USA

```
.REM %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6H-D-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 1
DATE:	JANUARY 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 1 OF THE DRIVE DIAGNOSTICS
 TO INSURE THAT THE
DISK IS CAPABLE OF PERFORMING ALL STATIC & CYCLE UP TESTS.
IT INSURES THAT THE DRIVE CAN WRITE AND READ HEADERS
 IN BOTH 20 & 22 SECTOR FORMATS.
FINALLY, IT INSURES THAT THE DISK CAN PERFORM SEEK

OPERATIONS BY DOING SEVERAL SEEK PATTERNS.
ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF PART 1, THE DRIVE
IS READY FOR PART 2 OF THE DRIVE DIAGNOSTICS.

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

*****CAUTION*****

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

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

PDP-11
CONSOLE TELETYPE
16K MEMORY
KW11-L OR KW11-P CLOCK
RK06 UNIBUS CONTROLLER (RK611)
1 TO 8 RK06 DRIVES

NOTES:
1. IF NEITHER KW11-L OR P CLOCK IS USED, ALL TIMING TESTS WILL BE BYPASSED. A MSG AT THE BEGINNING OF THE TESTS WILL CONFIRM THIS.

2. THE PROGRAM CAN WORK OFF EITHER FORMATTED OR NON-FORMATTED PACKS.

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFU

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

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

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

3.2 XXDP

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

CHAIN MODE OPERATION (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

DUMP MODE OPERATION (MANUAL)

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

3.3 ACT/APT

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

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

AUTOMATIC MODE (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

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

3.3.1 APT ETABLE DEFINITIONS

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

1. SOFTWARE ENVIRONMENT:
   =1 IF APT SCRIPT MODE
   =0 IF STANDALONE MODE

2. ENVIRONMENT MODE:

BIT 7 =	1	ETABLE DOES SIZING
	0	PROGRAM DOES SIZING
BIT 6 =	1	SPOOL MSGS TO APT IF SCRIPT MODE
	0	DON'T SPOOL TO APT
BIT 5 =	1	SUPPRESS CONSOLE OUTPUT
	0	ALLOW CONSOLE OUTPUT
BITS 4-0		NOT USED

3. SWITCH 1 (SOFTWARE SWITCH REGISTER)
IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1, THE SOFTWARE SWITCH REGISTER WILL BE USED, INSTEAD OF THE HARDWARE CONSOLE SWITCH REGISTER. REGARDLESS OF WHICH ONE IS USED, ALL BITS DEFINED IN SECTIONS 4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY USED WHEN RUNNING IN STANDALONE MODE. IN APT SCRIPT MODE, HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS BE SET TO 0.

4. SWITCH 2 (USER SWITCH REGISTER)
   NOT USED

5. CPU OPTIONS:
   NOT USED

6. MEMORY TYPES 1-4 AND MAX MEMORY ADDRESSES
   NOT USED

7. INTERRUPT VECTOR 1:
   USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 210

8. BUS PRIORITY 1:
   USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 5

9. INTERRUPT VECTOR 2:
   NOT USED

10. BUS PRIORITY 2:
    NOT USED

11. BASE ADDRESS:
    USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 177440

12. DEVICE MAP:
    USED WHEN ENVIRONMENT MODE BIT 7 = 1. EACH BIT SET TO 1 IN BITS 0-7 WILL SELECT THE CORRESPONDING DRIVE TO BE TESTED. BITS 8-15 ARE NOT USED.

13. CONTROLLER DESCRIPTOR WORDS:
NOT USED

14. DEVICE DESCRIPTOR CODES (IN WORDS):
NOT USED

3.4 DUAL ACCESS

THIS PROGRAM WILL NOT TEST OR SUPPORT DUAL-ACCESS. A DRIVE EQUIPED WITH DUAL ACCESS MUST BE SWITCHED TO THE PORT UNDER TEST TO PREVENT CONTENTION WITH THE OTHER PORT.

DUAL ACCESS TESTS WILL BE INCORPORATED IN A SEPARATE PROGRAM AT A LATER DATE.

3.5 MEMORY MANAGEMENT

MEMORY MANAGEMENT NOT USED

3.6 PARITY CHECK ENABLED

IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM, THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.

3.7 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR INFORMATION CONTAINED ON CYL 410, HEAD 2. PRINTOUTS OF DATA ERRORS DUE TO BAD SECTORS/TRACKS WILL BE MASKED OUT.

3.8 EXECUTION TIME

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

TOTAL TIME: 5 MIN, 30 SEC

A BREAKDOWN OF THE MORE LENGTHY TESTS ARE SHOWN BELOW:

TEST 16 STATIC CYL ADDRESS & DIFF REGS-PART 2: 2 MIN, 15 SEC
TEST 34 FORMAT PACK : 1 MIN
TEST 37 SEEK FROM CYL 0 TO ALL CYLS : 40 SEC
TEST 40 SEEK FROM CYL 410 TO ALL CYLS : 40 SEC

3.9 FAULT ISOLATION

TO BE DETERMINED.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OR FAILURE RATE

ANALYSIS.

3.11 DEFAULT UNIBUS ADDRESSES & VECTORS

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

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

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

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

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

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

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

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

4.1.4 PRESS 'START'

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

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

4.2 STARTING LOCATIONS

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

LOCATION 204 - SAME AS 200 START BUT BYPASS TEST 16 (N SQUARE)

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

LOCATION 230 - SAME AS 220 START BUT BYPASS TEST 16 (N SQUARE)

LOCATION 260 - RUN MODULE TEST ...DEFAULT MODE ONLY.
THIS SKIPS OVER THE FOLLOWING TESTS:

1. TEST 35 FORMAT PACK
2. TEST 36 DECREMENT FROM CYL 410 TO 0 & READ HEADERS
3. TEST 40 SEEK FROM CYL 0 TO ALL
4. TEST 41 SEEK FROM CYL 410 TO ALL

THE PURPOSE OF BYPASSING IS TO PROVIDE
A QUICK MODULE TEST

LOCATION 270 - SAME AS 260 START BUT BYPASS TEST 16 ALSO.

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

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

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

4.3.1 SW<15>

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

4.3.2 SW<14>

THE PROGRAM LOOPS ON THE TEST THAT IS BEING EXECUTED WHEN
THE SWITCH IS PUT ON. THIS SWITCH IS NORMALLY USED ALONG
WITH SW15.

4.3.3 SW<13>

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

4.3.4 SW<12>

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

4.3.5 SW<11>

EACH TEST WILL BE EXECUTED ONLY ONCE. NORMALLY AFTER THE
FIRST PASS, EACH SUBTEST IS ITERATED A NUMBER OF TIMES
(USUALLY 50, 5 IN SOME CASES). SETTING THIS SWITCH INHIBITS
ITERATIONS, SO THAT QUICK PASSES CAN BE MADE.

4.3.6 SW<10>

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

4.3.7 SW<09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK TO THE BEGINNING OF TEST.

4.3.8 SW<08>

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

4.4 'SOFTWARE' SWITCH REGISTER

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

SWR = NNNNNN NEW =

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

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

4.5 INPUT DIALOGUE

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

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

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

THE OPERATOR CAN ALSO TYPE IN THE SPECIFIC DRIVE NUMBERS TO BE TESTED, SEPARATED BY COMMAS & TERMINATED BY A CARRIAGE RETURN.

E.G. DRIVES TO BE TESTED: 1,2,4,6

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

4.5.2 BUS ADDRESS

THE REQUEST WILL BE:

TYPE IN BUSS ADDRESS IF NOT 177440

THE DEFAULT IS A CARRIAGE RETURN

4.5.3 CONTROLLER INTERRUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220.
ALL OPERATOR RESPONSES ARE UNDERLINED.

```
UNIBUS RK06 DRIVE DIAGNOSTIC
    PART 1
MAINDEC-11-DZR6H-D-PB

DRIVES TO BE TESTED: 1,3<CR>
                     -------

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>
                                   ----

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>
                                               ----
```

```
                                        WILL TEST DRIVES:
                                        1
                                        3

                                        DRIVE 1

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


                                4.6     PROGRAM EXAMPLE

                                        THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE
                                        DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.

                                        UNIBUS RK06 DRIVE DIAGNOSTIC
                                            PART 1
                                        MAINDEC-11-DZR6H-D-PB

                                        WILL TEST DRIVES:
                                        0
                                        1

                                        DRIVE 0

                                        DRIVE SERIAL NO. AAA
                                        CARTRIDGE SERIAL NO. BBB

                                        DRIVE 1

                                        DRIVE SERIAL NO. CCC
                                        CARTRIDGE SERIAL NO. DDD

                                        END PASS #1


                                        WILL TEST DRIVES:
                                        0
                                        1

                                        DRIVE 0


                                        DRIVE 1


                                        END PASS # 2

                                           (ETC)

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

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

4.7 HALTING THE PROGRAM

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

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

IF HEADS ARE LOADED & FORMATTING IS VALID,
THE PROGRAM WILL:

1. ECHO ↑C
2. TYPE "CPU HALTED"
3. HALT THE PROGRAM

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

1. ECHO ↑C
2. TYPE 'HALT PENDING, PLEASE WAIT'
3. DO THE TEST(S) THAT LOADS HEADS AND/OR FORMATS THE INVALID CYLS
4. TYPE 'CPU HALTED'
5. HALT THE PROGRAM

NOTES:

1. THE ABOVE EXAMPLE IS FOR THE PROGRAM RUNNING IN DUMP MODE (MANUAL). IF THE PROGRAM IS RUNNING IN CHAIN/AUTO MODE VIA XXDP,ACT,APT; IT WILL FIRST LOAD HEADS AND/OR FORMAT CORRECTLY, IF REQ'D, THEN IT WILL JUMP ON TO THE MONITOR WHERE THE NEXT PROGRAM CAN BE CALLED IN.

   THE TYPEOUTS WILL BE "ABORT PENDING - PLEASE WAIT"
   & "PROGRAM ABORTING"

2. OPERATING THE 'CONTINUE' SWITCH ON THE CPU CONSOLE WILL RETURN THE PROGRAM TO TEST 1 WHERE TESTING WILL BEGIN WITH THE 1'ST DRIVE AGAIN.

5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL

A. BASIC CONTROLLER TESTS, SIZING & SETUP

THESE TESTS DO BASIC CONTROLLER REGISTER REFERENCE TESTS, CHECKS OPERATOR INPUTS AGAINST DRIVES SEEN ON THE LINE OR DEFAULTS TO TEST ALL THE DRIVES SEEN ON THE LINE. IT CHECKS THE EXISTENCE OF AN L OR P CLOCKS FOR USE IN THE TIMING TESTS.

B. STATIC & CYCLE UP TESTS

```
                                THESE TESTS CHECK OUT THE ABILITY TO SELECT & DESELECT
                                THE DRIVE; TO DETECT PARITY, UNSAFE, AND FAULT CONDITIONS
                                WITH THE DRIVE READY TO OPERATE BUT WITHOUT THE SPINDLE
                                ON.

                                THE ENTIRE POWER UP SEQUENCE IS TESTED BY VERIFYING ALL
                                STATUS BITS SET/RESET IN PROPER SEQUENCE:  THE BRUSH
                                CYCLE, INNER-OUTER LIMIT DETECTION, FORWARD, REVERSE,
                                PIP...ETC STATUS BITS ARE CHECKED.

                            C.  SEEK, WRITE HEADER, READ HEADER TESTS

                                THESE TESTS CHECK THE ABILITY OF THE DRIVE TO DO SEEKS,
                                HEADER OPERATIONS & 20, 22 SECTOR FORMATTING.

                    5.2     TEST DESCRIPTIONS

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


                            TEST 1  REFERENCE ALL CONTROLLER REGISTERS

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


                            TEST 2  SIZE THE BUSS

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


                            TEST 3  VERIFY OPERATOR DRIVE SELECTIONS

                                THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
                                DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &

                                CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
                                PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
                                TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
                                BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
                                ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                                NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
```

```
                        NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
                        VERIFY IT WAS NOT SPECIFIED.


                TEST 4  FIND NEXT DRIVE TO BE TESTED

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

                TEST 5  UNLOAD DRIVE TO BE TESTED

                        THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
                        WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.
                        IT THEN WAITS FOR SPEED OK TO GO LOW BEFORE
                        PROCEEDING TO THE NEXT TEST.


                ;*******************************************************************
                STATIC & CYCLE UP TESTS
                ;*******************************************************************


                TEST 6  REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3

                         CHECKS THE ABILITY TO REFERENCE ALL
                        DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.


                TEST 7  PRINT DRIVE SERIAL NUMBER

                        THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
                        IN DECIMAL & IS PERFORMED ON THE 1ST PASS ONLY


                TEST 10 SET VV WITH PACK CMD

                        IF VV IS RESET, THE PACK CMD IS USED TO SET IT.


                TEST 11 RELEASE DRIVE

                        TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK


                TEST 12 DRIVE TYPE TEST

                        THIS TEST COMPARES DRIVE TYPE IN MSG A AGAINST 'DDT' IN RKDS.
                        WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.


                TEST 13 C-D PARITY ERROR DETECTION

                        TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
                        THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
```

```
                        THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
                        THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
                        CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
                        AN ERROR AND SET SPAR.
                        THE ERROR CONDITION IS RESET WITH THE CLEAR CMD


                TEST 14 VERIFY START SPINDLE CMD

                        THE PROGRAM CHECKS THE ENTIRE STARTUP SEQUENCE, IE:
                        BRUSH CYCLE, HEADS HOME, FWD, REV ETC.
                        BY VERIFYING ALL APPROPRIATE STATUS BITS FOR PROPER SEQUENCING.
                        THE CYL ADDRESS & CYL DIFFERENCE REGS ARE CHECKED
                        TO BE ZERO AT THE END OF THE SEQUENCE.


                ;*******************************************************************
                SEEK/READ HEADER/WRITE HEADER TESTS
                ;*******************************************************************


                TEST 15 STATIC CYL DIFF AND CYL ADDR REG TEST; PART 1

                        THIS TEST CHECKS EACH BIT OF THE CYL DIFFERENCE
                        AND CYL ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
                        MAJOR CYLS (0,1,2,4,8,16,32,64,128,256) WITH EVEN PARITY SET.
                        THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKIN
                        THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
                        NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.


                TEST 16 STATIC CYL DIFF & CYL ADDR REG TEST-PART 2

                        THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
                        DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
                        ALL CYLS FROM EVERY OTHER CYL. (N SQUARE SEEKS).
                        IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.


                TEST 17 HEAD REGISTER TEST

                        THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
                        VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
                        HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.

                        SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK CMDS,
                        SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
                        ILLEGAL ADDRESS. IF NOT,THIS MEANS THAT CHANGING HEAD ADDRESSES
                        ARE NOT TIED TO SEEK CMDS


                TEST 20 SEEK TO CYL 0

                        TESTS THE ABILITY TO DO A SEEK CMD.
                        VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
                        STATUS BITS.  VERIFIES CMD COMPLETION BETWEEN 10-15USEC.
```

```
                        READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.


                    TEST 21 TEST SECTOR COUNT REG. FOR 22 & 20 SECTOR FORMAT


                    TEST 22 DETECT OUTER LIMIT

                            THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYL 0
                            BY DETECTING OUTER LIMIT AS THE ADJACENT CYL.
                            AN ERROR IN THIS TEST INDICATES:

                                    A.  HEADS WERE NOT ON CYL 0
                            AND/OR  B.  COULD NOT SEEK IN REVERSE DIRECTION.


                    TEST 23 BASIC WRITE/READ HEADER & HEAD SWITCHING TEST

                            THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
                            ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
                            UNIQUE. 22 SECTOR FORMAT IS USED

                            I.E.    TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
                                    TRACK 1: 0101 FOR ALL SECTOR HEADERS
                                    TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS


                    TEST 24 BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS

                            USING HEAD 0, WRITE & READ 20 SECTOR HEADERS BY WRITING ALL
                            1'S AS HEADERS.  ATTEMPT TO FIND SECTORS 20 & 21.  VERIFY
                            THEY ARE NO LONGER THERE BY READING 22 SECTORS AND NOT
                            FINDING 0'S AS DATA FROM THE PREVIOUS TEST.


                    TEST 25 WRITE & READ HEADERS CYL 0, HEAD 0


                    TEST 26 SEEK FROM CYL 0 TO 1 & READ HEADERS

                            THIS TEST CHECKS MSG A & B WORDS 0,1,2 FOR CORRECT STATUS AFTER RDY
                            IS RECEIVED FROM A SEEK CMD TO DETERMINE
                            THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1.
                            AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS.
                            CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR
                            IN MSG B2 IS VERIFIED TO BE 1.

                            HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE
                            DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.


                    TEST 27 WRITE & READ HEADERS CYL 1, HEAD 0
```

```
                        TEST 30 TEST RECALIBRATE CMD & READ HEADERS

                                THIS TEST DOES A RECALIBRATE & READS HEADERS.
                                IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS
                                TEST DID NOT OVERWRITE CYL 0 HEADERS.

                                AN ERROR IN THIS TEST INDICATES THAT HEADS:


                                        A. MOVED TO A CYL OTHER THAN 1
                                OR      B. DID NOT GET BACK TO CYL 0


                        TEST 31 SINGLE INCREMENT SEEKS TO CYL 410

                                THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO CYL 410
                                WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY
                                DESTROY DATA.


                        TEST 32 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

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

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

                                THIS IS THE FIRST TEST WHERE THE READ DATA CMD IS PERFORMED


                        TEST 33 DETECT INNER LIMIT

                                THIS TEST VERIFIES THAT THE LAST CYL IN THE ABOVE
                                TEST WAS 410 BY DETECTING INNER LIMIT AS THE ADJACENT CYL.
                                IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON CYL 410
                                & THAT BSE INFO IS NOT VALID.  THE FORMAT PACK TEST
                                & ALL READ-WRITE TESTS ARE BYPASSED
                                TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
                                SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.


                        TEST 34 FORMAT PACK

                                THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
                                DOING 1 CYL INCREMENTAL SEEKS
                                FROM 0 TO 410 WITH WRITE HEADER CMDS (ALL TRACKS).
                                HEADERS WILL BE READ IN THE NEXT TEST
```

```
                        TEST 35 DECREMENT FROM CYL 410 TO 0 & READ HEADERS

                                THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
                                SINGLE CYL INCREMENTAL SEEKS.


                        TEST 36 SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS

                                THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLS & READS HEADERS.
                                IT THEN SEEKS CYL 0 & READS HEADERS.

                                MAJOR CYLS ARE: 1 (DECIMAL)  =  1 (OCTAL)
                                                2               2
                                                4               4
                                                8               10
                                                16              20
                                                32              40
                                                64              100
                                                128             200
                                                256             400


                        TEST 37 SEEK TO ALL CYLS FROM 0 & READ HEADERS


                        TEST 40 SEEK TO ALL CYLS FROM CYL 410 & READ HEADERS


                        TEST 41 SEEK TO ALL KEY INVALID CYLS

                                THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
                                PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
                                CYL IS SPECIFIED IN A SEEK.

                                INVALID CYLS ARE 411 THRU 511 (10)  IE. 633 THRU 777 (8)

                                THIS TEST CHECKS KEY INVALID CYLS 411,412,416,448 & 480
                                FOR A FULL LOGIC TEST


                        THE PROGRAM DOES NOT REQUIRE FORMATTED PACKS AS FORMATTING
                        IS PERFORMED IN ANY CASE.

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


                6.0     ERROR REPORTING


                6.1     ERROR INTERPRETATION

                        WHENEVER AN ERROR MSG IS PRINTED OUT, ALL REGISTERS
                        AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN.
                        MSG A(00), MSG B(01), RKER, RKBA...ETC, INDICATE THE
```

CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF ERROR.

EVERY ERROR MSG CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MSG, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVISED TO LOOK UP THE PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR MSGS ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLES:

EXAMPLE #1:

```
MSG A0 ERROR
AFTER START SPINDLE CMD & FWD SET

  TEST NO.  PC
  000014  016530
          EXPECT
A0        B0        A1        B1        A2        B2        B3
030144    100000    013704    000001
          ACTUAL
140144    100000    101744    000001
RKCS1     RKCS2     RKASOF    RKER      RKDS      RKDC
040200    000100    010000    000000    000000    000000
```

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR MSG REGISTERS A0, B0, A1 & B1.

MSGS A2, B2 & B3 WILL BE TYPED OUT ONLY AS REQUIRED IF THE CYL DIFFERENCE/OFFSET, CYL ADDRESS & HEAD & SECTOR INFORMATION IS A VARIABLE PARAMETER OF THE TEST.

EXAMPLE #2:

```
NO ATTN IN RKASOF
AFTER UNLOAD CMD

  TEST NO.  PC
  000003  014330
RKMR2     RKMR3     RKER      RKDS      RKCS1     RKCS2     RKASOF
```

```
1107                                    000144  100000  000000  100101  000206  000104  000000
1108
1109
1110
1111                                            [ END OF DOCUMENT ]
1112
1113
1114                                %
```

```
                        ; *** PGM REV 035 ***
                        .NLIST  CND,MC,MD
                        .LIST   ME
                        .ENABL  ABS,AMA

                        ;DEFINE SYSMAC MACROS

167400                  $SWR=   167400                  ;DEFINE SWITCHES 15,14,13,11,10,9,8
000001                  $TN=    1                       ;SET FIRST TEST NO. TO 1


                        .TITLE  UNIBUS RK06 DRIVE DIAGNOSTIC  PART 1
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY GARY PAPAZIAN
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*


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


                        .SBTTL  SUMMARY OF STARTING LOCATIONS
                        ;*
                        ;*
                        ;*              200     DEFAULT PARAMETERS
                        ;*              204     DEFAULT PARAMETERS & BYPASS TEST 16
                        ;*              220     INPUT PARAMETERS
                        ;*              230     INPUT PARAMETERS & BYPASS TEST 16
                        ;*              240     ODT11
                        ;*              260     RUN MODULE TEST VERSION-DEFAULT MODE ONLY BYPASS
                        ;*                      TESTS 35,36,40 & 41
                        ;*              270     SAME AS 260 START BUT BYPASS TEST 16 ALSO
                        ;*
```

```
                        .SBTTL  BASIC DEFINITIONS

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

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

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

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
```

```
                    SW01=   2
                    SW00=   1
                    .EQUIV  SW09,SW9
                    .EQUIV  SW08,SW8
                    .EQUIV  SW07,SW7
                    .EQUIV  SW06,SW6
                    .EQUIV  SW05,SW5
                    .EQUIV  SW04,SW4
                    .EQUIV  SW03,SW3
                    .EQUIV  SW02,SW2
                    .EQUIV  SW01,SW1
                    .EQUIV  SW00,SW0

                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000              BIT15=  100000
040000              BIT14=  40000
020000              BIT13=  20000
010000              BIT12=  10000
004000              BIT11=  4000
002000              BIT10=  2000
001000              BIT09=  1000
000400              BIT08=  400
000200              BIT07=  200
000100              BIT06=  100
000040              BIT05=  40
000020              BIT04=  20
000010              BIT03=  10
000004              BIT02=  4
000002              BIT01=  2
000001              BIT00=  1
                    .EQUIV  BIT09,BIT9
                    .EQUIV  BIT08,BIT8
                    .EQUIV  BIT07,BIT7
                    .EQUIV  BIT06,BIT6
                    .EQUIV  BIT05,BIT5
                    .EQUIV  BIT04,BIT4
                    .EQUIV  BIT03,BIT3
                    .EQUIV  BIT02,BIT2
                    .EQUIV  BIT01,BIT1
                    .EQUIV  BIT00,BIT0

                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004              ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010              RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014              TBITVEC=14              ;;"T" BIT
000014              TRTVEC= 14              ;;TRACE TRAP
000014              BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020              IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024              PWRVEC= 24              ;;POWER FAIL
000030              EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034              TRAPVEC=34              ;;"TRAP" TRAP
000060              TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064              TPVEC=  64              ;;TTY PRINTER VECTOR
000240              PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                    .SBTTL  RK06 CONTROLLER REGISTER DEFINITION
```

```
1280
1281                            ;       $BASE=177440
1282
1283    000000                  RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1284    000002                  RKWC=   2               ;WORD COUNT REGISTER
1285    000004                  RKBA=   4               ;BUS ADDRESS REGISTER
1286    000006                  RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1287    000010                  RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1288    000012                  RKDS=   12              ;DRIVE STATUS REGISTER
1289    000014                  RKER=   14              ;ERROR REGISTER
1290    000016                  RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1291    000020                  RKDC=   20              ;DESIRED CYL REGISTER
1292    000024                  RKDB=   24              ;DATA BUFFER
1293    000026                  RKMR1=  26              ;MAINTENANCE REGISTER 1
1294    000034                  RKMR2=  34              ;MAINTENANCE REGISTER 2 (MSG LINE A)
1295    000036                  RKMR3=  36              ;MAINTENANCE REGISTER 3 (MSG LINE B)
1296    000030                  RKECPS= 30              ;ECC POSITION INFORMATION
1297    000032                  RKECPT= 32              ;ECC PATTERN INFORMATION
1298
1299                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1300
1301                            ;       DRIVE CMDS
1302
1303    000001                  SELDRV= 1               ;SELECT DRIVE (GET STATUS)
1304    000003                  PACK=   3               ;PACK ACKNOWLEDGE
1305    000005                  CLEAR=  5               ;DRIVE CLEAR
1306    000007                  UNLOAD= 7               ;UNLOAD
1307    000011                  SRTSPL= 11              ;START SPINDLE
1308    000013                  RECAL=  13              ;RECALIBRATE
1309    000015                  OFFSET= 15              ;OFFSET
1310    000017                  SEEK=   17              ;SEEK
1311    000021                  RDDATA= 21              ;READ DATA
1312    000023                  WRDATA= 23              ;WRITE DATA
1313    000025                  RDHEAD= 25              ;READ HEADER
1314    000027                  WRHEAD= 27              ;WRITE HEADER AND DATA
1315    000031                  WRTCHK= 31              ;WRITE CHECK
1316
1317    000001                  GO=     BIT0            ;GO BIT
1318    000100                  IE=     BIT6            ;INTERRUPT ENABLE
1319    000200                  RDY=    BIT7            ;CONTROLLER READY
1320    000400                  BA16=   BIT8            ;BUS ADDRESS BIT 16
1321    001000                  BA17=   BIT9            ;BUS ADDRESS BIT 17
1322    002000                  CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06)
1323    004000                  CTO=    BIT11           ;CONTROLLER TIMEOUT
1324    010000                  CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1325    020000                  DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1326    040000                  DI=     BIT14           ;DRIVE INTERRUPT
1327    100000                  CERR=   BIT15           ;CONTROLLER ERROR
1328    100000                  CCLR=   BIT15           ;CONTROLLER CLEAR
1329
1330                            .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1331
1332    000007                  DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
1333    000010                  RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1334    000020                  BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
1335    000040                  SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
1336            000100          IR=     BIT6            ;INPUT READY
1337            000200          OR=     BIT7            ;OUTPUT READY
1338            000400          UFE=    BIT8            ;UNIT FIELD ERROR
1339            001000          MDS=    BIT9            ;MULTIPLE DRIVE SELECT
1340            002000          PGE=    BIT10           ;PROGRAMMING ERROR
1341            004000          NEM=    BIT11           ;NON-EXISTENT MEMORY
1342            010000          NED=    BIT12           ;NON-EXISTENT DRIVE
1343            020000          UPE=    BIT13           ;UNIBUS PARITY ERROR
1344            040000          WCE=    BIT14           ;WRITE CHECK ERROR
1345            100000          DLT=    BIT15           ;DATA LATE ERROR
1346
1347                            .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1348
1349            000001          ILF=    BIT0            ;ILLEGAL FUNCTION CODE
1350            000002          SKI=    BIT1            ;SEEK INCOMPLETE
1351            000004          NXF=    BIT2            ;NON-EXECUTABLE FUNCTION
1352            000010          DRPAR=  BIT3            ;DRIVE DETECTED SERCON PARITY ERROR
1353            000020          FMTE=   BIT4            ;FORMAT ERROR
1354            000040          DTYE=   BIT5            ;DRIVE TYPE ERROR
1355            000100          ECH=    BIT6            ;ECC HARD
1356            000200          BSE=    BIT7            ;BAD SECTOR ERROR
1357            000400          HVRC=   BIT8            ;HEADER VRC ERROR
1358            001000          COE=    BIT9            ;CYL ADDRESS OVERFLOW ERROR
1359            002000          IDAE=   BIT10           ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1360            004000          WLE=    BIT11           ;WRITE LOCK ERROR
1361            010000          DTE=    BIT12           ;DRIVE TIMING ERROR
1362            020000          OPI=    BIT13           ;OPERATION (SEARCH) INCOMPLETE
1363            040000          UNS=    BIT14           ;DRIVE UNSAFE
1364            100000          DCK=    BIT15           ;DATA CHECK
1365
1366                            .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1367
1368            000001          DRA=    BIT0            ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1369                                                    ;    THIS BIT IS RESET)
1370            000004          OFST=   BIT2            ;DRIVE OFFSET
1371            000010          ACLO=   BIT3            ;AC LOW
1372            000020          DCLO=   BIT4            ;DC LOW
1373            000040          DROT=   BIT5            ;DRIVE OFF TRACK
1374            000100          VV=     BIT6            ;VOLUME VALID
1375            000200          DRDY=   BIT7            ;DRIVE READY
1376            000400          DDT=    BIT8            ;DRIVE TYPE (0=RK06)
1377            004000          WRL=    BIT11           ;WRITE LOCK
1378            020000          PIP=    BIT13           ;POSITIONING IN PROGRESS
1379            040000          DSC=    BIT14           ;DRIVE STATUS CHANGE
1380            100000          SVAL=   BIT15           ;STATUS VALID
1381
1382                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1383
1384            000017          MESMSK= 17              ;MSG MASK
1385            000020          PAT=    BIT4            ;FORCE EVEN PARITY ON SERCON MSG LINES
1386            000040          DMD=    BIT5            ;DIAGNOSTIC MODE
1387            000100          MSP=    BIT6            ;MAINTENANCE SECTOR PULSE
1388            000200          MIND=   BIT7            ;MAINTENANCE INDEX
1389            000400          MCLK=   BIT8            ;MAINTENANCE CLOCK
1390            001000          MERD=   BIT9            ;MAINTENANCE ENCODED READ DATA
1391            002000          MEWD=   BIT10           ;MAINTENANCE ENCODED WRITE DATA
```

```
1392    004000          PCA=    BIT11           ;PRECOMPENSATION ADVANCE
1393    010000          PCD=    BIT12           ;PRECOMPENSATION DELAY
1394    020000          ECCW=   BIT13           ;ECC WORD IS BEING READ OR WRITTEN
1395    040000          WRTGAT= BIT14           ;WRITE GATE
1396    100000          RDGATE= BIT15           ;READ GATE
1397
1398                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MSG A (RKMR2:34)
1399
1400    000040          D.DRA=  BIT5            ;DRIVE AVAILABLE
1401    000100          D.VV=   BIT6            ;VOLUME VALID
1402    000200          D.DRDY= BIT7            ;DRIVE READY
1403    000400          D.DDT=  BIT8            ;DRIVE TYPE (0=RK06)
1404    001000          D.FORM= BIT9            ;DRIVE FORMAT
1405    002000          D.OFF=  BIT10           ;OFFSET ON
1406    004000          D.WRL=  BIT11           ;WRITE LOCK
1407    010000          D.SPIN= BIT12           ;SPINDLE ON
1408    020000          D.PIP=  BIT13           ;POSITIONING IN PROGRESS
1409    040000          D.DSC=  BIT14           ;DRIVE STATUS CHANGE
1410
1411                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MSG A (RKMR2:34)
1412
1413    000020          D.SSP=  BIT4            ;SERVO SIG PRESENT
1414    000040          D.HDHM= BIT5            ;HEADS HOME
1415    000100          D.BRHM= BIT6            ;BRUSHES HOME
1416    000200          D.DOOR= BIT7            ;DOOR INTERLOCKED
1417    000400          D.CART= BIT8            ;CARTRAGE INTERLOCK
1418    001000          D.SPOK= BIT9            ;SPEED OK
1419    002000          D.FWD=  BIT10           ;FORWARD
1420    004000          D.REV=  BIT11           ;REVERSE
1421    010000          D.LOAD= BIT12           ;HEADS LOADING
1422    020000          D.RTZ=  BIT13           ;RETURN TO ZERO
1423    040000          D.UNLD= BIT14           ;HEADS UNLOADING
1424
1425                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MSG B (RKMR3:36)
1426
1427    000040          D.IDAE= BIT5            ;INVALID DISK ADDRESS ERROR:HEAD/CYL
1428    000100          D.ACLO= BIT6            ;AC LOW
1429    000200          D.FLT=  BIT7            ;DRIVE FAULT
1430    000400          D.ILF=  BIT8            ;ILLEGAL FUNCTION CODE
1431    001000          D.PAR=  BIT9            ;DRIVE DETECTED SERCON PARITY ERROR
1432    002000          D.SKI=  BIT10           ;SEEK INCOMPLETE
1433    004000          D.WLE=  BIT11           ;WRITE LOCK ERROR
1434    010000          D.SPLS= BIT12           ;SPEED LOSS
1435    020000          D.DROT= BIT13           ;DRIVE OFF TRACK
1436    040000          D.UNS=  BIT14           ;R/W UNSAFE
1437
1438                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MSG B (RKMR3:36)
1439
1440    000020          D.SECT= BIT4            ;SECTOR ERROR
1441    000040          D.WCUR= BIT5            ;WRITE CURRENT AND NO WRITE GATE
1442    000100          D.WGAT= BIT6            ;WRITE GATE AND NO TRANSISTIONS
1443    000200          D.HDFL= BIT7            ;HEAD FAULT
1444    000400          D.MHD=  BIT8            ;MULTIPLE HEAD SELECT
1445    001000          D.XERROR=       BIT9            ;INDEX ERROR
1446    002000          D.TIB=  BIT10           ;TRIBIT ERROR
1447    004000          D.PLO=  BIT11           ;PLO ERROR
```

```
1448            010000                  D.NMOV= BIT12                   ;SEEK AND NO MOTION
1449            020000                  D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1450            040000                  D.SUNS= BIT14                   ;SERVO UNSAFE
1451
1452                                    .SBTTL  COMMON MASKS AND OTHER BITS: MSG A (RKMR2:34)
1453
1454            000007                  M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1455            017760                  M.CDIF= 17760                   ;CYL DIFF, BYTE 10
1456            017760                  M.OFST= 17760                   ;OFFSET VALUE, BYTE 10
1457            077770                  M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1458
1459                                    .SBTTL  COMMON MASKS AND OTHER BITS: MSG B (RKMR3:36)
1460
1461            000003                  M.ID=   3                       ;BYTE ID, ALL BYTES
1462            017760                  M.CADD= 17760                   ;CYL ADDRESS, BYTE 10
1463            040000                  M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1464            000760                  M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1465            007000                  M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1466            100000                  M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
1467
1468                                    .SBTTL  TRAP CATCHER
1469
1470            000000                          .=0
1471                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1472                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1473                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1474            000174                          .=174
1475  000174   000000                  DISPREG:  .WORD  0              ;;SOFTWARE DISPLAY REGISTER
1476  000176   000000                  SWREG:    .WORD  0              ;;SOFTWARE SWITCH REGISTER
1477                                    .SBTTL  STARTING ADDRESS(ES)
1478  000200   000137  007132                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1479            000204                        .=204
1480  000204   000137  007026                  JMP     BYT16           ;BYPASS N-SQUARE TEST IN DEFAULT MODE
1481            000220                        .=220
1482  000220   000137  007006                  JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING
1483            000230                        .=230
1484  000230   000137  007046                  JMP     BYT16A  ;BYPASS N-SQUARE TEST IN PARAM MODE
1485            000240                        .=240
1486  000240   000137  067712                  JMP     0.ODT           ;ENTER ODT11
1487            000260                        .=260
1488  000260   000137  007070                  JMP     MDTST           ;MODULE TESTS DEFAULT MODE ONLY
1489            000270                        .=270                   ;BYPASS SEVERAL TESTS
1490  000270   000137  007110                  JMP     MDTSTA          ;SAME AS 260 & BYPASS N-SQUARE TEST ALSO
1491
1492                                    .SBTTL  ACT11 HOOKS
1493
1494                                    ;;*****************************************************************
1495                                    ;HOOKS REQUIRED BY ACT11
1496            000274                          $SVPC=.                 ;SAVE PC
1497            000046                          .=46
1498  000046   043220                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1499            000052                          .=52
1500  000052   100000                          .WORD   100000          ;;2)SET LOC.52 TO 100000
1501            000274                          .=$SVPC                 ;; RESTORE PC
1502            001000                          .=1000
1503                                    .SBTTL  APT PARAMETER BLOCK
1504
1505                                    ;;*****************************************************************
1506                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1507                                    ;;*****************************************************************
1508            001000                          .$X=.   ;;SAVE CURRENT LOCATION
1509            000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1510  000024   000200                          200     ;;FOR APT START UP
1511            000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1512  000044   001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
1513            001000                          .=.$X   ;;RESET LOCATION COUNTER
1514                                    ;;*****************************************************************
1515                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1516                                    ;INTERFACE SPEC.
1517
1518  001000                            $APTHD:
1519  001000   000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1520  001002   001210                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1521  001004   000430                  $TSTM:  .WORD   280.    ;;RUN TIM OF LONGEST TEST
1522  001006   001130                  $PASTM: .WORD   600.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
```

```
1523  001010  001130  $UNITM: .WORD   600.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1524  001012  000042          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1525
1526
1527                          .LIST   MD
1528                  ;
1529                  ;USE LOOP X TO OMIT JSR PC, SUBCLR
1530                  ;
1531
1532                  .MACRO  LOOP    A
1533                          SCOP1
1534                          MOV     #STACK,SP       ;RESTORE STK PTR
1535
1536                  .IF     B       A
1537                          JSR     PC,SUBCLR
1538                          ERROR   24              ;CERR AFTER SCLR
1539
1540                  .ENDC
1541                  .ENDM   LOOP
1542
1543                  ;
1544                  ; THIS MACRO FILLS EXPECTED MSG A0,B0,A1,B1,A2,B2 & B3 WITH STANDARD BITS SET
1545                  ; A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1546                  ; NOTE: A CAN BE ANY BIT COMBINATION DESIRED
1547
1548                  .MACRO  F.EAB   A
1549
1550                          MOV     #<A!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
1551                          CLR     E.B0            ;EXPECTED MSG B0
1552                          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
1553                          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
1554                          CLR     E.A2            ;EXPECTED MSG A2
1555                          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
1556                          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
1557                  .ENDM   F.EAB
1558                  ;
1559                  ; THIS MACRO ASSUMES DRIVE MSG A0,B0,A1,B1 WILL ALWAYS BE TESTED
1560                  ; USE   A,C,D,E FOR MSG A0,B0,A1,B1 ERROR NUMBERS RESP.
1561                  ; USE   G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1562                  ;       H=T.B2  TO READ MSG B2 & PUT INFOR INTO 'CYLADD'
1563                  ;       I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
1564                  ;
1565                  ;       F=      < ERROR DESCRIPTION>
1566                  ;
1567                  .MACRO  CHECK   A,C,D,E,F,G,H,I
1568
1569                          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
1570                          .WORD   G!H!I           ;& MSGS SPECIFIED HERE
1571                          ERROR   A               ;MSG A0 ERROR F
1572                          ERROR   C               ;MSG B0 ERROR
1573                          ERROR   D               ;MSG A1 ERROR
1574                          ERROR   E               ;MSG B1 ERROR
1575                  .ENDM   CHECK
1576                  ;
1577                  ; A=CYL DIFF/OFFSET ERROR #
1578                  ; B=CYL ADDR ERROR #
```

```
1579                                    ; C= <ERROR DESCRIPTION>
1580                                    ;
1581                                    .MACRO  CWD2    A,B,C,?D,?E
1582
1583                                            TST     CYLDIF          ;SEE IF MSG A2=0
1584                                            BEQ     D               ;BR IF YES
1585                                            ERROR   A               ;MSG A2 NOT CLEARED C
1586                                    D:      TST     CYLADD          ;SEE IF MSG B2=0
1587                                            BEQ     E               ;BR IF YES
1588                                            ERROR   B               ;MSG B2 NOT CLEARED C
1589                                    E:
1590                                    .ENDM   CWD2
1591
1592
1593                                    .MACRO  LPCHK   ?A
1594                                            CLR     $ESCAPE
1595                                            TST     LPFLG
1596                                            BEQ     A
1597                                            JMP     @$LPERR         ;SW 9 WAS SET.
1598                                    A:      JMP     @$LPADR         ;SW 14 OR 8 WAS SET
1599                                    .ENDM   LPCHK
1600
1601                                    .MACRO  SW814
1602                                            JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
1603                                            SKIP R,<GO TO NEXT TEST>        ;RETURN HERE IF NEITHER IS SET
1604                                                                    ;RETURN HERE IF SW 14 IS SET OR
1605                                                                    ;SW 8 WITH SWR <7:0> APPLY
1606                                    .ENDM   SW814
1607
1608                                    ;
1609                                    ;SWR9 (LOOP ON ERROR) TEST      A=BRANCH POINT TO RECONDITION DRIVE
1610                                    ;                               B=JMP POINT TO RE-ENTER MAIN LINE
1611                                    .MACRO  TSTSW9  A,B
1612                                            INC     LPFLG
1613                                            BIT     #SW9,@SWR       ;LOOP ON ERROR?
1614                                            BNE     A               ;YES, RECONDITION DRIVE
1615                                            JMP     B               ;RETURN TO MAINLINE
1616                                    .ENDM   TSTSW9
1617
1618
1619                                    ;
1620                                    ; USE DRCLR     X    TO OMIT CHECKING MSG A0,B0,A1 & B1
1621                                    ;
1622                                    .MACRO  DRCLR   A,?C
1623
1624                                            MOV     #CCLR,RKCS1(R5)
1625                                            MOV     $UNIT,RKCS2(R5) ;DRIVE#
1626                                            MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
1627                                            MOV     T10,TEMP1
1628                                            JSR     PC,FRDY         ;FIND RDY
1629                                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
1630                                            JSR     PC,TSTATN       ;TEST FOR ATTN
1631                                            BR      C
1632                                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1633                                    C:
1634                                    .IF B   A
```

```
                        F.EAB   0
                        CHECK   273,265,274,266,<AFTER DRIVE CLEAR CMD>,T.A2,T.B2,0
                .ENDC

                .ENDM   DRCLR


                ;
                ;A=BLANK TO CHECK A0 THRU B2
                ;A=NON BLANK TO OMIT CHECKING A0 THRU B2
                ;D=BLANK TO CHECK A0 THRU B2 IN DRCLR
                ;D=NON-BLANK TO OMIT CHECKING A0 THRU B2 IN DRCLR
                ;
                .MACRO  CALIB   A,D,?C

                        MOV     #CCLR,RKCS1(R5)
                        MOV     $UNIT,RKCS2(R5)
                        MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                                ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                                ;IN RKMR2 & RKMR3 RESP.
                        MOV     T10,TEMP1
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD

                        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                        JSR     PC,GSTAT
                        BIT     #D.RTZ,HMR2
                        BNE     C
                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
                C:      MOV     T10,TEMP2       ;SETUP TIMEOUT
                        JSR     PC,FATT1        ;FIND ATTN
                        ERROR   55              ;NO ATTN AFTER RECAL CMD
                .IF B   A
                        F.EAB   D.DSC
                        CHECK   221,275,222,276,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
                        CWD2    47,50,<AFTER RECAL CMD>
                .ENDC
                        DRCLR   D

                .ENDM   CALIB

                ;
                ;IDAE IS CLEARED ONLY BY RECAL & DRIVE CLEAR
                ;
                .MACRO  CIDAE   ?A

                        MOV     #CCLR,RKCS1(R5)
                        MOV     $UNIT,RKCS2(R5)
                        MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                        MOV     T10,TEMP1
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
                        DRCLR   X

                        JSR     PC,GSTAT
                        BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
```

```
1691                                    BEQ     A               ;BR IF YES
1692                                    ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
1693
1694                            A:      MOV     #CCLR,RKCS1(R5)
1695                                    MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
1696                                    JSR     PC,FATT1
1697                                    ERROR   55              ;NO ATTN AFTER RECAL CMD
1698
1699                            .ENDM   CIDAE
1700
1701                            ;
1702                            ; A=D.FWD/D.REV
1703                            ;
1704                            .MACRO  SKRDY A
1705
1706                                    MOV     #SEEK,RKCS1(R5) ;SEEK CMD
1707                                    MOV     T10,TEMP1       ;SETUP  TIMEOUT
1708                                    JSR     PC,FRDY         ;FIND RDY
1709                                    ERROR   131             ;NO RDY AFTER SEEK CMD
1710                                    MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
1711                                    CLR     E.B0
1712                                    MOV     #<A!D.SPOR!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
1713                                    MOV     #1,E.B1
1714                                    CHECK   203,204,205,206,<DURING SEEK CMD>,T.A2,T.B2,0
1715
1716                            .ENDM   SKRDY
1717
1718                            .MACRO  SKATN ?A,?B
1719
1720                                    JSR     PC,FATT2        ;FIND ATTN
1721                                    ERROR   132             ;NO ATTN AFTER SEEK CMD
1722                                    BIT     #CERR,HCS1
1723                                    BEQ     A
1724                                    ERROR   210             ;CERR AFTER SEEK CMD
1725                            A:      F.EAB   D.DSC
1726                                    CHECK   133,134,135,136,<AFTER SEEK CMD>,T.A2,T.B2,0
1727                                    TST     CYLDIF
1728                                    BEQ     B
1729                                    ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
1730
1731                            B:      DRCLR
1732                            .ENDM   SKATN
1733
1734                            ;
1735                            ;QUICK START SPINDLE.
1736                            ;
1737                            .MACRO  QKSRT   A
1738
1739                                    JSR     PC,SUBCLR
1740                                    ERROR   24              ;CERR AFTER SCLR
1741
1742                                    MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
1743                                    MOV     T10,TEMP1       ;SET TIMEOUT
1744                                    JSR     PC,FRDY         ;FIND RDY
1745                                    ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
1746
```

```
1747                                    MOV     T500,TEMP2      ;SETUP  TIMEOUT
1748                                    JSR     PC,FATT1        ;FIND ATTN
1749                                    ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
1750
1751                                    CLR     UNLD
1752                                .IF B   A
1753                                    TSTSW9  10$,2$
1754                                .ENDC
1755                                .ENDM   QKSRT
1756
1757                                ;
1758                                ;QUICK SEEK.    ENTER WITH CYL # IN RKDC
1759                                ;
1760                                .MACRO  QKSEEK  ?A
1761
1762                                    MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
1763                                    MOV     T10,TEMP1       ;SETUP TIMEOUT
1764                                    JSR     PC,FRDY         ;FIND RDY
1765                                    ERROR   131             ;NO RDY AFTER SEEK CMD.
1766
1767                                    MOV     T50000,TEMP1
1768                                    JSR     PC,FATT2        ;FIND ATTN
1769                                    ERROR   132             ;NO ATTN AFTER SEEK CMD
1770                                    BIT     #CERR,HCS1
1771                                    BEQ     A
1772                                    ERROR   210             ;CERR AFTER SEEK CMD.
1773
1774                                A:  JSR     PC,SUBCLR
1775                                    ERROR   24              ;CERR AFTER SCLR
1776
1777                                .ENDM   QKSEEK
1778
1779                                ;
1780                                ;QUICK REPETITIVE SEEKS
1781                                ;       A=INC/DEC    CYL#
1782                                ;       B=FINAL VALUE OF CYL# BEFORE EXITING
1783
1784                                .MACRO  QKRPSK  A,B,?C,?D
1785
1786                                    JSR     PC,SUBCLR
1787                                    ERROR   24              ;CERR AFTER SCRL
1788
1789                                C:  MOV     TOCYL,RKDC(R5)  ;CYL#
1790                                    QKSEEK
1791                                    CMP     TOCYL,#B        ;ALL CYL DONE?
1792                                    BEQ     D               ;BR IF YES
1793                                    A       TOCYL           ;ELSE DO ANOTHER
1794                                    BR      C
1795
1796                                D:  JSR     PC,SUBCLR
1797                                    ERROR   24              ;CERR AFTER SCLR
1798
1799                                    LPCHK
1800
1801                                .ENDM   QKRPSK
1802
```

```
1803                                    ;
1804                                    ;QUICK UNLOAD
1805                                    ;D=BLANK TO DO SUBCLR & LPCHK
1806                                    ;D=NON-BLANK TO BYPASS
1807                                    ;
1808                                    .MACRO  QKUNLD  D
1809
1810                                            JSR     PC,SUBCLR
1811                                            ERROR   24              ;CERR AFTER SCLR
1812
1813                                            MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
1814                                            MOV     T10,TEMP1
1815                                            JSR     PC,FRDY         ;FIND RDY
1816                                            ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
1817                                            JSR     PC,TSTATN
1818                                            ERROR   12              ;NO ATTN AFTER UNLOAD CMD
1819
1820                                            JSR     PC,SUBCLR
1821                                            ERROR   24              ;CERR AFTER SCLR
1822
1823                                            MOV     T10,TEMP2
1824                                            JSR     PC,FSPOK
1825                                            ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
1826                                    .IF     B       D
1827
1828                                            JSR     PC,SUBCLR
1829                                            ERROR   24              ;CERR AFTER SCLR
1830
1831                                            LPCHK
1832                                    .ENDC
1833
1834
1835
1836                                    .ENDM   QKUNLD
1837
1838                                    ;
1839                                    ; A=WRHEAD/<CFMT!WRHEAD>
1840                                    ; USE WRHDR     <A>,X   TO OMIT CHECKING A0,B0,A1,B1
1841                                    ;
1842                                    .MACRO  WRHDR   A,C,?D
1843
1844                                            MOV     #<A>,RKCS1(R5)  ;WRITE HEADER CMD
1845                                            MOV     T50000,TEMP1    ;SETUP TIMEOUT
1846                                            JSR     PC,FRDY         ;FIND RDY
1847                                            ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
1848                                            JSR     PC,GSTAT        ;GET FRESH STATUS
1849                                            BIT     #CERR,HCS1
1850                                            BEQ     D
1851                                            ERROR   201             ;CERR AFTER WRITE HEADER CMD
1852                                            TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
1853                                            JMP     $EOP            ;ABORT DRIVE
1854                                    D:
1855                                    .IF B   C
1856                                            F.EAB   0
1857                                            CHECK   277,267,300,270,<AFTER WRITE HEADER CMD>,T.A2,T.B2,0
1858                                    .ENDC
```

```
1859
1860                                    .ENDM   WRHDR
1861
1862                                    ;
1863                                    ; A=RDHEAD/<CFMT!RDHEAD>
1864                                    ; USE RDHDR <A>,X TO OMIT CHECKING A0,B0,A1,B1
1865                                    ;
1866                                    .MACRO  RDHDR   A,C,?D,?E
1867
1868                                            MOV     #RHTAB,R0
1869                                            MOV     #<A>,RKCS1(R5)  ;READ HEADER CMD
1870                                            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
1871                                            JSR     PC,FRDY         ;FIND RDY
1872                                            ERROR   171             ;NO RDY AFTER READ HEADER CMD
1873                                            BIT     #CERR,HCS1
1874                                            BEQ     D
1875                                            ERROR   174             ;CERR AFTER READ HEADER CMD
1876                                            TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
1877                                            JMP     $EOP            ;ABORT DRIVE
1878
1879                                    D:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
1880                                            MOV     RKDB(R5),(R0)+  ;2'ND WORD
1881                                            MOV     RKDB(R5),(R0)+  ;3'RD WORD
1882
1883
1884                                            BIT     #DLT,RKCS2(R5)
1885                                            BEQ     E
1886                                            JSR     PC,GSTAT
1887                                            ERROR   173             ;DLT AFTER READ HEADER CMD
1888                                            TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
1889                                            JMP     $EOP            ;ABORT DRIVE
1890                                    E:
1891                                    .IF     B       C
1892                                            F.EAB   0
1893                                            CHECK   301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
1894                                    .ENDC
1895
1896                                    .ENDM   RDHDR
1897
1898                                    ;
1899                                    ;A=TOCYL/FRCYL
1900                                    ;B=310 FOR TOCYL/311 FOR FRCYL
1901                                    ;
1902
1903                                    .MACRO  HDCHK3  A,B,?C
1904
1905                                            RDHDR   RDHEAD,X
1906                                            CMP     RHTAB,A ;CHECK WORD 0 (CYL#) ONLY
1907                                            BEQ     C               ;BR IF SAME
1908                                            ERROR   B               ;READ CYL WORD HEADER ERROR
1909                                    C:
1910
1911                                    .ENDM   HDCHK3
1912
1913                                    .MACRO  RALLHD  ?A,?B,?C,?D,?E
1914
```

```
                        MOV     #RHTAB,R0

A:                      MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
                        MOV     T500,TEMP1      ;SETUP  TIMEOUT
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
                        BIT     #CERR,HCS1
                        BEQ     B
                        ERROR   174             ;CERR AFTER READ HEADER  CMD
                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE

B:                      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
                        MOV     RKDB(R5),(R0)+  ;2'ND WORD
                        MOV     RKDB(R5),(R0)+  ;3'RD WORD

                        BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
                        BEQ     C
                        JSR     PC,GSTAT
                        ERROR   173             ;DATA LATE ON READ HEADER
                        TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE

C:                      CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
                        BNE     A               ;BR IF NO

                        JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
                                                ;BEGINS WITH SECTOR 0
                        CLR     WDCNT           ;WORD COUNT
                        MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
                        MOV     #HDTAB,R1       ;CALC HEADER TABLE

D:                      MOV     (R0)+,HDWD
                        MOV     (R1)+,TEMP1
                        CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
                        BEQ     E               ;BR IF COMPARE
                        ERROR   202             ;READ HEADER MISMATCH

E:                      INC     WDCNT
                        CMP     WDCNT,#66.      ;ALL WORDS DONE?
                        BNE     D               ;BR IF NO

.ENDM   RALLHD


;
; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
;
.MACRO  HDTBL   A,B,C

                        MOV     A,CALADD        ;SETUP
                        MOV     #B,HEAD         ;TO FILL
                        MOV     #C,FORMAT       ;HEADER
                        JSR     PC,FHDTAB       ;TABLE
```

```
1971                                    .ENDM   MDTBL
1972                                    ;
1973                                    ;USE FSECA FS022,RDSEC,22 FOR 22 SECTOR FORMAT
1974                                    ;USE FSECA FS020,R20SEC,20 FOR 20 SECTOR FORMAT.
1975                                    ;
1976                                    .MACRO  FSECA   A,B,C
1977
1978                                    ;
1979                                    ;FIND SECTOR 0 IN C SECTOR FORMAT.
1980                                    ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
1981                                    ;
1982                                    A:      MOV     TEMP1,-(SP)     ;SAVE TEMP1
1983                                            MOV     T5000,TEMP1     ;SETUP TIMEOUT
1984                                    1$:     JSR     PC,B            ;READ SECTOR
1985                                            TST     SECTOR          ;LOOK FOR SECTOR 0
1986                                            BNE     2$
1987                                            JSR     PC,B
1988                                            TST     SECTOR
1989                                            BEQ     3$              ;BR IF SAME TWICE
1990                                    2$:     DEC     TEMP1
1991                                            BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
1992                                            MOV     (SP)+,TEMP1     ;ELSE RESTORE TEMP1
1993                                            RTS     PC              ;EXIT
1994                                    3$:     MOV     (SP)+,TEMP1
1995                                            ADD     #2,(SP)         ;SKIP OVER ERROR
1996                                            RTS     PC
1997                                    .ENDM
1998                                    ;
1999                                    ;USE FSECB FNS22,RDSEC,22 FOR 22 SECTOR FORMAT
2000                                    ;USE FSECB FNS20,R20SEC,20 FOR 20 SECTOR FORMAT
2001                                    ;
2002                                    .MACRO  FSECB   A,B,C
2003
2004                                    ;
2005                                    ;FIND NEXT SECTOR IN C SECTOR FORMAT
2006                                    ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
2007                                    ;
2008                                    A:      MOV     TEMP1,-(SP)     ;SAVE TEMP 1
2009                                            MOV     T500,TEMP1      ;SETUP TIMEOUT
2010                                    1$:     JSR     PC,B            ;READ SECTOR
2011                                            CMP     PSEC,SECTOR
2012                                            BEQ     3$              ;BR IF SAME
2013                                            JSR     PC,B            ;ELSE TRY READ DIFFERENT TWICE
2014                                            CMP     PSEC,SECTOR
2015                                            BNE     2$              ;BR IF DIFFERENT TWICE
2016                                    3$:     DEC     TEMP1           ;ELSE TRY AGAIN IF TIME LEFT
2017                                            BNE     1$
2018                                            MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
2019                                            RTS     PC
2020                                    2$:     MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
2021                                            ADD     #2,(SP)         ;SKIP OVER ERROR
2022                                            RTS     PC
2023                                    .ENDM
2024
2025                                    ;
2026                                    ;USE SECTST FS022,FNS22,RDSEC FOR 22 SECTOR FORMAT
```

```
                        ;USE SECTST FS020,FNS20,R20SEC FOR 20 SECTOR FORMAT
                        ;
                        .MACRO  SECTST  D,E,F?A,?B,?C

                                JSR     PC,D            ;FIND SECTOR 0
                                ERROR   142             ;SECTOR 0 NOT FOUND BY TIMEOUT

                                CLR PSEC                ;PREVIOUS SECTOR
                        A:      JSR     PC,E            ;FIND NEXT SECTOR
                                ERROR   143             ;DIFFERENT SECTOR  NOT FOUND BY TIMEOUT
                                MOV     PSEC,ESEC
                                ADD     #1,ESEC         ;SETUP EXPECTED SECTOR
                                MOV     SECTOR,PSEC     ;UPDATE PREV SECTOR
                                JSR     PC,F            ;READ SECTOR
                                CMP     SECTOR,PSEC
                                BEQ     B               ;BR IF READ SAME TWICE
                                JSR     PC,F
                                CMP     SECTOR,PSEC
                                BEQ     B               ;TRY 1 MORE TIME
                                ERROR   144             ;MSG B3 ERROR, SECTOR REG UNSTABLE
                                                        ;MAY BE DURING SECTOR PULSE TIME
                        B:      CMP     SECTOR,ESEC
                                BEQ     C
                                ERROR   145             ;MSG B3 ERROR BETWEEN SECTOR COUNTS
                        C:      DEC     SECNT
                                BNE     A               ;BR IF SECTOR COUNT NOT DONE

                        .ENDM   SECTST


                        ;
                        ;DETECT OUTER LIMIT: 1.0,D.REV,OUTER
                        ;DETECT INNER LIMIT: 409.,410.,D.FWD,INNER
                        ;
                        .MACRO  LIMIT A,B,C,D

                                JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
                                ERROR   24              ;CERR AFTER SCLR

                                CLR     LPFLG
                                INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
                                INC     UNLD            ;USED FOR VALID HALT

                                MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
                                MOV     #A,RKDC(R5)     ;CYL A
                                MOV     #SEEK,RKCS1(R5) ;SEEK CMD
                                MOV     T10,TEMP1
                                JSR     PC,FRDY         ;FIND RDY
                                ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
                                JSR     PC,TSTATN       ;TEST FOR ATTN
                                ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
                                MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
                                MOV     #<D.FLT!D.PAR>,E.B0
                                MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
                                MOV     #1,E.B1
                                CHECK   110,111,146,147,<AFTER SEEK WITH BAD PARITY>,0,0,0
```

```
                        DRCLR

                        MOV     #B,RKDC(R5)         ;CYL B
                        MOV     #SEEK,RKCS1(R5)     ;SEEK TO CYL B
                        MOV     T10,TEMP1
                        JSR     PC,FRDY             ;FIND RDY
                        ERROR   131                 ;NO RDY AFTER SEEK CMD
                        MOV     #CCLR,RKCS1(R5)
                        JSR     PC,GSTAT
                        JSR     PC,FLIM             ;FIND LIMIT DETECT
                        ERROR   160                 ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT

                        BIT     #D.UNLD,HMR2
                        BNE     1$
                        ERROR   305                 ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
                        JMP     30$                 ;BYPASS REST OF TEST

                1$:     MOV     #20$,$ESCAPE        ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
                        MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
                        MOV     #<D.SKI!D.FLT>,E.B0
                        MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
                        MOV     #<D.LIMD!D.NMOV!1>,E.B1
                        CHECK   161,162,163,164,<AFTER D LIMIT DETECT>,0,0,0

                        JSR     PC,TSTATN
                        ERROR   165                 ;NO ATTN AFTER D LIMIT DETECT
                        CLR     BYPCERR             ;ALLOW CHECKING CERR IN GSTAT1

                        JSR     PC,SUBCLR           ;SUBSYS CLR
                        ERROR   24                  ;CERR AFTER SCLR
                        MOV     T10,TEMP2           ;SET UP TIMEOUT
                        JSR     PC,FHDHM            ;FIND HEAD HOME
                        ERROR   166                 ;HEAD HOME NOT FOUND BEFORE TIMEOUT
                        JSR     PC,FLOAD            ;FIND LOAD HEADS
                        ERROR   167                 ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
                        MOV     T100,TEMP2          ;SETUP TIMEOUT
                        JSR     PC,FATT1            ;FIND ATTN
                        ERROR   67                  ;ATTN NOT FOUND BEFORE TIMEOUT
                2$:     CLR     $ESCAPE
                        CLR     UNLD                ;CLEAR FLAG
                        F.EAB   D.DSC
                        CHECK   63,64,65,66,<AT END OF HEAD LOADING>,T.A2,T.B2,0
                        CWD2    175,176,<AT END OF HEAD LOADING>
                        DRCLR
                        SW814


                .ENDM   LIMIT


                ;
                ; A=CYL#, B=HEAD#
                ;
                .MACRO  HEADER  A,B

                        NEWTST <<WRITE & READ HEADERS CYL A, HEAD B>>,1
```

```
2139                            MOV     #STACK,SP       ;RESTORE STK PTR
2140
2141                            JSR     PC,SUBCLR
2142                            ERROR   24              ;CERR AFTER SCLR
2143
2144                            INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
2145
2146                            MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
2147                            MOV     #-66.,RKWC(R5)  ;WORD COUNT.
2148                            MOV     #A,TOCYL
2149                            HDTBL   TOCYL,0,0
2150                            MOV     #A,RKDC(R5)     ;CYL#
2151                            WRHDR   WRHEAD
2152                            CLR     SECNT           ;SECTOR COUNT
2153                            LOOP
2154                            MOV     #A,RKDC(R5)     ;CYL #
2155                            RALLHD
2156
2157                            CLR BYPFMT              ;ALLOW CORRECT FORMATTING
2158
2159                    .ENDM HEADER
2160
2161
2162
2163                    ;
2164                    ;SEEK TO MAJOR CYL: 0,1,TEMP3,TEMP4,D.FWD,D.REV,ASL,400,DEC,0
2165                    ;SEEK 0 TO ALL CYL: 0,1,TEMP3,TEMP4,D.FWD,D.REV,INC,410.,DEC,0
2166                    ;SEEK 410 TO ALL CYL: 410.,409.,TEMP4,TEMP3,D.REV,D.FWD,DEC,0,INC,410.
2167                    ;
2168                    .MACRO  SKOSC   A,B,C,D,E,F,G,H,I,J
2169
2170                            MOV     #A,FRCYL        ;SETUP FROM CYL
2171                            MOV     #B,TOCYL        ;SETUP TO CYL
2172
2173                    1$:     LOOP
2174                            MOV     #10$,$ESCAPE
2175                            MOV     FRCYL,TEMP3     ;SETUP
2176                            MOV     TOCYL,TEMP4     ;CYL DIFF
2177                            SUB     C,D     ;FOR
2178                            MOV     D,CALDIF        ;ERROR PRINTOUT
2179
2180                            MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
2181                            SKRDY   E
2182                    2$:     MOV     #12$,$ESCAPE
2183                            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
2184                            SKATN
2185                            CMP     CYLADD,TOCYL
2186                            BEQ     3$
2187                            ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
2188
2189                    3$:     LOOP
2190                            CLR     $ESCAPE
2191                            MOV     TOCYL,RKDC(R5)  ;CYL #
2192                            HDCHK3  TOCYL,310
2193
2194                            LOOP
```

```
                        MOV     #14$,$ESCAPE
                        MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
                        MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROUTINE
                        SKRDY   F

                4$:     MOV     #16$,$ESCAPE
                        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
                        SKATN
                        CMP     CYLADD,FRCYL
                        BEQ     5$
                        ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC

                5$:     LOOP
                        CLR     $ESCAPE
                        MOV     FRCYL,RKDC(R5)  ;CYL #
                        HDCHK3  FRCYL,311

                        CMP     TOCYL,#H        ;ALL CYL DONE?
                        BEQ     6$              ;BR IF YES
                        G       TOCYL           ;ELSE DO ANOTHER
                        JMP     1$
                6$:     SW814
                8$:     QKRPSK  I,J
                10$:    TSTSW9  8$,2$
                12$:    TSTSW9  8$,3$
                14$:    TSTSW9  8$,4$
                16$:    TSTSW9  8$,5$
                .ENDM   SKOSC

                .MACRO  EOPGM

                        SCOPE
                        MOV     #1,$TIMES
                        MOV     #$TACK,SP
                        INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
                        CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
                        BEQ     $EOP1+2         ;BR IF YES
                        JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
                $EOP1:  SCOPE
                .ENDM   EOPGM

                        .NLIST  MD
```

```
                                .SBTTL  COMMON TAGS

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

                                        .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                          .WORD   0
001102  000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                          .WORD   0
001134  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                          .WORD   0
001140  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560                  $TKS:   177560                  ;;TTY KBD STATUS
001146  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000                     $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001164  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001166  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001170  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001172  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001174  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001176  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001200  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001204  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
001205  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001206  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001210                          $MAIL:                          ;;APT MAILBOX
001210  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
```

```
2293  001212  000000                $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
2294  001214  000000                $TESTN: .WORD   ATESTN  ;;TEST NUMBER
2295  001216  000000                $PASS:  .WORD   APASS   ;;PASS COUNT
2296  001220  000000                $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
2297  001222  000000                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
2298  001224  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
2299  001226  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
2300  001230                        $ETABLE:                ;;APT ENVIRONMENT TABLE
2301  001230      000               $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
2302  001231      000               $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
2303  001232  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
2304  001234  000000                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
2305  001236  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
2306                                ;*                      BITS 15-11=CPU TYPE
2307                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2308                                ;*                              11/70=06,PDQ=07,Q=10
2309                                ;*                      BIT 10=REAL TIME CLOCK
2310                                ;*                      BIT  9=FLOATING POINT PROCESSOR
2311                                ;*                      BIT  8=MEMORY MANAGEMENT
2312  001240      000               $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
2313  001241      000               $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
2314                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
2315                                ;*                              900 NSEC CORE=001
2316                                ;*                              300 NSEC BIPOLAR=002
2317                                ;*                              500 NSEC MOS=003
2318  001242  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
2319                                ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2320  001244      000               $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
2321  001245      000               $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
2322  001246  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
2323  001250      000               $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
2324  001251      000               $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
2325  001252  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
2326  001254      000               $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
2327  001255      000               $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
2328  001256  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
2329  001260  000000                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
2330  001262  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
2331  001264  177440                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
2332  001266  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
2333  001270  000000                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
2334  001272  000000                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
2335  001274  000000                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
2336  001276  000000                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
2337  001300  000000                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
2338  001302  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
2339  001304  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
2340  001306  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
2341  001310  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
2342  001312  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
2343  001314                        $ETEND:
2344                                .MEXIT
2345          177440                ABASE=  177440                  ;DEFAULT BUSS ADDRESS
2346  001314  000210                RKVEC:  210                     ;DEFAULT CONTROLLER INTERRUPT VECTOR
2347  001316  000240                RKPRI:  PR5                     ;PRIORITY
2348  001320  172540                PKS:    172540                  ;P-CLOCK STATUS REG
```

```
2349  001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
2350  001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
2351  001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.
2352
2353  001330  000100          LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
2354  001332  000104          PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.
2355
2356          000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
2357          172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
2358  001334  000000          TRAPPC: 0               ;PC FOR MEMORY CHECK ENABLE TRAP
2359
2360  001336  000000          PARAM:  0               ;1 FOR 220 OR 230 START, NO DEFAULT
2361  001340  000000          BYPT16: 0               ;1 FOR 210, 230, 270 START
2362  001342  000000          MODTST: 0               ;1 FOR 260 OR 270 START
2363  001344  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
2364
2365  001346  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
2366                                                  ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
2367  001350  000000          FRCYL:  0               ;FROM CYL
2368  001352  000000          TOCYL:  0               ;TO CYL
2369  001354  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
2370  001356  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
2371  001360  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
2372  001362  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
2373  001364  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
2374  001366  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
2375
2376  001370  000074          HZ:     60.             ;60 FOR 60 CPS
2377                                                  ;50 FOR 50 CPS
2378  001372  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
2379                                                  ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
2380  001374  000000          SEC:    0               ;SECOND COUNTER
2381  001376  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
2382  001400  000000          SECNT:  0               ;SECTOR COUNT
2383  001402  000000          PSEC:   0               ;PREVIOUS SECTOR
2384  001404  000000          ESEC:   0               ;EXPECTED SECTOR
2385  001406  000000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
2386
2387  001410  000000          LPFLG:  0               ;SET TO 0 TO RETURN TO $LPADR
2388                                                  ;IF SW14 OR SW8 SET
2389                                                  ;SET TO 1 TO RETURN TO $LPERR
2390                                                  ;IF SW9 SET
2391  001412  000001          T1:     1               ;TIMEOUT CONSTANTS
2392  001414  000012          T10:    10.
2393  001416  000144          T100:   100.
2394  001420  000764          T500:   500.
2395  001422  004704          T2500:  2500.
2396  001424  011610          T5000:  5000.
2397  001426  141520          T50000: 50000.
2398
2399  001430  000000          HEAD:   0               ;HEAD NUMBER
2400  001432  000000          HEADA:  0               ;HEAD # FROM H.B3 RIGHT JUSTIFIED
2401  001434  000000          HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
2402  001436  000000          FORMAT: 0               ;FORMAT TYPE
2403  001440  000000          FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
2404  001442  000000          WDCNT:  0               ;WORD COUNT
```

```
                                
001444  000000          DATA0:  0               ;ALL 0'S
001446  052525          DATA01: 52525           ;0101 PATT
001450  177777          DATA1:  177777          ;ALL 1'S
                                
001452  000000          WORD:   0               ;HEADER/DATA WORD
001454  000000          HDWD:   0               ;HEADER WORD FROM RKDB
                                
001456  000000          BSERR:  0               ;CANNOT READ BSE INFO WHEN SET
001460  000000          LIMERR: 0               ;LIMIT DETECT ERROR FLAG
                                
001462  000000          BYPCERR:0               ;SET TO 1 TO BYPASS CKCERR IN GSTAT1 ROUTINE
001464  000000          BYPFMT: 0               ;BYPASS FORMAL FORMATTING OF HEADERS
                                                ;UNTIL BSE INFO HAS BEEN STORED.
                                                ;IF SET, BIT 14,15 = 1
                                
001466  000000          CHKFLG: 0               ;WORDS TO BE CHECKED
                                
001470  000102          HDTAB:  .BLKW 66.       ;CALCULATED HEADER WORD TABLE
001674  000102          RHTAB:  .BLKW 66.       ;FILLED AFTER READ HEADER CMD
002100  000102          SRTTAB: .BLKW 66.       ;ABOVE RHTAB SORTED STARTING FORM
                                                ;SECTOR 0 BY SORT ROUTINE
002304  000400          BSE22H: .BLKW 256.      ;22 SECTOR HARDWARE BSE INFO.
                                                ;22 SECTOR SOFTWARE BSE INFO
                                                ;OVERLAYS MSG1
                                
003304  000633          INVCYL: 411.            ;INVALID CYL ADDR
003306  000634                  412.
003310  000640                  416.
003312  000700                  448.
003314  000740                  480.
                                
003316  000000          UNLD:   0               ;SET TO 0 IF HEADS ARE LOADED
                                                ;SET TO 1 IF HEADS UNLOADED
003320  000000          BADHDR: 0               ;SET TO 0 IF FORMATTING OK
                                                ;SET TO 1 IF FORMATTING ALTERED
003322  000000          HPEND:  0               ;SET TO 0 IF HALT NOT PENDING
                                                ;SET TO 1 IF HALT PENDING
                                
                                                ;THE ABOVE 3 FLAGS ARE USED
                                                ;BY 'STOP' ROUTINE TO BRING
                                                ;THE CPU TO A VALID HALT.
                                
003324  001 002 004 ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
003327  010 020 040
003332  100 200
                                        .EVEN
                                
                                
                                ;
                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
                                ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
                                ;
                                
003334  000000          HCS1:   0               ;HOLD RKCS1
003336  000000          HCS2:   0               ;HOLD RKCS2
```

```
2461   003340   000000                HWC:    0                       ;HOLD RKWC
2462   003342   000000                HBA:    0                       ;ETC.
2463   003344   000000                HDA:    0
2464   003346   000000                HDS:    0
2465   003350   000000                HER:    0
2466   003352   000000                HASOF:  0
2467   003354   000000                HDC:    0
2468   003356   000000                HDB:    0
2469   003360   000000                HMR1:   0
2470   003362   000000                HMR2:   0
2471   003364   000000                HMR3:   0
2472   003366   000000                HPOS:   0
2473   003370   000000                HPAT:   0
2474
2475   003372   000000                TEMP1:  0                       ;TEMPORARY STORAGE.
2476   003374   000000                TEMP2:  0
2477   003376   000000                TEMP3:  0
2478   003400   000000                TEMP4:  0
2479   003402   000000                TEMP5:  0
2480                                  ;
2481                                  ; THE FOLLOWING ARE HOLDING REGISTERS FOR MSG A (0-3) & MSG B (0-3)
2482                                  ;
2483   003404   000000                H.A0:   0
2484   003406   000000                H.B0:   0
2485   003410   000000                H.A1:   0
2486   003412   000000                H.B1:   0
2487   003414   000000                H.A2:   0
2488   003416   000000                H.B2:   0
2489   003420   000000                H.A3:   0
2490   003422   000000                H.B3:   0
2491                                  ;
2492                                  ; THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE
2493                                  ;
2494   003424   000000                E.A0:   0
2495   003426   000000                E.B0:   0
2496   003430   000000                E.A1:   0
2497   003432   000000                E.B1:   0
2498   003434   000000                E.A2:   0
2499   003436   000000                E.B2:   0
2500   003440   000000                E.A3:   0
2501   003442   000000                E.B3:   0
2502
2503                                  ;
2504                                  ; THE FOLLOWING ARE IDENTITIES FOR DRIVE MSG WORDS TO BE TESTED
2505                                  ;
2506            000001                T.A2=BIT0                       ;TEST MSG A2 IF SET
2507            000002                T.B2=BIT1
2508            000004                T.B3=BIT2
2509
2510                                  ;
2511                                  ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
2512                                  ;
2513
2514   003444   000000                DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
2515   003446   000000                DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
2516   003450   000000                ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
```

```
2517  003452  000000                PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
2518  003454  000000                DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
2519
2520                                ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
2521                                ;IS PRESENT AND IS TO BE TESTED.
2522
2523  003456  000000                DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
2524  003460  000000                DRIV1:  0                       ;FOR DRIVE 1
2525  003462  000000                DRIV2:  0                       ;FOR DRIVE 2
2526  003464  000000                DRIV3:  0                       ;FOR DRIVE 3
2527  003466  000000                DRIV4:  0                       ;FOR DRIVE 4
2528  003470  000000                DRIV5:  0                       ;FOR DRIVE 5
2529  003472  000000                DRIV6:  0                       ;FOR DRIVE 6
2530  003474  000000                DRIV7:  0                       ;FOR DRIVE 7
2531
2532  003476  000000                LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
2533  003500  000000                PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
2534  003502  000000                DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
2535  003504  000000                SIZFLG: 0                       ;SET IF DEFAULT DO SIZING IN TEST 1
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


003506                          $ERRTB:

                                ;ERROR 1
003506  057054                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
003510  063736                          DH1
003512  066044                          DT1
003514  066712                          DF1

                                ;ERROR 2
003516  057324                          EM5                     ;DETECTED MDS
003520  063736                          DH1
003522  066044                          DT1
003524  066712                          DF1

                                ;ERROR 3
003526  057345                          EM6                     ;DETECTED UFE
003530  063736                          DH1
003532  066044                          DT1
003534  066712                          DF1

                                ;ERROR 4
003536  057366                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
003540  063736                          DH1
003542  066044                          DT1
003544  066712                          DF1
                                ;ERROR 5
003546  057455                          EM8                     ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
003550  063736                          DH1
003552  066044                          DT1
003554  066712                          DF1

                                ;ERROR 6
003556  057525                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
003560  063736                          DH1
003562  066044                          DT1
003564  066712                          DF1

                                ;ERROR 7
003566  057575                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
003570  063736                          DH1
003572  066044                          DT1
003574  066712                          DF1
```

```
                                ;ERROR 10
003576  057640                          EM11            ;DRA & NED BOTH SET
003600  063736                          DH1
003602  066044                          DT1
003604  066712                          DF1

                                ;ERROR 11
003606  057704                          EM12            ;CONTROLLER NOT READY
003610  064527                          DH18            ;AFTER UNLOAD CMD.
003612  066044                          DT1
003614  067036                          DF10

                                ;ERROR 12
003616  057735                          EM13            ;NO ATTN
003620  064527                          DH18            ;AFTER UNLOAD CMD
003622  066044                          DT1
003624  067036                          DF10
                                ;ERROR 13
003626  057757                          EM14            ;WRONG ATTN
003630  064527                          DH18
003632  066044                          DT1
003634  067036                          DF10
                                ;ERROR 14
003636  060004                          EM15            ;DRDY NOT CLEARED
003640  064527                          DH18
003642  066044                          DT1
003644  067036                          DF10
                                ;ERROR 15
003646  060036                          EM16            ;DSC NOT SET
003650  064527                          DH18
003652  066044                          DT1
003654  067036                          DF10
                                ;ERROR 16
003656  060063                          EM17            ;MSG A0 ERROR
003660  064127                          DH8             ;IN UNLD
003662  066446                          DT13
003664  067206                          DF20
                                ;ERROR 17
003666  060100                          EM18            ;MSG B0 ERROR
003670  064127                          DH8             ;IN UNLD
003672  066446                          DT13
003674  067206                          DF20
                                ;ERROR 20
003676  060115                          EM19            ;MSG A1 ERROR
003700  064127                          DH8             ;IN UNLD
003702  066446                          DT13
003704  067206                          DF20
                                ;ERROR 21
003706  060132                          EM20            ;MSG B1 ERROR
003710  064127                          DH8             ;IN UNLD
003712  066446                          DT13
003714  067206                          DF20
                                ;ERROR 22
003716  061464                          EM46            ;MSG A2 ERROR
003720  064127                          DH8             ;IN UNLD
```

```
2648  003722  066526                  DT14
2649  003724  067262                  DF22
2650                          ;ERROR 23
2651  003726  061477                  EM47          ;MSG B2 ERROR
2652  003730  064127                  DH8           ;IN UNLD
2653  003732  066526                  DT14
2654  003734  067262                  DF22
2655
2656                          ;ERROR 24
2657  003736  060147                  EM21          ;CERR SET
2658  003740  064616                  DH21          ;AFTER SCLR
2659  003742  066044                  DT1
2660  003744  067036                  DF10
2661                          ;ERROR 25
2662  003746  060171                  EM22          ;RLS DID NOT SET CERR
2663  003750  063736                  DH1
2664  003752  066044                  DT1
2665  003754  066712                  DF1
2666
2667                          ;ERROR 26
2668  003756  060230                  EM23          ;SACK SET AFTER RLS SENT
2669  003760  063736                  DH1
2670  003762  066044                  DT1
2671  003764  066712                  DF1
2672
2673                          ;ERROR 27
2674  003766  060310                  EM24          ;VOL VALID NOT SET
2675  003770  064550                  DH19          ;AFTER PACK CMD
2676  003772  066044                  DT1
2677  003774  067036                  DF10
2678                          ;ERROR 30
2679  003776  060334                  EM25          ;DRIVE TYPE SET IN MR2
2680  004000  063736                  DH1
2681  004002  066044                  DT1
2682  004004  066712                  DF1
2683                          ;ERROR 31
2684  004006  060362                  EM26          ;DDT SET IN RKDS
2685  004010  063736                  DH1
2686  004012  066044                  DT1
2687  004014  066712                  DF1
2688                          ;ERROR  32
2689  004016  060402                  EM27          ;DTYE SET IN RKER
2690  004020  063736                  DH1
2691  004022  066044                  DT1
2692  004024  066712                  DF1
2693                          ;ERROR 33
2694  004026  060423                  EM28          ;DTYE NOT SET IN RKER
2695  004030  063736                  DH1
2696  004032  066044                  DT1
2697  004034  066712                  DF1
2698                          ;ERROR 34
2699  004036  060450                  EM29          ;DTYE DID NOT SET CERR
2700  004040  063736                  DH1
2701  004042  066044                  DT1
2702  004044  066712                  DF1
2703                          ;ERROR 35
```

```
2704  004046  060517                EM30            ;C-D PARITY ERROR SET IN MR3
2705  004050  063736                DH1
2706  004052  066044                DT1
2707  004054  066712                DF1
2708                          ;ERROR 36
2709  004056  060550                EM31            ;D-C PARITY  SET IN CS1
2710  004060  063736                DH1
2711  004062  066044                DT1
2712  004064  066712                DF1
2713                          ;ERROR 37
2714  004066  060575                EM32            ;FAULT NOT SET IN MR3
2715  004070  063736                DH1
2716  004072  066044                DT1
2717  004074  066712                DF1
2718                          ;ERROR 40
2719  004076  060622                EM33            ;C-D PARITY ERROR NOT SET IN MR3
2720  004100  063736                DH1
2721  004102  066044                DT1
2722  004104  066712                DF1
2723                          ;ERROR 41
2724  004106  060657                EM34            ;D-C PARITY  NOT SET IN CS1
2725  004110  063736                DH1
2726  004112  066044                DT1
2727  004114  066712                DF1
2728                          ;ERROR 42
2729  004116  060710                EM35            ;DCPAR DID NOT SET CERR
2730  004120  063736                DH1
2731  004122  066044                DT1
2732  004124  066712                DF1
2733                          ;ERROR 43
2734  004126  060763                EM36            ;CYL ADDR IN B2 NOT = RKDC
2735  004130  064415                DH14            ;AFTER SEEK WITH BAD PARITY
2736  004132  066526                DT14
2737  004134  067262                DF22
2738                          ;ERROR 44
2739  004136  061023                EM37            ;CYL DIFF IN A2 NOT=RKDC
2740  004140  064415                DH14
2741  004142  066526                DT14
2742  004144  067262                DF22
2743                          ;ERROR 45
2744  004146  060763                EM36            ;CYL ADDR IN RKMR3 NOT=RKDC
2745  004150  064415                DH14
2746  004152  066104                DT4
2747  004154  067012                DF6
2748                          ;ERROR 46
2749  004156  061063                EM38            ;CYL DIFF IN RKMR2 NOT=CALDIF
2750  004160  064415                DH14
2751  004162  066104                DT4
2752  004164  067012                DF6
2753                          ;ERROR 47
2754  004166  061134                EM39            ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2755  004170  064507                DH17            ;AFTER RECAL CMD
2756  004172  066526                DT14
2757  004174  067262                DF22
2758                          ;ERROR 50
2759  004176  061171                EM40            ;CYL ADDR IN RKMR3 NOT CLEARED
```

ERROR POINTER TABLE

```
2760  004200  064507                    DH17                    ;AFTER RECAL COMD
2761  004202  066526                    DT14
2762  004204  067262                    DF22
2763                            ;ERROR 51
2764  004206  060063                    EM17                    ;A0 ERROR
2765  004210  064732                    DH26                    ;AFTER READ DATA CMD
2766  004212  066446                    DT13
2767  004214  067206                    DF20
2768                            ;ERROR 52
2769  004216  060100                    EM18                    ;B0 ERROR
2770  004220  064732                    DH26
2771  004222  066446                    DT13
2772  004224  067206                    DF20
2773                            ;ERROR 53
2774  004226  061260                    EM43                    ;HEAD DECODE IN B3 NOT CLEARED
2775  004230  064507                    DH17                    ;AFTER RECAL CMD
2776  004232  066616                    DT15
2777  004234  067316                    DF23
2778                            ;ERROR 54
2779  004236  061307                    EM44                    ;B3 HEAD DECODE INCORRECT
2780  004240  064445                    DH16
2781  004242  066616                    DT15
2782  004244  067316                    DF23
2783                            ;ERROR 55
2784  004246  057735                    EM13                    ;NO ATTN
2785  004250  064507                    DH17                    ;AFTER RECAL CMD
2786  004252  066044                    DT1
2787  004254  067036                    DF10
2788                            ;ERROR 56
2789  004256  062431                    EM64                    ;MSG B3 HEAD REG NOT CLEARED
2790  004260  064127                    DH8                     ;IN UNLOAD
2791  004262  066616                    DT15
2792  004264  067316                    DF23
2793                            ;ERROR 57
2794  004266  060063                    EM17                    ;MSG A0 ERROR
2795  004270  064153                    DH9                     ;AFTER START SPINDLE CMD REC'D BY DRIVE
2796  004272  066446                    DT13
2797  004274  067206                    DF20
2798                            ;ERROR 60
2799  004276  060100                    EM18                    ;MSG B0 ERROR
2800  004300  064153                    DH9
2801  004302  066446                    DT13
2802  004304  067206                    DF20
2803                            ;ERROR 61
2804  004306  060115                    EM19                    ;MSG A1 ERROR
2805  004310  064153                    DH9
2806  004312  066446                    DT13
2807  004314  067206                    DF20
2808                            ;ERROR 62
2809  004316  060132                    EM20                    ;MSG B1 ERROR
2810  004320  064153                    DH9
2811  004322  066446                    DT13
2812  004324  067206                    DF20
2813                            ;ERROR 63
2814  004326  060063                    EM17
2815  004330  064217                    DH10                    ;AT END OF HEAD LOADING
```

```
2816  004332  066446                    DT13
2817  004334  067206                    DF20
2818                              ;ERROR 64
2819  004336  060100                    EM18
2820  004340  064217                    DH10
2821  004342  066446                    DT13
2822  004344  067206                    DF20
2823                              ;ERROR 65
2824  004346  060115                    EM19
2825  004350  064217                    DH10
2826  004352  066446                    DT13
2827  004354  067206                    DF20
2828                              ;ERROR 66
2829  004356  060132                    EM20
2830  004360  064217                    DH10
2831  004362  066446                    DT13
2832  004364  067206                    DF20
2833                              ;ERROR 67
2834  004366  057735                    EM13          ;NO ATTN
2835  004370  064217                    DH10          ;AT END OF HEAD LOADING.
2836  004372  066044                    DT1
2837  004374  067036                    DF10
2838                              ;ERROR 70
2839  004376  061613                    EM50          ;FWD NOT SET WITHIN 60 SEC FROM
2840  004400  063736                    DH1           ;START SPINDLE CMD.
2841  004402  066044                    DT1
2842  004404  066712                    DF1
2843                              ;ERROR 71
2844  004406  060063                    EM17
2845  004410  064246                    DH11          ;AFTER START SPINDLE CMD & FWD SET.
2846  004412  066446                    DT13
2847  004414  067206                    DF20
2848                              ;ERROR 72
2849  004416  060100                    EM18
2850  004420  064246                    DH11
2851  004422  066446                    DT13
2852  004424  067206                    DF20
2853                              ;ERROR 73
2854  004426  060115                    EM19
2855  004430  064246                    DH11
2856  004432  066446                    DT13
2857  004434  067206                    DF20
2858                              ;ERROR 74
2859  004436  060132                    EM20
2860  004440  064246                    DH11
2861  004442  066446                    DT13
2862  004444  067206                    DF20
2863                              ;ERROR 75
2864  004446  061664                    EM51          ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
2865  004450  063736                    DH1           ;FROM START SPINDLE CMD.
2866  004452  066044                    DT1
2867  004454  066712                    DF1
2868                              ;ERROR 76
2869  004456  060063                    EM17
2870  004460  064305                    DH12          ;AT INNER LIMIT FROM START SPINDLE CMD.
2871  004462  066446                    DT13
```

```
2872   004464   067206                          DF20
2873                                    ;ERROR 77
2874   004466   060100                          EM18
2875   004470   064305                          DH12
2876   004472   066446                          DT13
2877   004474   067206                          DF20
2878                                    ;ERROR 100
2879   004476   060115                          EM19
2880   004500   064305                          DH12
2881   004502   066446                          DT13
2882   004504   067206                          DF20
2883                                    ;ERROR 101
2884   004506   060132                          EM20
2885   004510   064305                          DH12
2886   004512   066446                          DT13
2887   004514   067206                          DF20
2888                                    ;ERROR 102
2889   004516   061525                          EM49        ;FWD NOT SET WITHIN 4 SEC IN RTZ PORTION
2890   004520   063736                          DH1         ;OF START SPIN CMD.
2891   004522   066044                          DT1
2892   004524   066712                          DF1
2893                                    ;ERROR 103
2894   004526   060063                          EM17
2895   004530   064346                          DH13        ;FROM OUTER LIMIT TO CYL 0 DURING LOADING
2896   004532   066446                          DT13
2897   004534   067206                          DF20
2898                                    ;ERROR 104
2899   004536   060100                          EM18
2900   004540   064346                          DH13
2901   004542   066446                          DT13
2902   004544   067206                          DF20
2903                                    ;ERROR 105
2904   004546   060115                          EM19
2905   004550   064346                          DH13
2906   004552   066446                          DT13
2907   004554   067206                          DF20
2908                                    ;ERROR 106
2909   004556   060132                          EM20
2910   004560   064346                          DH13
2911   004562   066446                          DT13
2912   004564   067206                          DF20
2913                                    ;ERROR 107
2914   004566   061343                          EM45        ;DRIVE READY NOT SET WITHIN 1 SEC
2915   004570   063736                          DH1         ;FROM FWD IN RTZ PORTION OF START SPIN CMD.
2916   004572   066044                          DT1
2917   004574   066712                          DF1
2918                                    ;ERROR 110
2919   004576   060063                          EM17        ;MSG AO ERROR
2920   004600   064415                          DH14        ;AFTER SEEK WITH BAD PARITY
2921   004602   066526                          DT14
2922   004604   067262                          DF22
2923                                    ;ERROR 111
2924   004606   060100                          EM18        ;MSG BO ERROR
2925   004610   064415                          DH14        ;AFTER SEEK WITH BAD PARITY
2926   004612   066526                          DT14
2927   004614   067262                          DF22
```

```
2928
2929  004616  060115          ;ERROR 112
2930  004620  064732                EM19                    ;A1 ERROR
2931  004622  066446                DH26                    ;AFTER READ DATA CMD
2932  004624  067206                DT13
2933                             DF20
2934  004626  060132          ;ERROR 113
2935  004630  064732                EM20                    ;B1 ERROR
2936  004632  066446                DH26
2937  004634  067206                DT13
2938                             DF20
2939  004636  060063          ;ERROR 114
2940  004640  064445                EM17
2941  004642  066446                DH16                    ;AFTER LOADING HEAD REGISTER & SEEK CMD
2942  004644  067206                DT13
2943                             DF20
2944  004646  060100          ;ERROR 115
2945  004650  064445                EM18
2946  004652  066446                DH16
2947  004654  067206                DT13
2948                             DF20
2949  004656  057704          ;ERROR 116
2950  004660  064550                EM12                    ;CONT NOT RDY
2951  004662  066044                DH19                    ;AFTER PACK CMD
2952  004664  067036                DT1
2953                             DF10
2954  004666  057704          ;ERROR 117
2955  004670  064567                EM12                    ;CONT NOT RDY
2956  004672  066044                DH20                    ;AFTER SEL DR CMD
2957  004674  067036                DT1
2958                             DF10
2959  004676  057704          ;ERROR 120
2960  004700  064616                EM12
2961  004702  066044                DH21                    ;AFTER SUBSYS CLEAR
2962  004704  067036                DT1
2963                             DF10
2964  004706  057704          ;ERROR 121
2965  004710  064153                EM12
2966  004712  066044                DH9                     ;AFTER START SPINDLE CMD
2967  004714  067036                DT1
2968                             DF10
2969  004716  057704          ;ERROR 122
2970  004720  064415                EM12
2971  004722  066044                DH14                    ;AFTER SEEK WITH BAD PARITY
2972  004724  067036                DT1
2973                             DF10
2974  004726  063542          ;ERROR 123
2975  004730  063736                EM88                    ;NO DRIVES FOUND
2976  004732  066044                DH1
2977  004734  066712                DT1
2978                             DF1
2979  004736  057704          ;ERROR 124
2980  004740  064507                EM12
2981  004742  066044                DH17                    ;AFTER RECAL CMD
2982  004744  067036                DT1
2983                             DF10
                               ;ERROR 125
```

```
 2984  004746  057735                EM13            ;NO ATTN
 2985  004750  064415                DH14            ;FROM SEEK WITH BAD PARITY
 2986  004752  066044                DT1
 2987  004754  067036                DF10
 2988                          ;ERROR 126
 2989  004756  063633                EM89            ;NO DRVS FOUND IN DEVICE MAP
 2990  004760  063736                DH1
 2991  004762  066044                DT1
 2992  004764  066712                DF1
 2993                          ;ERROR 127
 2994  004766  061134                EM39            ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
 2995  004770  064616                DH21            ;AFTER SCLR
 2996  004772  066044                DT1
 2997  004774  067036                DF10
 2998                          ;ERROR 130
 2999  004776  061171                EM40            ;CYL ADDR IN RKMR3 NOT CLEARED
 3000  005000  064616                DH21
 3001  005002  066044                DT1
 3002  005004  067036                DF10
 3003                          ;ERROR 131
 3004  005006  057704                EM12            ;NO RDY
 3005  005010  064713                DH25            ;AFTER SEEK CMD
 3006  005012  066044                DT1
 3007  005014  067036                DF10
 3008                          ;ERROR 132
 3009  005016  057735                EM13            ;NO ATTN
 3010  005020  064713                DH25
 3011  005022  066044                DT1
 3012  005024  067036                DF10
 3013                          ;ERROR 133
 3014  005026  060063                EM17            ;MSG A0 ERROR
 3015  005030  064713                DH25
 3016  005032  066446                DT13
 3017  005034  067206                DF20
 3018                          ;ERROR 134
 3019  005036  060100                EM18            ;MSG B0 ERROR
 3020  005040  064713                DH25
 3021  005042  066446                DT13
 3022  005044  067206                DF20
 3023                          ;ERROR 135
 3024  005046  060115                EM19            ;MSG A1 ERROR
 3025  005050  064713                DH25
 3026  005052  066446                DT13
 3027  005054  067206                DF20
 3028                          ;ERROR 136
 3029  005056  060132                EM20            ;MSG B1 ERROR.
 3030  005060  064713                DH25
 3031  005062  066446                DT13
 3032  005064  067206                DF20
 3033                          ;ERROR 137
 3034  005066  061134                EM39            ;CYL DIFF/OFFSET IN A2 NOT CLEARED
 3035  005070  064713                DH25
 3036  005072  066526                DT14
 3037  005074  067262                DF22
 3038                          ;ERROR 140
 3039  005076  061171                EM40            ;CYL ADDR IN B2 NOT CLEARED
```

```
3040  005100  064713                    DH25
3041  005102  066526                    DT14
3042  005104  067262                    DF22
3043                          ;ERROR 141
3044  005106  061764                    EM52          ;20 SECTOR FORMAT NOT SET IN RKMR2
3045  005110  063736                    DH1
3046  005112  066044                    DT1
3047  005114  066712                    DF1
3048                          ;ERROR 142
3049  005116  062023                    EM53          ;SECTOR 0 NOT FOUND WITHIN 50 MS
3050  005120  063736                    DH1
3051  005122  066044                    DT1
3052  005124  066712                    DF1
3053                          ;ERROR 143
3054  005126  062054                    EM54          ;DIFF SECTOR NOT FOUND WITHIN 3MS
3055  005130  063736                    DH1
3056  005132  066044                    DT1
3057  005134  066712                    DF1
3058                          ;ERROR 144
3059  005136  061512                    EM48          ;MSG B3 ERROR
3060  005140  065071                    DH34          ;SECTOR REG UNSTABLE
3061  005142  066044                    DT1
3062  005144  067036                    DF10
3063                          ;ERROR 145
3064  005146  061512                    EM48
3065  005150  065115                    DH35          ;BETWEEN SECTOR COUNTS
3066  005152  066152                    DT6
3067  005154  067056                    DF12
3068                          ;ERROR 146
3069  005156  060115                    EM19          ;MSG A1 ERROR
3070  005160  064415                    DH14          ;AFTER SEEK WITH BAD PARITY
3071  005162  066526                    DT14
3072  005164  067262                    DF22
3073                          ;ERROR 147
3074  005166  060132                    EM20          ;MSG B1 ERROR
3075  005170  064415                    DH14
3076  005172  066526                    DT14
3077  005174  067262                    DF22
3078                          ;ERROR 150
3079  005176  060115                    EM19          ;MSG A1 ERROR
3080  005200  065166                    DH37
3081  005202  066044                    DT1
3082  005204  067036                    DF10
3083                          ;ERROR 151
3084  005206  057704                    EM12          ;NO RDY
3085  005210  064644                    DH22          ;AFTER CLEAR CMD
3086  005212  066044                    DT1
3087  005214  067036                    DF10
3088                          ;ERROR 152
3089  005216  000000                    0
3090  005220  000000                    0
3091  005222  000000                    0
3092  005224  000000                    0
3093                          ;ERROR 153
3094  005226  000000                    0
3095  005230  000000                    0
```

```
 3096  005232  000000                    0
 3097  005234  000000                    0
 3098                            ;ERROR 154
 3099  005236  062107                    EM55            ;ATTN NOT CLEARED
 3100  005240  064644                    DH22
 3101  005242  066044                    DT1
 3102  005244  067036                    DF10
 3103                            ;ERROR 155
 3104  005246  063377                    EM85            ;IDAE NOT CLEARED
 3105  005250  064507                    DH17            ;AFTER RECAL CMD
 3106  005252  066044                    DT1
 3107  005254  067036                    DF10
 3108                            ;ERROR 156
 3109  005256  057704                    EM12            ;CONT NOT READY
 3110  005260  065716                    DH51            ;AFTER SEEK TO SELF
 3111  005262  066044                    DT1
 3112  005264  067036                    DF10
 3113                            ;ERROR 157
 3114  005266  057735                    EM13            ;NO ATTN
 3115  005270  065716                    DH51
 3116  005272  066044                    DT1
 3117  005274  067036                    DF10
 3118                            ;ERROR 160
 3119  005276  062267                    EM59            ;LIMIT DETECT NOT FOUND
 3120  005300  063736                    DH1
 3121  005302  066044                    DT1
 3122  005304  066712                    DF1
 3123                            ;ERROR 161
 3124  005306  060063                    EM17            ;MSG A0 ERROR
 3125  005310  065232                    DH38            ;AFTER LIMIT DETECT
 3126  005312  066446                    DT13
 3127  005314  067206                    DF20
 3128                            ;ERROR 162
 3129  005316  060100                    EM18            ;MSG B0 ERROR
 3130  005320  065232                    DH38
 3131  005322  066446                    DT13
 3132  005324  067206                    DF20
 3133                            ;ERROR 163
 3134  005326  060115                    EM19            ;MSG A1 ERROR
 3135  005330  065232                    DH38
 3136  005332  066446                    DT13
 3137  005334  067206                    DF20
 3138                            ;ERROR 164
 3139  005336  060132                    EM20            ;MSG B1 ERROR
 3140  005340  065232                    DH38
 3141  005342  066446                    DT13
 3142  005344  067206                    DF20
 3143                            ;ERROR 165
 3144  005346  057735                    EM13            ;NO ATTN
 3145  005350  065232                    DH38
 3146  005352  066044                    DT1
 3147  005354  067036                    DF10
 3148                            ;ERROR 166
 3149  005356  062320                    EM60            ;HEADS HOME NOT FOUND
 3150  005360  065232                    DH38
 3151  005362  066044                    DT1
```

```
3152  005364  067036                        DF10
3153                          ;ERROR 167
3154  005366  062354                        EM61          ;LOAD HEADS NOT FOUND
3155  005370  065232                        DH38
3156  005372  066044                        DT1
3157  005374  067036                        DF10
3158                          ;ERROR 170
3159  005376  057211                        EM4           ;FATAL ERROR
3160  005400  065464                        DH45          ;LIMIT DETECT ERROR ON PREVIOUS TEST
3161  005402  066044                        DT1
3162  005404  067142                        DF16
3163                          ;ERROR 171
3164  005406  057704                        EM12          ;NO RDY
3165  005410  064776                        DH30          ;AFTER READ HEADER CMD
3166  005412  066044                        DT1
3167  005414  067036                        DF10
3168                          ;ERROR 172
3169  005416  061134                        EM39          ;CYL DIFF/OFFSET NOT CLEARED
3170  005420  064776                        DH30          ;AFTER READ HEADER CMD
3171  005422  066526                        DT14
3172  005424  067262                        DF22
3173                          ;ERROR 173
3174  005426  062410                        EM63          ;DLT SET
3175  005430  064776                        DH30
3176  005432  066044                        DT1
3177  005434  067122                        DF15
3178                          ;ERROR 174
3179  005436  060147                        EM21          ;CERR SET
3180  005440  064776                        DH30
3181  005442  066044                        DT1
3182  005444  067122                        DF15
3183                          ;ERROR 175
3184  005446  061134                        EM39          ;CYL DIFF NOT CLEARED
3185  005450  064217                        DH10          ;AT END OF HEAD LOADING
3186  005452  066526                        DT14
3187  005454  067262                        DF22
3188                          ;ERROR 176
3189  005456  061171                        EM40          ;CYL ADDR NOT CLEARED.
3190  005460  064217                        DH10
3191  005462  066526                        DT14
3192  005464  067262                        DF22
3193                          ;ERROR 177
3194  005466  062706                        EM72          ;FORMAT TEST BYPASSED
3195  005470  065543                        DH46          ;COULD NOT READ BSE INFO
3196  005472  066044                        DT1
3197  005474  067142                        DF16
3198                          ;ERROR 200
3199  005476  057704                        EM12          ;NO RDY
3200  005500  065250                        DH39          ;AFTER WRITE HEADER CMD
3201  005502  066044                        DT1
3202  005504  067122                        DF15
3203                          ;ERROR 201
3204  005506  060147                        EM21          ;CERR SET
3205  005510  065250                        DH39
3206  005512  066044                        DT1
3207  005514  067122                        DF15
```

ERROR POINTER TABLE

```
3208
3209  005516  062460          ;ERROR 202
3210  005520  063736                EM65                    ;READ HEADER ERROR
3211  005522  066216                DH1
3212  005524  067102                DT7
3213                                 DF14
3214  005526  060063          ;ERROR 203
3215  005530  065051                EM17                    ;MSG AQ ERROR
3216  005532  066446                DH33                    ;DURING SEEK CMD
3217  005534  067206                DT13
3218                                 DF20
3219  005536  060100          ;ERROR 204
3220  005540  065051                EM18                    ;MSG BQ ERROR
3221  005542  066446                DH33
3222  005544  067206                DT13
3223                                 DF20
3224  005546  060115          ;ERROR 205
3225  005550  065051                EM19                    ;MSG A1 ERROR
3226  005552  066446                DH33
3227  005554  067206                DT13
3228                                 DF20
3229  005556  060132          ;ERROR 206
3230  005560  065051                EM20                    ;MSG B1 ERROR
3231  005562  066446                DH33
3232  005564  067206                DT13
3233                                 DF20
3234  005566  060763          ;ERROR 207
3235  005570  064713                EM36                    ;CYL ADDR IN RKMR3 INCORRECT
3236  005572  066104                DH25                    ;AFTER SEEK CMD
3237  005574  067012                DT4
3238                                 DF6
3239  005576  060147          ;ERROR 210
3240  005600  064713                EM21                    ;CERR SET
3241  005602  066044                DH25
3242  005604  067036                DT1
3243                                 DF10
3244  005606  062534          ;ERROR 211
3245  005610  064713                EM67                    ;READ CYL 0 HEADERS ON CYL 1
3246  005612  066044                DH25
3247  005614  067036                DT1
2248                                 DF10
3249  005616  061063          ;ERROR 212
3250  005620  065051                EM38                    ;CYL DIFF IN RKMR2 NOT = CALDIF
3251  005622  066104                DH33                    ;DURING SEEK CMD
3252  005624  067012                DT4
3253                                 DF6
3254  005626  060063          ;ERROR 213
3255  005630  065333                EM17                    ;MSG AQ ERROR
3256  005632  066446                DH41                    ;DURING RECAL CMD
3257  005634  067206                DT13
3258                                 DF20
3259  005636  060100          ;ERROR 214
3260  005640  065333                EM18                    ;MSG BQ ERROR
3261  005642  066446                DH41
3262  005644  067206                DT13
3263                                 DF20
                               ;ERROR 215
```

```
3264  005646  060115                EM19                    ;MSG A1 ERROR
3265  005650  065333                DH41
3266  005652  066446                DT13
3267  005654  067206                DF20
3268                          ;ERROR 216
3269  005656  060132                EM20                    ;MSG B1 ERROR
3270  005660  065333                DH41
3271  005662  066446                DT13
3272  005664  067206                DF20
3273                          ;ERROR 217
3274  005666  061063                EM38                    ;CYL DIFF IN RKMR2 NOT=CALDIF
3275  005670  065333                DH41
3276  005672  066104                DT4
3277  005674  067012                DF6
3278                          ;ERROR 220
3279  005676  060147                EM21                    ;CERR SET
3280  005700  064507                DH17                    ;AFTER RECAL CMD
3281  005702  066044                DT1
3282  005704  067036                DF10
3283                          ;ERROR 221
3284  005706  060063                EM17                    ;MSG A0 ERROR
3285  005710  064507                DH17
3286  005712  066446                DT13
3287  005714  067206                DF20
3288                          ;ERROR 222
3289  005716  060115                EM19                    ;MSG A1 ERROR
3290  005720  064507                DH17
3291  005722  066446                DT13
3292  005724  067206                DF20
3293                          ;ERROR 223
3294  005726  061134                EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3295  005730  064507                DH17
3296  005732  066044                DT1
3297  005734  067036                DF10
3298                          ;ERROR 224
3299  005736  062500                EM66                    ;CYL ADDR IN RKMR3 INCORRECT
3300  005740  064507                DH17
3301  005742  066044                DT1
3302  005744  067036                DF10
3303                          ;ERROR 225
3304  005746  062573                EM68                    ;READING CYL 1 HEADERS ON CYL 0
3305  005750  064507                DH17
3306  005752  066044                DT1
3307  005754  067036                DF10
3308                          ;ERROR 226
3309  005756  057704                EM12                    ;NO RDY
3310  005760  064732                DH26                    ;AFTER READ DATA CMD
3311  005762  066044                DT1
3312  005764  067036                DF10
3313                          ;ERROR 227
3314  005766  060147                EM21                    ;CERR SET
3315  005770  064732                DH26
3316  005772  066044                DT1
3317  005774  067122                DF15
3318                          ;ERROR 230
3319  005776  063515                EM87                    ;CANT READ BSE INFO
```

```
3320  006000  065775                DH53                    ;ON SECT 10,12,14,16,18,20
3321  006002  066044                DT1
3322  006004  067162                DF17
3323                          ;ERROR 231
3324  006006  000000                0
3325  006010  000000                0
3326  006012  000000                0
3327  006014  000000                0
3328                          ;ERROR 232
3329  006016  060763                EM36                    ;CYL ADDR IN RKMR3 NOT=RKDC
3330  006020  064713                DH25                    ;AFTER SEEK CMD
3331  006022  066044                DT1
3332  006024  066772                DF5
3333                          ;ERROR 233
3334  006026  063515                EM87                    ;CANNOT READ BSE INFO
3335  006030  065354                DH42                    ;ON SECT 0,2,4,6,8
3336  006032  066044                DT1
3337  006034  067162                DF17
3338                          ;ERROR 234
3339  006036  000000                0
3340  006040  000000                0
3341  006042  000000                0
3342  006044  000000                0
3343                          ;ERROR 235
3344  006046  062632                EM69                    ;ALIGN CARTRIDGE USED
3345  006050  065414                DH44                    ;WILL BYPASS FORMAT & ALL R/W TESTS
3346  006052  066044                DT1
3347  006054  067036                DF10
3348                          ;ERROR 236
3349  006056  062142                EM56                    ;UNEXP MEM PARITY TRAP
3350  006060  064672                DH23                    ;TEST #, TRAP PC
3351  006062  066442                DT11
3352  006064  066752                DF3
3353                          ;ERROR 237
3354  006066  062665                EM71                    ;DSC SET
3355  006070  064644                DH22                    ;AFTER DRIVE CLEAR CMD
3356  006072  066044                DT1
3357  006074  067036                DF10
3358                          ;ERROR 240
3359  006076  062573                EM68                    ;READ CYL 1 HEADERS ON CYL 0
3360  006100  064507                DH17                    ;AFTER RECAL CMD
3361  006102  066044                DT1
3362  006104  067036                DF10
3363                          ;ERROR 241
3364  006106  060763                EM36                    ;RKMR3 NOT = RKDC
3365  006110  064415                DH14                    ;AFTER SEEK WITH BAD PARITY
3366  006112  066264                DT8
3367  006114  067012                DF6
3368                          ;ERROR 242
3369  006116  061063                EM38                    ;CYL DIFF IN RKMR2 INCORRECT
3370  006120  064415                DH14
3371  006122  066264                DT8
3372  006124  067012                DF6
3373
3374                          ;ERROR 243
3375  006126  060763                EM36                    ;CYL ADDR IN RKMR3 INCORRECT
```

```
                        ERROR POINTER TABLE

3376  006130  064713            DH25                    ;AFTER SEEK CMD
3377  006132  066264            DT8
3378  006134  067012            DF6
3379                      ;ERROR 244
3380  006136  062754            EM74                    ;RTZ NOT SET
3381  006140  065333            DH41                    ;DURING RECAL CMD
3382  006142  066044            DT1
3383  006144  067036            DF10
3384                      ;ERROR 245
3385  006146  057735            EM13                    ;NO ATTN
3386  006150  065610            DH48                    ;AFTER SEEK TO INVALID CYL
3387  006152  066044            DT1
3388  006154  067036            DF10
3389                      ;ERROR 246
3390  006156  063001            EM75                    ;IDAE NOT SET
3391  006160  065610            DH48
3392  006162  066104            DT4
3393  006164  067012            DF6
3394                      ;ERROR 247
3395  006166  060575            EM32                    ;FAULT NOT SET
3396  006170  065610            DH48
3397  006172  066104            DT4
3398  006174  067012            DF6
3399                      ;ERROR 250
3400  006176  063027            EM76                    ;PIP SET
3401  006200  065610            DH48
3402  006202  066104            DT4
3403  006204  067012            DF6
3404                      ;ERROR 251
3405  006206  060036            EM16                    ;DSC NOT SET
3406  006210  065610            DH48
3407  006212  066104            DT4
3408  006214  067012            DF6
3409                      ;ERROR 252
3410  006216  060063            EM17                    ;MSG A0 ERROR
3411  006220  065610            DH48
3412  006222  066446            DT13
3413  006224  067206            DF20
3414                      ;ERROR 253
3415  006226  060100            EM18                    ;MSG B0 ERROR
3416  006230  065610            DH48
3417  006232  066446            DT13
3418  006234  067206            DF20
3419                      ;ERROR 254
3420  006236  060115            EM19                    ;MSG A1 ERROR
3421  006240  065610            DH48
3422  006242  066446            DT13
3423  006244  067206            DF20
3424                      ;ERROR 255
3425  006246  060132            EM20                    ;MSG B1 ERROR
3426  006250  065610            DH48
3427  006252  066446            DT13
3428  006254  067206            DF20
3429                      ;ERROR 256
3430  006256  061063            EM38                    ;CYL DIFF IN RKMR2 NOT='CYL DIF'
3431  006260  065610            DH48
```

```
3432  006262  066104                DT4
3433  006264  067012                DF6
3434                        ;ERROR 257
3435  006266  060763                EM36          ;CYL ADDR IN RKMR3 NOT=RKDC
3436  006270  065610                DH48
3437  006272  066104                DT4
3438  006274  067012                DF6
3439                        ;ERROR 260
3440  006276  000000                0
3441  006300  000000                0
3442  006302  000000                0
3443  006304  000000                0
3444                        ;ERROR 261
3445  006306  000000                0
3446  006310  000000                0
3447  006312  000000                0
3448  006314  000000                0
3449                        ;ERROR 262
3450  006316  063050                EM77          ;FAULT NOT CLEARED
3451  006320  064644                DH22          ;AFTER DRIVE CLEAR CMD
3452  006322  066044                DT1
3453  006324  067036                DF10
3454                        ;ERROR 263
3455  006326  063076                EM78          ;CYL DIFF IN RKMR2 NOT=1 IN SEEK TO SELF
3456  006330  064415                DH14          ;AFTER SEEK WITH BAD PARITY
3457  006332  066264                DT8
3458  006334  067012                DF6
3459                        ;ERROR 264
3460  006336  061217                EM41          ;CYL ADDR NOT CLEARED
3461  006340  064776                DH30          ;AFTER READ HEADER CMD
3462  006342  066526                DT14
3463  006344  067262                DF22
3464                        ;ERROR 265
3465  006346  060100                EM18          ;MSG BO ERROR
3466  006350  064644                DH22          ;AFTER DRIVE CLEAR CMD
3467  006352  066446                DT13
3468  006354  067206                DF20
3469                        ;ERROR 266
3470  006356  060132                EM20          ;MSG B1 ERROR
3471  006360  064644                DH22
3472  006362  066446                DT13
3473  006364  067206                DF20
3474                        ;ERROR 267
3475  006366  060100                EM18
3476  006370  065250                DH39          ;AFTER WRITE HEADER CMD
3477  006372  066446                DT13
3478  006374  067206                DF20
3479                        ;ERROR 270
3480  006376  060132                EM20
3481  006400  065250                DH39
3482  006402  066446                DT13
3483  006404  067206                DF20
3484                        ;ERROR 271
3485  006406  060100                EM18
3486  006410  064776                DH30          ;AFTER READ HEADER CMD
3487  006412  066446                DT13
```

```
3488  006414  067206                          DF20
3489                                  ;ERROR 272
3490  006416  060132                          EM20
3491  006420  064776                          DH30
3492  006422  066446                          DT13
3493  006424  067206                          DF20
3494                                  ;ERROR 273
3495  006426  060063                          EM17          ;MSG A0 ERROR
3496  006430  064644                          DH22          ;AFTER DRV CLR CMD
3497  006432  066446                          DT13
3498  006434  067206                          DF20
3499                                  ;ERROR 274
3500  006436  060115                          EM19          ;MSG A1 ERROR
3501  006440  064644                          DH22
3502  006442  066446                          DT13
3503  006444  067206                          DF20
3504                                  ;ERROR 275
3505  006446  060100                          EM18          ;MSG B0 ERROR
3506  006450  064507                          DH17          ;AFTER RECAL CMD
3507  006452  066446                          DT13
3508  006454  067206                          DF20
3509                                  ;ERROR 276
3510  006456  060132                          EM20          ;MSG B1 ERROR
3511  006460  064507                          DH17
3512  006462  066446                          DT13
3513  006464  067206                          DF20
3514                                  ;ERROR 277
3515  006466  060063                          EM17          ;MSG A0 ERROR
3516  006470  065250                          DH39          ;AFTER WRITE HEADER CMD
3517  006472  066446                          DT13
3518  006474  067206                          DF20
3519                                  ;ERROR 300
3520  006476  060115                          EM19          ;MSG A1 ERROR
3521  006500  065250                          DH39
3522  006502  066446                          DT13
3523  006504  067206                          DF20
3524                                  ;ERROR 301
3525  006506  060063                          EM17
3526  006510  064776                          DH30          ;AFTER READ HEADER CMD
3527  006512  066446                          DT13
3528  006514  067206                          DF20
3529                                  ;ERROR 302
3530  006516  060115                          EM19
3531  006520  064776                          DH30
3532  006522  066446                          DT13
3533  006524  067206                          DF20
3534                                  ;ERROR 303
3535  006526  061134                          EM39          ;CYL DIFF/OFFSET NOT CLEARED
3536  006530  065250                          DH39          ;AFTER WRITE HEADER CMD
3537  006532  066526                          DT14
3538  006534  067262                          DF22
3539                                  ;ERROR 304
3540  006536  061217                          EM41          ;CYL ADDR NOT CLEARED
3541  006540  065250                          DH39
3542  006542  066526                          DT14
3543  006544  067262                          DF22
```

```
3544
3545
3546  006546  063204          ;ERROR 305
3547  006550  065232                EM80            ;UNLD NOT SET
3548  006552  066044                DH38            ;AFTER LIMIT DETECT
3549  006554  067036                DT1
3550                           DF10
3551  006556  063234          ;ERROR 306
3552  006560  064153                EM81            ;SPIN NOT SET
3553  006562  066044                DH9             ;AFTER START SPIN CMD.
3554  006564  067036                DT1
3555                           DF10
3556  006566  063262          ;ERROR 307
3557  006570  065333                EM82            ;RTZ NOT SET
3558  006572  066044                DH41            ;DURING RECAL CMD
3559  006574  067036                DT1
3560                           DF10
3561  006576  063307          ;ERROR 310
3562  006600  063736                EM83            ;READ HEADER ERROR
3563  006602  066332                DH1
3564  006604  067242                DT9
3565                           DF21
3566  006606  063307          ;ERROR 311
3567  006610  063736                EM83
3568  006612  066376                DH1
3569  006614  067242                DT10
3570                           DF21
3571  006616  063347          ;ERROR 312
3572  006620  065250                EM84            ;FORMAT NOT SET
3573  006622  066044                DH39            ;AFTER WRITE HEADER CMD
3574  006624  067036                DT1
3575                           DF10
3576  006626  063347          ;ERROR 313
3577  006630  064776                EM84
3578  006632  066044                DH30            ;AFTER READ HEADER CMD
3579  006634  067036                DT1
3580                           DF10
3581  006636  062165          ;ERROR 314
3582  006640  063736                EM57            ;WCE AT CYL 411,TRK 2,SEC 21
3583  006642  066044                DH1
3584  006644  066756                DT1
3585                           DF4
3586  006646  062221          ;ERROR 315
3587  006650  064527                EM58            ;SPOK NOT CLEARED
3588  006652  066044                DH18            ;AFTER UNLD CMD
3589  006654  067036                DT1
3590                           DF10
3591  006656  062652          ;ERROR 316
3592  006660  064153                EM70            ;UNEXP ATTN
3593  006662  066044                DH9             ;AFTER START SPIN CMD
3594  006664  067036                DT1
3595                           DF10
3596  006666  062652          ;ERROR 317
3597  006670  064246                EM70
3598  006672  066044                DH11            ;AFT SPIN CMD &  FWD SET
3599  006674  067036                DT1
                               DF10
```

```
3600
3601  006676  062652          ;ERROR 320
3602  006700  064305                EM70
3603  006702  066044                DH12          ;AT INNER LIMIT FROM ST SPIN CMD
3604  006704  067036                DT1
3605                           DF10
3606  006706  062652          ;ERROR 321
3607  006710  064346                EM70
3608  006712  066044                DH13          ;FROM OUTER LIM TO CYL 0 DURING LOADING
3609  006714  067036                DT1
3610                           DF10
3611  006716  060115          ;ERROR 322
3612  006720  064445                EM19          ;MSG A1 ERROR
3613  006722  066446                DH16          ;AFT LD HEAD REG & SEEK
3614  006724  067206                DT13
3615                           DF20
3616  006726  060132          ;ERROR 323
3617  006730  064445                EM20          ;MSG B1 ERROR.
3618  006732  066446                DH16
3619  006734  067206                DT13
3620                           DF20
3621  006736  061464          ;ERROR 324
3622  006740  064445                EM46          ;MSG A2 ERROR
3623  006742  066526                DH16
3624  006744  067262                DT14
3625                           DF22
3626  006746  061477          ;ERROR 325
3627  006750  064445                EM47          ;MSG B2 ERROR
3628  006752  066526                DH16
3629  006754  067262                DT14
3630                           DF22
3631  006756  062733          ;ERROR 326
3632  006760  063430                EM73          ;CTO SET
3633  006762  066044                EM86          ;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
3634  006764  066726                DT1
3635                           DF2
3636  006766  063163          ;ERROR 327
3637  006770  063430                EM79          ;NED SET
3638  006772  066044                EM86
3639  006774  066726                DT1
3640                           DF2
3641
3642  006776  057324          ;ERROR 330
3643  007000  063430                EM5           ;MDS SET
3644  007002  066044                EM86
3645  007004  066726                DT1
                               DF2
```

```
3646
3647                                    .SBTTL  PROGRAM SETUP
3648
3649  007006  012737  000001  001336  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
3650  007014  005037  001340                  CLR     BYPT16
3651  007020  005037  001342                  CLR     MODTST
3652  007024  000450                          BR      PRGSRT          ;START PROGRAM
3653
3654  007026  005037  001336          BYT16:  CLR     PARAM
3655  007032  012737  000001  001340          MOV     #1,BYPT16       ;SET FLAG TO BYPASS TEST 16
3656  007040  005037  001342                  CLR     MODTST
3657  007044  000440                          BR      PRGSRT
3658
3659  007046  012737  000001  001336  BYT16A: MOV     #1,PARAM
3660  007054  012737  000001  001340          MOV     #1,BYPT16
3661  007062  005037  001342                  CLR     MODTST
3662  007066  000427                          BR      PRGSRT
3663
3664  007070  005037  001336          MDTST:  CLR     PARAM
3665  007074  005037  001340                  CLR     BYPT16
3666  007100  012737  000001  001342          MOV     #1,MODTST
3667  007106  000417                          BR      PRGSRT
3668
3669  007110  005037  001336          MDTSTA: CLR     PARAM
3670  007114  012737  000001  001340          MOV     #1,BYPT16
3671  007122  012737  000001  001342          MOV     #1,MODTST
3672  007130  000406                          BR      PRGSRT
3673  007132  005037  001336          START:  CLR     PARAM           ;CLEAR FOR 200 START
3674  007136  005037  001340                  CLR     BYPT16
3675  007142  005037  001342                  CLR     MODTST
3676  007146  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
3677  007150  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3678  007154  012746  000000                  MOV     #PRO,-(SP)      ;PSW LOADED TO BE
3679  007160  012746  007166                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
3680  007164  000002                          RTI                     ;ENABLE ALL INTERRUPTS
3681
3682  007166  004737  052276          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
3683                                                                  ;& TURN ON KB INTERRUPT
3684
3685
3686                                          ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3687                                          ;*** ANY DEVICE WHICH SETS ITS   ***
3688                                          ;*** INTERRUPT ENABLE BIT WILL   ***
3689                                          ;*** SERVICED.                   ***
3690
3691                                          ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3692                                          ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3693                                          ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3694
3695                                          ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3696
3697                                                                  ;SYSMAC 'SETUP'
3698                                  .SBTTL  INITIALIZE THE COMMON TAGS
3699                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3700  007172  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3701  007176  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
```

```
                        CMP     #SWR,R6 ;;DONE?
                        BNE     .-6             ;;LOOP BACK IF NO
                        MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                ;;INITIALIZE A FEW VECTORS
                        MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
                        MOV     #340,@#IOTVEC+2 ;;LEVEL 7
                        MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
                        MOV     #340,@#EMTVEC+2 ;;LEVEL 7
                        MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
                        MOV     #340,@#TRAPVEC+2;LEVEL 7
                        MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
                        MOV     #340,@#PWRVEC+2 ;;LEVEL 7
                        MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
                        CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
                        CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
                        MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
                        MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                        MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
                        MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
                        MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
                        MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
                        MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
                        CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
                        BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                ;;AND  THE HARDWARE SWR IS NOT = -1
                        BR      65$             ;;BRANCH IF NO TIMEOUT
                64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
                        RTI
                65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
                        MOV     #DISPREG,DISPLAY
                66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

                        CLR     $PASS           ;;CLEAR PASS COUNT
                        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
                        BEQ     67$             ;;YES,USE NON-APT SWITCH
                        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
                67$:
                MEMPAR: MOV     #1$,ERRVEC      ;SET TIMEOUT VECTOR
                        MOV     #PR7,ERRVEC+2

                        MOV     #MEMBAS,R1      ;ADDR OF MEM CSR
                3$:     CLR     (R1)            ;SEE IF CAN REFERENCE
                        MOV     #1,(R1)         ;SET ENABLE BIT IF YES
                        MOV     #MEMERR,MEMVEC  ;LOAD MEMORY CHECK VECTOR IF NO TIMEOUT
                        MOV     #PR7,MEMVEC+2
                        BR      2$

                1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
                2$:     ADD     #2,R1           ;TRY NEXT CSR
                        CMP     R1,#MEMBAS+40   ;SEE IF TRIED ALL
                        BNE     3$              ;BR IF NO
                        MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
                        CLR     ERRVEC+2
```

```
                        JSR     PC,CLRFLG       ;CLEAR DDUMP THRU SIZFLG
                        CLR     $DEVCT
                        CLR     $UNIT

                        ;
                        ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
                        ;
START1: TST     42
                        BNE     1$              ;BR IF AUTO
                        JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
                        CMPB    41,#13          ;13=LOADED BY XXDP
                        BNE     2$
                        INC     DDUMP           ;SET RK06 DUMP MODE FLAG
                        TYPE    ,MSG2           ;REPLACE DRO PACK W/SCRATCH & DO<CR>
                        JMP     ST2
1$:     JMP     ST3
2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG

                        ;
                        ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
                        ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
                        ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
                        ;       EX:     DRIVES TO BE TESTED:  1,2,4<CR>
                        ;
ST2:    TST     PARAM
                        BNE     1$              ;BR IF 220 START
                        JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
                        JSR     PC,GDRVS        ;GET DR NOS.
                        TYPE    ,MSG4           ;BUSS ADDR
                        JSR     PC,GBA          ;GET BA
                        TYPE    ,MSG5           ;CONT INT VECTOR
                        JSR     PC,GINT         ;GET INT VECTOR
                        BR      ST5

                        ;
                        ;AUTO MODE
                        ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
                        ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
                        ;ON THE BUSS
                        ;
ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
                        BNE     1$
                        INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
                        JSR     PC,TITLE
                        TYPE    ,MSG7           ;DRO NOT TSTD
                        BR      ST4
1$:     INC     ACT11           ;SET ACT AUTO FLAG.

ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
                        MOV     #210,RKVEC      ;DEFAULT VALUE
                        JSR     PC,SETINT
                        INC     SIZFLG          ;DO "SIZE THE BUSS" TEST
```

```
3814
3815  007740  005037  003316          ST5:    CLR     UNLD            ;INITIALIZE FLAGS
3816  007744  005037  003320                  CLR     BADHDR                  ;USED IN 'STOP ROUTINE
3817  007750  005037  003322                  CLR     HPEND                   ;FOR VALID PROGRAM HALTS
3818  007754  005037  001176                  CLR     $ESCAPE
3819  007760  012737  003456  001346          MOV     #DRIV0,DRVPTR   ;SETUP
3820  007766  005037  001220                  CLR     $DEVCT          ;NO. OF DRVS DONE
3821  007772  005037  001222                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
3822  007776  012737  010044  000004          MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
3823  010004  005777  171316                  TST     @LKS            ;SEE IF L-CLOCK THERE
3824  010010  005237  003476                  INC     LCLKF           ;PRESENT, SET FLAG.
3825  010014  013700  001330                  MOV     LCVEC,R0        ;VECTOR ADDR
3826  010020  012737  010106  000004          MOV     #2$,ERRVEC
3827  010026  005777  171266                  TST     @PKS            ;SEE IF P-CLOCK THERE
3828  010032  005237  003500                  INC     PCLKF           ;PRESENT, SET FLAG
3829  010036  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3830  010042  000412                          BR      3$
3831
3832  010044  022626          1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
3833  010046  012737  010112  000004          MOV     #4$,ERRVEC
3834  010054  005777  171240                  TST     @PKS            ;SEE IF P-CLOCK THERE
3835  010060  005237  003500                  INC     PCLKF           ;PRESENT, SET FLAG
3836  010064  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3837  010070  005237  003502          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
3838  010074  012720  047344                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
3839  010100  012710  000300                  MOV     #PR6,(R0)
3840  010104  000407                          BR      TST1            ;;GO TO NEXT TEST
3841
3842  010106  022626          2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
3843  010110  000767                          BR      3$
3844
3845  010112  022626          4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
3846  010114  005037  003502                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
3847  010120  104401  056150                  TYPE    ,MSG13          ;ALL TIMING TESTS BYPASSED
3848
3849
```

```
                        .SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

                        ;;*****************************************************************
                        ;*TEST 1         REFERENCE ALL CONTROLLER REGISTERS
                        ;*
                        ;*      THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
                        ;*      CAN BE ACCESSED.  THE INABILITY  TO BE ACCESSED WILL
                        ;*      RESULT IN A TIMEOUT TRAP WITH  AN ERROR MSG.  ANY
                        ;*      ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                        ;*      TESTS AND JUMPING TO  'END OF PASS'
                        ;*
                        ;;*****************************************************************
010124  000004         TST1:  SCOPE
010126  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
010134  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
                        
010140  012746  000000            MOV     #PRO,-(SP)      ;RESET PSW TO PRIORITY 0
010144  012746  010152            MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
010150  000002                    RTI
010152                  5$:
                        
010152  012737  010270  000004        MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
010160  013705  001264            MOV     $BASE,R5        ;SETUP INDEX REG.
010164  005765  000000            TST     RKCS1(R5)       ;REFERENCE ALL THE
010170  005765  000010            TST     RKCS2(R5)       ;CONTROLLER REGISTERS
010174  005765  000002            TST     RKWC(R5)
010200  005765  000004            TST     RKBA(R5)
010204  005765  000006            TST     RKDA(R5)
010210  005765  000012            TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
010214  005765  000014            TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
010220  005765  000016            TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
010224  005765  000020            TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
010230  005765  000024            TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
010234  005765  000026            TST     RKMR1(R5)
010240  005765  000034            TST     RKMR2(R5)
010244  005765  000036            TST     RKMR3(R5)
010250  005765  000030            TST     RKECPS(R5)
010254  005765  000032            TST     RKECPT(R5)
                        
010260  012737  050146  000004        MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
010266  000404                    BR      TST2            ;;GO TO NEXT TEST
                        
010270  022626           1$:      CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
010272  104007                    ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
010274  000137  043132            JMP     $EOP1
                        
                        
                        ;;*****************************************************************
                        ;*TEST 2         SIZE THE BUSS
                        ;*
                        ;*      THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                        ;*      EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                        ;*      MANUAL MODE.
                        ;*      EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
                        ;*      CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
                        ;*      DRIVE WILL BE TESTED.  IF SET, THE PROGRAM WILL BYPASS
```

```
                                T2      SIZE THE BUSS

                                ;*      TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                ;*      MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                ;*      DICATING THE OTHER PORT IS ACCESSED.
                                ;*
                                ;;**********************************************************************
010300  000004                  TST2:   SCOPE
010302  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
010310  012706  001100                  MOV     #$STACK,SP      ;RESTORE STK PTR
010314  005237  001462                  INC     BYPCERR         ;DO NOT TEST CERR IN 'FRDY'

010320  132737  000200  001231          BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
010326  001002                          BNE     14$             ;BR IF YES
010330  000137  010444                  JMP     12$             ;ELSE DO NORM SIZING OR VERIFY

010334  104401  056030          14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
010340  005037  003454                  CLR     DRIVS           ;# OF DRIVES PRESENT
010344  005000                          CLR     R0              ;DRV ADDR
010346  012701  003456                  MOV     #DRIVO,R1       ;DRV FLAG
010352  013702  001266                  MOV     $DEVM,R2        ;APT DEVICE MAP

010356  032702  000001          15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
010362  001410                          BEQ     16$             ;BR IF NO
010364  005237  003454                  INC     DRIVS           ;ELSE INCR DRIVE COUNT
010370  005211                          INC     (R1)            ;& SET DRIVE PRESENT FLAG
010372  104401  001205                  TYPE    ,$CRLF
010376  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DRIVE #
010400  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
010402      001                         .BYTE   1               ;;TYPE 1 DIGIT(S)
010403      000                         .BYTE   0               ;;SUPPRESS LEADING ZEROS

010404  005721          16$:    TST     (R1)+           ;ADV POINTER TO NEXT FLAG
010406  005200                  INC     R0              ;INC DRIVE #
010410  022700  000010          CMP     #8.,R0          ;ALL 8 TESTED?
010414  001402                  BEQ     17$             ;BR IF YES

010416  006002                  ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
010420  000756                  BR      15$             ;& TRY AGAIN

010422  005737  003454  17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
010426  001402                  BEQ     18$             ;BR IF NO
010430  000137  011362          JMP     NUDRV           ;ELSE EXIT TEST

010434  104126          18$:    ERROR   126             ;NO DRIVES FOUND IN $DEVM
010436  000000                  HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
010440  000137  007740          JMP     ST5             ;TO TRY AGAIN

010444  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
010452  013737  001414  003372          MOV     T10,TEMP1
010460  004737  043612                  JSR     PC,FRDY         ;FIND RDY
010464  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
010466  005737  003504                  TST     SIZFLG
010472  001562                          BEQ     TST3            ;;DO NOT SIZE, GOTO NEXT TEST
010474  104401  056030                  TYPE    ,MSG10          ;WILL TEST DRIVES
010500  005037  003454                  CLR     DRIVS           ;# OF DRIVES PRESENT
010504  005000                          CLR     R0              ;DRV ADDR
```

```
3962  010506  012701  003456              MOV     #DRIV0,R1       ;DRV FLAG
3963  010512                        1$:
3964  010512  104415                     SCOP1
3965  010514  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
3966
3967  010520  012765  000040  000010      MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
3968  010526  013737  001414  003372      MOV     T10,TEMP1
3969  010534  004737  043612              JSR     PC,FRDY         ;FIND RDY
3970  010540  104120                     ERROR   120             ;RDY NOT SET BY END OF SCLR
3971  010542  010065  000010              MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
3972  010546  012765  000001  000000      MOV     #SELDRV,RKCS1(R5)
3973  010554  013737  001414  003372      MOV     T10,TEMP1
3974  010562  004737  043612              JSR     PC,FRDY         ;FIND RDY
3975  010566  104117                     ERROR   117             ;NO RDY AFTER SELECT DR. CMD
3976  010570  032737  100000  003334      BIT     #CERR,HCS1
3977  010576  001046                     BNE     2$
3978  010600  013737  003362  003372      MOV     HMR2,TEMP1
3979  010606  042737  177770  003372      BIC     #^C<DRVMSK>,TEMP1
3980  010614  020037  003372              CMP     R0,TEMP1        ;S/B SAME
3981  010620  001016                     BNE     3$
3982  010622  005700                     TST     R0
3983  010624  001003                     BNE     4$
3984  010626  005737  003446              TST     DDPCH           ;SEE IF XXDP CHAIN MODE
3985  010632  001014                     BNE     5$
3986  010634  005237  003454        4$:   INC     DRIVS           ;INC DRIVE COUNT.
3987  010640  005211                     INC     (R1)            ;SET DRIVE PRESENT FLAG
3988  010642  104401  001205              TYPE    ,$CRLF
3989  010646  010046                     MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3990                                                             ;;TYPE DR #
3991  010650  104403                     TYPOS                   ;;GO TYPE--OCTAL ASCII
3992  010652     001                     .BYTE   1               ;;TYPE 1 DIGIT(S)
3993  010653     000                     .BYTE   0               ;;SUPPRESS LEADING ZEROS
3994  010654  000403                     BR      5$
3995
3996  010656  004737  044320        3$:   JSR     PC,BYP          ;TYPE BYPASS DR #
3997  010662  104001                     ERROR   1               ;SELECTED DR # DOES NOT MATCH RKMR2 DR #
3998
3999  010664  005721                5$:   TST     (R1)+           ;ADVANCE PTR TO NEXT DR. FLAG
4000  010666  005200                     INC     R0              ;INC DR #
4001  010670  022700  000010              CMP     #8.,R0
4002  010674  001306                     BNE     1$              ;MORE LEFT.
4003  010676  005737  003454              TST     DRIVS
4004  010702  001054                     BNE     10$
4005  010704  104123                     ERROR   123             ;NO DRIVES FOUND
4006  010706  000000                     HALT                    ;SETUP CORRECTLY
4007  010710  000137  007740              JMP     ST5             ;& PRESS 'CONT'
4008
4009  010714  032737  001000  003336  2$:   BIT     #MDS,HCS2
4010  010722  001015                     BNE     6$
4011  010724  032737  000400  003336      BIT     #UFE,HCS2
4012  010732  001015                     BNE     7$
4013  010734  032737  000001  003346      BIT     #DRA,HDS
4014  010742  001015                     BNE     8$
4015  010744  032737  010000  003336      BIT     #NED,HCS2
4016  010752  001424                     BEQ     9$
4017  010754  000743                     BR      5$
```

```
4018
4019  010756  004737  044320          6$:   JSR     PC,BYP          ;TYPE BYP DR #
4020  010762  104002                        ERROR   2               ;MDS DETECTED
4021  010764  000737                        BR      5$
4022
4023  010766  004737  044320          7$:   JSR     PC,BYP
4024  010772  104003                        ERROR   3               ;UFE DETECTED
4025  010774  000733                        BR      5$
4026
4027  010776  032737  010000  003336  8$:   BIT     #NED,HCS2
4028  011004  001713                        BEQ     4$
4029  011006  104401  056251                TYPE    ,MSG15          ;DRV#
4030  011012  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
4031                                                                ;;TYPE DR#
4032  011014  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
4033  011016     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
4034  011017     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
4035  011020  104010                        ERROR   10              ;DRA & NED BOTH SET
4036  011022  000720                        BR      5$
4037
4038  011024  004737  044320          9$:   JSR     PC,BYP
4039  011030  104004                        ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
4040  011032  000714                        BR      5$
4041  011034  000137  011362          10$:  JMP     NUDRV
4042
4043                                  ;;**********************************************************************
4044                                  ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
4045                                  ;*
4046                                  ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
4047                                  ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
4048                                  ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
4049                                  ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
4050                                  ;*      TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
4051                                  ;*      BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
4052                                  ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
4053                                  ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
4054                                  ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
4055                                  ;*      VERIFY IT WAS NOT SPECIFIED.
4056                                  ;*
4057                                  ;**********************************************************************
4058  011040  000004                  TST3:   SCOPE
4059  011042  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4060  011050  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4061  011054  005000                          CLR     R0              ;DRIVE ADDR
4062  011056  012701  003456                  MOV     #DRIV0,R1       ;DRIVE FLAG
4063  011062                          1$:
4064  011062  104415                          SCOP1
4065  011064  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4066
4067  011070  012765  000040  000010          MOV     #SCLR,RKCS2(R5)
4068  011076  013737  001414  003372          MOV     T10,TEMP1
4069  011104  004737  043612                  JSR     PC,FRDY         ;FIND RDY
4070  011110  104120                          ERROR   120             ;NO RDY AFTER SCLR
4071  011112  010065  000010                  MOV     R0,RKCS2(R5)    ;DRV ADDR
4072  011116  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
4073  011124  013737  001414  003372          MOV     T10,TEMP1
```

```
4074  011132  004737  043612          JSR     PC,FRDY         ;FIND RDY
4075  011136  104117                  ERROR   117             ;NO RDY AFTER SELDRV CMD
4076  011140  032737  100000  003334  BIT     #CERR,HCS1
4077  011146  001036                  BNE     2$
4078  011150  013737  003362  003372  MOV     HMR2,TEMP1
4079  011156  042737  177770  003372  BIC     #↑C<DRVMSK>,TEMP1
4080  011164  020037  003372          CMP     R0,TEMP1        ;S/B SAME
4081  011170  001010                  BNE     3$
4082  011172  005711          11$:    TST     (R1)
4083  011174  001417                  BEQ     5$
4084  011176  005721          4$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR FLAG
4085  011200  005200                  INC     R0              ;INC DR#
4086  011202  022700  000010          CMP     #8.,R0
4087  011206  001325                  BNE     1$              ;MORE LEFT
4088  011210  000466                  BR      TST4            ;;GO TO NEXT TEST
4089
4090  011212  004737  044320  3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
4091  011216  104001                  ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
4092  011220  005711                  TST     (R1)
4093  011222  001765                  BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
4094  011224  005337  003454  12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
4095  011230  005011                  CLR     (R1)            ;CLEAR DRIVE FLAG
4096  011232  000761                  BR      4$
4097
4098  011234  004737  044320  5$:     JSR     PC,BYP
4099  011240  104005                  ERROR   5               ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
4100  011242  000755                  BR      4$
4101
4102  011244  032737  001000  003336  2$:     BIT     #MDS,HCS2
4103  011252  001027                  BNE     6$
4104  011254  032737  000400  003336          BIT     #UFE,HCS2
4105  011262  001027                  BNE     7$
4106  011264  032737  000001  003346          BIT     #DRA,HDS
4107  011272  001005                  BNE     8$
4108  011274  032737  010000  003336          BIT     #NED,HCS2
4109  011302  001423                  BEQ     9$
4110  011304  000404                  BR      10$
4111  011306  032737  010000  003336  8$:     BIT     #NED,HCS2
4112  011314  001726                  BEQ     11$
4113  011316  005711          10$:    TST     (R1)
4114  011320  001726                  BEQ     4$
4115
4116  011322  004737  044320          JSR     PC,BYP          ;TYPE BYPASS DRIVE#
4117  011326  104006                  ERROR   6
4118  011330  000735                  BR      12$
4119
4120  011332  004737  044320  6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
4121  011336  104002                  ERROR   2               ;MDS DETECTED
4122  011340  000762                  BR      8$
4123
4124  011342  004737  044320  7$:     JSR     PC,BYP
4125  011346  104003                  ERROR   3               ;UFE DETECTED
4126  011350  000756                  BR      8$
4127
4128  011352  004737  044320  9$:     JSR     PC,BYP
4129  011356  104004                  ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
```

```
4130  011360  000752                          BR      8$
4131
4132
4133                                          ;
4134                                          ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
4135                                          ;DRIVE PRESENT
4136                                          ;
4137                                          ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
4138                                          ;UNDER TEST
4139
4140  011362  005037  001462          NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
4141                                                                  ;& TEST CERR IN 'FRDY'
4142
4143                                  ;;**************************************************************
4144                                  ;*TEST 4         FIND NEXT DRIVE TO BE TESTED
4145                                  ;*
4146                                  ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
4147                                  ;*      ADDRESS IN '$UNIT'.
4148                                  ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
4149                                  ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
4150                                  ;*
4151                                  ;;**************************************************************
4152  011366  000004                  TST4:   SCOPE
4153  011370  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4154  011376  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4155  011402  012737  000004  001214          MOV     #STN-1,$TESTN
4156  011410  012737  000004  001102          MOV     #STN-1,$TSTNM
4157
4158  011416  005737  003454                  TST     DRIVS           ;ANY DRIVES PRESENT?
4159  011422  001004                          BNE     4$              ;YES BRANCH
4160  011424  104401  056367                  TYPE    ,MSG19          ;ALL DRIVES TESTED
4161  011430  000137  043132                  JMP     $EOP1           ;NO, GO TO END
4162
4163  011434  013701  001346          4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
4164  011440  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
4165  011444  001402                          BEQ     2$              ;YES, BRANCH
4166  011446  005237  001222          1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
4167  011452  005721                  2$:     TST     (R1)+           ;IS DRIVE PRESENT?
4168  011454  001774                          BEQ     1$              ;NO,FIND NEXT DRIVE PRESENT
4169  011456  005737  003446                  TST     DDPCH           ;DDP CHAIN MODE?
4170  011462  001403                          BEQ     3$              ;NO, BRANCH
4171  011464  005737  001222                  TST     $UNIT           ;YES, IS IT DRIVE 0?
4172  011470  001766                          BEQ     1$              ;IF YES, DON'T TEST DR 0
4173  011472  010137  001346          3$:     MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT DR. FLAG
4174  011476  104401  056251                  TYPE    ,MSG15          ;"DRIVE"
4175  011502  013700  001222                  MOV     $UNIT,R0
4176  011506  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
4177                                                                  ;;DRIVE #
4178  011510  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
4179  011512     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
4180  011513     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
4181
4182  011514  104401  001205                  TYPE    ,$CRLF
4183
4184                                  ;;**************************************************************
4185                                  ;*TEST 5         UNLOAD DRIVE TO BE TESTED
```

```
                                ;*
                                ;*      THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
                                ;*      WAITS FOR ATTN & VERIFIES IT CAME FROM  THE CORRECT DRIVE.
                                ;*      IT THEN WAITS FOR SPEED OK TO GO LOW BEFORE
                                ;*      PROCEEDING TO THE NEXT TEST
                                ;*
                                ;;*****************************************************************
011520  000004                  TST5:   SCOPE
011522  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
011530  012706  001100                  MOV     #STACK,SP
011534  005237  003316                  INC     UNLD            ;USED TO CHECK VALID HALT
011540  004737  045522                  JSR     PC,SUBCLR
011544  104024                          ERROR   24              ;CERR AFTER SCLR
011546  012765  000007  000000          MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
011554  013737  001414  003372          MOV     T10,TEMP1
011562  004737  043612                  JSR     PC,FRDY         ;FIND RDY
011566  104011                          ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
011570  004737  044074                  JSR     PC,TSTATN
011574  104012                          ERROR   12              ;NO ATTN AFTER UNLOAD CMD
011576  004737  045522                  JSR     PC,SUBCLR
011602  104024                          ERROR   24              ;CERR AFTER SCLR
011604  013737  001414  003374          MOV     T10,TEMP2
011612  004737  046556                  JSR     PC,FSPOK
011616  104315                          ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
011620                          PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
                                .SBTTL STATIC & CYCLE UP TESTS
                                ;;*****************************************************************
                                ;*TEST 6        REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3
                                ;*
                                ;*       CHECKS THE ABILITY TO REFERENCE ALL
                                ;*      DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.
                                ;*
                                ;;*****************************************************************
011620  000004                  TST6:   SCOPE
011622  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
011630  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
011634  004737  045522                  JSR     PC,SUBCLR
011640  104024                          ERROR   24              ;CERR AFTER SCLR
011642  004737  045150                  JSR     PC,GSTAT
011646  032737  000100  003362          BIT     #D.VV,HMR2
011654  001004                          BNE     4$              ;BR IF VV SET
011656  012737  000040  003424          MOV     #D.DRA,E.AD     ;LOAD EXPECTED VALUE FOR AD
011664  000403                          BR      5$
```

```
4242
4243  011666  012737  000140  003424  4$:     MOV     #<D.DRA!D.VV>,E.A0
4244  011674  005037  003426          5$:     CLR     E.B0            ;EXPECTED MSG B0
4245  011700  012737  000740  003430          MOV     #<D.HDHM!D.BRHM!D.DOOR!D.CART>,E.A1      ;EXPECTED MSG A1
4246  011706  012737  000001  003432          MOV     #1,E.B1         ;EXPECTED MSG B1
4247  011714  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
4248  011720  012737  000002  003436          MOV     #2,E.B2         ;EXPECTED MSG B2
4249  011726  012737  000003  003442          MOV     #3,E.B3         ;EXPECTED MSG B3
4250
4251  011734  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4252  011740  000007                          .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
4253  011742  104016                          ERROR   16              ;MSG A0 ERROR FOR DRIVE UNLOADED
4254  011744  104017                          ERROR   17              ;MSG B0 ERROR
4255  011746  104020                          ERROR   20              ;MSG A1 ERROR
4256  011750  104021                          ERROR   21              ;MSG B1 ERROR
4257
4258  011752  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
4259  011756  001401                          BEQ     64$             ;BR IF YES
4260  011760  104022                          ERROR   22              ;MSG A2 NOT CLEARED FOR DRIVE UNLOADED
4261  011762  005737  001364          64$:    TST     CYLADD          ;SEE IF MSG B2=0
4262  011766  001401                          BEQ     65$             ;BR IF YES
4263  011770  104023                          ERROR   23              ;MSG B2 NOT CLEARED FOR DRIVE UNLOADED
4264  011772                          65$:
4265  011772  023727  001432  000001          CMP     HEADA,#1        ;FOR HEAD 0, B3=1
4266  012000  001401                          BEQ     TST7            ;;GO TO NXT TST IF YES
4267  012002  104056                          ERROR   56              ;HEAD REG IN B3 NOT 0 IN UNLOAD
4268
4269
4270
4271                                  ;;*****************************************************************
4272                                  ;*TEST 7        PRINT DRIVE SERIAL NUMBER
4273                                  ;*
4274                                  ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
4275                                  ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
4276                                  ;*
4277                                  ;;*****************************************************************
4278  012004  000004          TST7:   SCOPE
4279  012006  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4280  012014  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4281
4282  012020  005737  001216                  TST     $PASS
4283  012024  001046                          BNE     TST10           ;;GO TO NEXT IF NOT FIRST PASS
4284  012026  004737  045522                  JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
4285  012032  104024                          ERROR   24              ;CERR AFTER SCLR
4286
4287  012034  104401  056263                  TYPE    ,MSG16          ;DRIVE SERIAL NO.
4288  012040  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
4289  012046  004737  045150                  JSR     PC,GSTAT        ;GET STATUS
4290  012052  013701  003362                  MOV     HMR2,R1         ;GET SERIAL #
4291  012056  012704  054000                  MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
4292  012062  010446                          MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
4293  012064  012703  000003                  MOV     #3,R3           ;SETUP CHAR COOUNT
4294  012070  006101                          ROL     R1              ;INITIALIZE BIT POSITIONS
4295  012072  006101                          ROL     R1
4296  012074  006101                  1$:     ROL     R1              ;GET NEXT 4 BITS
4297  012076  006101                          ROL     R1
```

```
                                        ROL     R1
                                        ROL     R1
                                        MOV     R1,R0           ;GET WORKING COPY
                                        BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
                                        BIS     #60,R0          ;CONVERT TO ASCII DIGIT
                                        MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
                                        DEC     R3
                                        BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
                                        CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR
                                        JSR     PC,$SUPRS       ;TYPE
                                        TYPE    ,$CRLF
                                        TYPE    ,$CRLF

                                ;;**************************************************************
                                ;*TEST 10        SET VV WITH PACK CMD
                                ;*
                                ;*      IF VV IS RESET, THE PACK CMD IS USED TO SET IT.
                                ;*
                                ;;**************************************************************
                                TST10:  SCOPE
                                        MOV     #1,$TIMES       ;;DO 1 ITERATION
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                        CLR     RKMR1(R5)       ;SELECT BYTE 0
                                        JSR     PC,GSTAT        ;GET STATUS
                                        BIT     #D.VV,HMR2
                                        BNE     TST11           ;;GO TO NEXT TEST IF VV SET

                                        SCOP1
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                        JSR     PC,SUBCLR
                                        ERROR   24              ;CERR AFTER SCLR

                                        MOV     #PACK,RKCS1(R5) ;CMD TO SET VV
                                        MOV     T10,TEMP1
                                        JSR     PC,FRDY         ;FIND RDY
                                        ERROR   116             ;RDY NOT SET AFTER PACK CMD

                                        BIT     #D.VV,HMR2
                                        BNE     TST11           ;;GO TO NEXT TEST IF VV NOW SET
                                        ERROR   27              ;PACK DID NOT SET V.V.

                                ;;**************************************************************
                                ;*TEST 11        RELEASE DRIVE
                                ;*
                                ;*      TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK
                                ;*
                                ;;**************************************************************
                                TST11:  SCOPE
                                        MOV     #1,$TIMES       ;;DO 1 ITERATION
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                        JSR     PC,SUBCLR       ;DO SUBSYS CLEAR & GET STATUS
                                        ERROR   24              ;CONTR ERROR SET AFTER SCLR
                                        BIT     #UFE,HCS2
```

```
4354  012276  001401                          BEQ     1$
4355  012300  104003                          ERROR   3               ;UFE SET AFTER SCLR
4356
4357  012302                          1$:
4358  012302  104415                          SCOP1
4359  012304  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4360
4361  012310  004737  045522                  JSR     PC,SUBCLR
4362  012314  104024                          ERROR   24              ;CERR AFTER SCLR
4363
4364  012316  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;ADD RELEASE BIT TO $UNIT
4365  012324  004737  045150                  JSR     PC,GSTAT        ;GET STATUS
4366
4367  012330  032737  100000  003334          BIT     #CERR,HCS1      ;CHECK FOR CONTR ERROR
4368  012336  001401                          BEQ     2$
4369  012340  104025                          ERROR   25              ;RLS SET CERR
4370  012342  032737  000400  003336  2$:     BIT     #UFE,HCS2
4371  012350  001401                          BEQ     TST12           ;;GO TO NEXT TEST IF SET
4372  012352  104026                          ERROR   26              ;SACK SET AFTER RLS SENT
4373
4374                                  ;;*****************************************************************
4375                                  ;*TEST 12       DRIVE TYPE TEST
4376                                  ;*
4377                                  ;*      THIS TEST COMPARES DRIVE TYPE IN MSG A AGAINST 'DDT' IN RKDS.
4378                                  ;*      WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.
4379                                  ;*
4380                                  ;;*****************************************************************
4381  012354  000004                  TST12:  SCOPE
4382  012356  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4383  012364  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4384
4385  012370  004737  045522                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
4386  012374  104024                          ERROR   24              ;CONT ERROR SET AFT SUBSYS CLEAR
4387  012376  032737  000400  003362          BIT     #D.DDT,HMR2
4388  012404  001401                          BEQ     2$
4389  012406  104030                          ERROR   30              ;DR TYPE SET IN MR2
4390  012410  032737  000400  003346  2$:     BIT     #DDT,HDS
4391  012416  001401                          BEQ     3$
4392  012420  104031                          ERROR   31              ;DDT SET IN RKDS
4393  012422  032737  000040  003350  3$:     BIT     #DTYE,HER
4394  012430  001401                          BEQ     4$
4395  012432  104032                          ERROR   32              ;DTYE SET IN RKER
4396
4397  012434                          4$:
4398  012434  104415                          SCOP1
4399  012436  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4400
4401  012442  004737  045522                  JSR     PC,SUBCLR
4402  012446  104024                          ERROR   24              ;CERR AFTER SCLR
4403
4404  012450  012765  002001  000000          MOV     #<CDT!SELDRV>,RKCS1(R5)  ;GET STATUS WITH CDT SET
4405  012456  013737  001414  003372          MOV     T10,TEMP1
4406  012464  004737  043612                  JSR     PC,FRDY         ;FIND RDY
4407  012470  104117                          ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
4408  012472  032737  000400  003362          BIT     #D.DDT,HMR2
4409  012500  001401                          BEQ     5$
```

```
                                        ERROR   30              ;DR TYPE SET IN MR2
012502  104030
012504  032737  000400  003346  5$:   BIT     #DDT,HDS
012512  001401                        BEQ     6$
012514  104031                        ERROR   31              ;DDT SET IN RKDS
012516  032737  000040  003350  6$:   BIT     #DTYE,HER       ;DTYE=DDT(NOT)*CDT
012524  001001                        BNE     7$
012526  104033                        ERROR   33              ;DTYE NOT SET AFT WRITING CDT=1
012530  032737  100000  003334  7$:   BIT     #CERR,HCS1
012536  001001                        BNE     TST13           ;;GO TO NEXT TEST
012540  104034                        ERROR   34              ;CERR NOT SET AFT WRITING CDT=1

                                ;;*****************************************************************
                                ;*TEST 13       C-D PARITY ERROR DETECTION
                                ;*
                                ;*      TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
                                ;*      THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
                                ;*      THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
                                ;*      THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
                                ;*      CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
                                ;*      AN ERROR AND SET DCPAR.
                                ;*      THE ERROR CONDITION IS RESET WITH THE CLEAR CMD
                                ;*
                                ;*****************************************************************
012542  000004                TST13:  SCOPE
012544  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
012552  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

012556  004737  045522                JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
012562  104024                        ERROR   24              ;CONT ERROR AFTER SUBSYS CLR
012564  032737  001000  003364        BIT     #D.PAR,HMR3
012572  001401                        BEQ     2$
012574  104035                        ERROR   35              ;C-D PARITY ERROR SET IN MR3
012576  032737  020000  003334  2$:   BIT     #DCPAR,HCS1
012604  001401                        BEQ     3$
012606  104036                        ERROR   36              ;DCPAR SET IN CS1

012610                          3$:
012610  104415                        SCOP1
012612  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

012616  004737  045522                JSR     PC,SUBCLR
012622  104024                        ERROR   24              ;CERR AFTER SCLR

012624  012765  000020  000026        MOV     #PAT,RKMR1(R5)  ;SELECT BYTE 0 & EVEN PARITY
012632  004737  045150                JSR     PC,GSTAT        ;GET STATUS
012636  032737  000200  003364        BIT     #D.FLT,HMR3
012644  001001                        BNE     4$
012646  104037                        ERROR   37              ;FAULT NOT SET IN MR3
012650  032737  001000  003364  4$:   BIT     #D.PAR,HMR3
012656  001001                        BNE     5$
012660  104040                        ERROR   40              ;C-D PARITY ERROR NOT SET IN MR3
012662  032737  020000  003334  5$:   BIT     #DCPAR,HCS1
012670  001001                        BNE     6$
012672  104041                        ERROR   41              ;DCPAR NOT SET AFT WRITING PAT IN MR1
012674  032737  100000  003334  6$:   BIT     #CERR,HCS1
012702  001001                        BNE     TST14           ;;GO TO NEXT TEST
```

```
                        ERROR   42              ;CERR NOT SET BY WRITING PAT IN MR1

                        ;;*****************************************************************
                        ;*TEST 14       VERIFY START SPINDLE CMD
                        ;*
                        ;*      THE PROGRAM CHECKS THE ENTIRE STARTUP SEQUENCE, IE:
                        ;*      BRUSH CYCLE, HEADS HOME, FWD, REV ETC.
                        ;*      BY VERIFYING ALL APPROPRIATE STATUS BITS FOR PROPER SEQUENCING.
                        ;*      THE CYL ADDRESS & CYL DIFFERENCE REGS ARE CHECKED
                        ;*      TO BE ZERO AT THE END OF THE SEQUENCE.
                        ;*
                        ;;*****************************************************************
012706  000004          TST14:  SCOPE
012710  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
012716  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

012722  004737  045522                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
012726  104024                          ERROR   24              ;CERR AFTER SCLR

012730  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5) ;START SPINDLE CMD
012736  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
012744  004737  043612                  JSR     PC,FRDY
012750  104121                          ERROR   121             ;RDY NOT SET AFTER START SPIN CMD

012752  004737  045150                  JSR     PC,GSTAT        ;WORD 0
012756  032737  010000  003362          BIT     #D.SPIN,HMR2
012764  001001                          BNE     13$
012766  104306                          ERROR   306             ;SPIN NOT SET AFTER START SPIN CMD

012770  012737  014212  001176  13$:    MOV     #25$,$ESCAPE
012776  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
013002  000401                          BR      15$
013004  104316                          ERROR   316
013006  012737  010140  003424  15$:    MOV     #<D.SPIN!D.VV!D.DRA>,E.A0       ;LOAD IN EXPECTED VALUES
013014  005037  003426                  CLR     E.B0
013020  012737  000740  003430          MOV     #<D.CART!D.DOOR!D.HDHM!D.BRHM>,E.A1
013026  012737  000001  003432          MOV     #1,E.B1

013034  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
013040  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
013042  104057                          ERROR   57              ;MSG A0 ERROR AFTER START SPIN CMD REC'D BY DRIVE
013044  104060                          ERROR   60              ;MSG B0 ERROR
013046  104061                          ERROR   61              ;MSG A1 ERROR
013050  104062                          ERROR   62              ;MSG B1 ERROR
013052  005737  003502                  TST     @0TIM
013056  001135                          BNE     3$              ;BRANCH IF P OR L CLOCK PRESENT
013060  012737  014236  001176  1$:     MOV     #30$,$ESCAPE
013066  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
013074  013737  001416  003374          MOV     T100,TEMP2      ;SETUP TIMEOUT
013102  004737  044126                  JSR     PC,FATT1        ;FIND ATTN
013106  104067                          ERROR   67              ;NO ATTN AFTER HEAD LOADING
013110                          2$:

013110  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
013116  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
013122  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
```

```
4522  013130  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4523  013136  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
4524  013142  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4525  013150  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4526
4527  013156  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4528  013162  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4529  013164  104063                          ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOAD
4530  013166  104064                          ERROR   64              ;MSG B0 ERROR
4531  013170  104065                          ERROR   65              ;MSG A1 ERROR
4532  013172  104066                          ERROR   66              ;MSG B1 ERROR
4533
4534  013174  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
4535  013200  001401                          BEQ     64$             ;BR IF YES
4536  013202  104175                          ERROR   175             ;MSG A2 NOT CLEARED AT END OF HEAD LOAD
4537  013204  005737  001364          64$:    TST     CYLADD          ;SEE IF MSG B2=0
4538  013210  001401                          BEQ     65$             ;BR IF YES
4539  013212  104176                          ERROR   176             ;MSG B2 NOT CLEARED AT END OF HEAD LOAD
4540  013214                          65$:
4541
4542  013214  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4543  013222  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4544  013230  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4545  013236  013737  001414  003372          MOV     T10,TEMP1
4546  013244  004737  043612                  JSR     PC,FRDY         ;FIND RDY
4547  013250  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4548  013252  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
4549  013256  000401                          BR      66$
4550  013260  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4551  013262                          66$:
4552
4553  013262  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
4554  013270  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
4555  013274  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
4556  013302  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4557  013310  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
4558  013314  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4559  013322  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4560
4561  013330  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4562  013334  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4563  013336  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4564  013340  104265                          ERROR   265             ;MSG B0 ERROR
4565  013342  104274                          ERROR   274             ;MSG A1 ERROR
4566  013344  104266                          ERROR   266             ;MSG B1 ERROR
4567
4568  013346  000137  014024                  JMP     12$
4569
4570  013352  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
4571  013360  013737  001370  001372          MOV     HZ,COUNT
4572  013366  012737  000074  001374          MOV     #60.,SEC
4573  013374  004737  047304                  JSR     PC,CLKON        ;TURN CLK INTR ON FOR 60 SEC MAX
4574  013400  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4575  013406  004737  045150          4$:     JSR     PC,GSTAT
4576  013412  032737  002000  003362          BIT     #D.FWD,HMR2
4577  013420  001004                          BNE     5$
```

```
4578  013422  005737  001376            TST     TIMUP           ;IS 60 SEC DELAY UP?
4579  013426  001767                    BEQ     4$              ;BRANCH IF NO & REPEAT
4580  013430  104070                    ERROR   70              ;FWD NOT SET WITHIN 60 SEC FROM
4581                                                            ;START SPINDLE CMD.
4582  013432  004737  047400      5$:   JSR     PC,CLKOF        ;TURN OFF CLOCK INTERRUPT
4583  013436  012765  100000  000000    MOV     #CCLR,RKCS1(R5)
4584
4585  013444  013737  001370  001372    MOV     HZ,COUNT
4586  013452  012737  000005  001374    MOV     #5,SEC
4587  013460  004737  047304            JSR     PC,CLKON        ;TURN CLK INTR ON FOR 5 SEC MAX
4588  013464  012765  000001  000026  6$:   MOV     #1,RKMR1(R5)    ;WORD 1
4589  013472  004737  045150            JSR     PC,GSTAT
4590  013476  032737  002000  003362    BIT     #D.FWD,HMR2
4591  013504  001404                    BEQ     7$
4592  013506  005737  001376            TST     TIMUP
4593  013512  001764                    BEQ     6$
4594  013514  104075                    ERROR   75              ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
4595                                                            ;FROM START SPINDLE CMD.
4596  013516  004737  047400      7$:   JSR     PC,CLKOF        ;TURN OFF CLK INTERRUPT
4597  013522  004737  044074            JSR     PC,TSTATN       ;TEST FOR ATTN
4598  013526  000401                    BR      17$
4599  013530  104320                    ERROR   320             ;UNEXP ATTN AFTER INNER LIM DETECT
4600  013532  012737  030140  003424  17$:   MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
4601  013540  005037  003426            CLR     E.B0
4602  013544  012737  025720  003430    MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4603  013552  012737  000001  003432    MOV     #1,E.B1
4604
4605  013560  004737  044334            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4606  013564  000000                    .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4607  013566  104076                    ERROR   76              ;MSG A0 ERROR AT INNER LIMIT DETECT
4608  013570  104077                    ERROR   77              ;MSG B0 ERROR
4609  013572  104100                    ERROR   100             ;MSG A1 ERROR
4610  013574  104101                    ERROR   101             ;MSG B1 ERROR
4611
4612  013576  013737  001370  001372    MOV     HZ,COUNT
4613  013604  012737  000004  001374    MOV     #4,SEC
4614  013612  004737  047304            JSR     PC,CLKON        ;TURN CLK INTR ON FOR 4 SEC MAX
4615  013616  012765  000001  000026  8$:   MOV     #1,RKMR1(R5)    ;WORD 1
4616  013624  004737  045150            JSR     PC,GSTAT
4617  013630  032737  002000  003362    BIT     #D.FWD,HMR2
4618  013636  001004                    BNE     9$
4619  013640  005737  001376            TST     TIMUP
4620  013644  001764                    BEQ     8$
4621  013646  104102                    ERROR   102             ;FWD NOT DETECTED WITHIN 4 SEC IN RTZ PORTION OF
4622                                                            ;START SPINDLE CMD.
4623  013650  004737  047400      9$:   JSR     PC,CLKOF        ;TURN CLOCK INTR OFF.
4624  013654  004737  044074            JSR     PC,TSTATN       ;TEST FOR ATTN
4625  013660  000401                    BR      18$
4626  013662  104321                    ERROR   321             ;UNEXP ATTN AFTER OUTER LIM TO CYL 0
4627  013664  012737  030140  003424  18$:   MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
4628  013672  005037  003426            CLR     E.B0
4629  013676  012737  023720  003430    MOV     #<D.RTZ!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4630  013704  012737  000001  003432    MOV     #1,E.B1
4631
4632  013712  004737  044334            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4633  013716  000000                    .WORD   0!0!0           ;& MSGS SPECIFIED HERE
```

```
4634  013720  104103                        ERROR   103             ;MSG A0 ERROR FROM OUT LIM TO CYL 0 DURING LOAD
4635  013722  104104                        ERROR   104             ;MSG B0 ERROR
4636  013724  104105                        ERROR   105             ;MSG A1 ERROR
4637  013726  104106                        ERROR   106             ;MSG B1 ERROR
4638  013730  013737  001370  001372        MOV     HZ,COUNT
4639  013736  012737  000001  001374        MOV     #1,SEC
4640  013744  004737  047304                JSR     PC,CLKON        ;TURN CLK INTR ON FOR 1 SEC MAX
4641  013750  005065  000026          10$:  CLR     RKMRI(R5)       ;WORD 0
4642  013754  004737  045150                JSR     PC,GSTAT
4643  013760  032737  000200  003362        BIT     #D.DRDY,HMR2    ;SEE IF DRIVE READY
4644  013766  001004                        BNE     11$
4645  013770  005737  001376                TST     TIMUP
4646  013774  001765                        BEQ     10$
4647  013776  104107                        ERROR   107             ;DRIVE READY NOT SET WITHIN 1 SEC FROM
4648                                                                ;FWD IN RTZ PORTION OF START SPIN CMD
4649  014000  004737  047400          11$:  JSR     PC,CLKOF        ;TURN CLOCKS OFF
4650  014004  004737  044074                JSR     PC,TSTATN       ;TEST ATTN
4651  014010  104067                        ERROR   67              ;NO ATTN AFTER START SPIN CMD
4652  014012  012737  014236  001176        MOV     #30$,$ESCAPE
4653  014020  000137  013110                JMP     2$              ;CHECK RKMR 2 & 3 WORDS 0 & 1
4654
4655  014024  005037  001176          12$:  CLR     $ESCAPE
4656  014030  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
4657  014036  005065  000026                CLR     RKMRI(R5)
4658  014042  004737  045150                JSR     PC,GSTAT
4659  014046  032737  010000  003362        BIT     #D.SPIN,HMR2
4660  014054  001003                        BNE     14$
4661  014056  104306                        ERROR   306             ;SPIN NOT SET AFTER ST. SPIN CMD
4662  014060  000137  043076                JMP     $EOP            ;ABORT DRIVE
4663  014064  005037  003316          14$:  CLR     UNLD            ;USED FOR VALID HALT
4664  014070  004737  047526                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
4665  014074  000472                        BR      TST15           ;;GO TO NEXT TEST
4666                                                                ;RETURN HERE IF SW 14 IS SET OR
4667                                                                ;SW 8 WITH SWR <7:0> APPLY
4668  014076  005237  003316          20$:  INC     UNLD
4669
4670  014102  004737  045522                JSR     PC,SUBCLR
4671  014106  104024                        ERROR   24              ;CERR AFTER SCLR
4672
4673  014110  012765  000007  000000        MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
4674  014116  013737  001414  003372        MOV     T10,TEMP1
4675  014124  004737  043612                JSR     PC,FRDY         ;FIND RDY
4676  014130  104011                        ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
4677  014132  004737  044074                JSR     PC,TSTATN
4678  014136  104012                        ERROR   12              ;NO ATTN AFTER UNLOAD CMD
4679
4680  014140  004737  045522                JSR     PC,SUBCLR
4681  014144  104024                        ERROR   24              ;CERR AFTER SCLR
4682
4683  014146  013737  001414  003374        MOV     T10,TEMP2
4684  014154  004737  046556                JSR     PC,FSPOK
4685  014160  104315                        ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
4686
4687  014162  004737  045522                JSR     PC,SUBCLR
4688  014166  104024                        ERROR   24              ;CERR AFTER SCLR
4689
```

```
 4690  014170  005037  001176              CLR     $ESCAPE
 4691  014174  005737  001410              TST     LPFLG
 4692  014200  001402                      BEQ     67$
 4693  014202  000177  164702              JMP     @$LPERR         ;SW 9 WAS SET.
 4694  014206  000177  164674      67$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 4695
 4696
 4697
 4698  014212  004737  047400      25$:    JSR     PC,CLKOF
 4699  014216  005237  001410              INC     LPFLG
 4700  014222  032777  001000  164710      BIT     #SW9,@SWR       ;LOOP ON ERROR?
 4701  014230  001322                      BNE     20$             ;YES, RECONDITION DRIVE
 4702  014232  000137  013060              JMP     1$              ;RETURN TO MAINLINE
 4703  014236  004737  047400      30$:    JSR     PC,CLKOF
 4704  014242  005237  001410              INC     LPFLG
 4705  014246  032777  001000  164664      BIT     #SW9,@SWR       ;LOOP ON ERROR?
 4706  014254  001310                      BNE     20$             ;YES, RECONDITION DRIVE
 4707  014256  000137  014024              JMP     12$             ;RETURN TO MAINLINE
 4708                          .SBTTL SEEK/READ HEADER/WRITE HEADER TESTS
 4709
 4710                          ;;*****************************************************************
 4711                          ;*TEST 15       STATIC CYL DIFF AND CYL ADDR REG TEST; PART 1
 4712                          ;*
 4713                          ;*      THIS TEST CHECKS EACH BIT OF THE CYL DIFFERENCE
 4714                          ;*      AND CYL ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
 4715                          ;*      MAJOR CYLS (0,1,2,4,8,16,32,64,128,256) WITH EVEN PARITY SET.
 4716                          ;*      THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKING.
 4717                          ;*      THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
 4718                          ;*      NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.
 4719                          ;*
 4720                          ;;*****************************************************************
 4721  014262  000004          TST15:  SCOPE
 4722  014264  012737  000001  001174      MOV     #1,$TIMES       ;;DO 1 ITERATION
 4723  014272  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
 4724
 4725  014276  005000                      CLR     R0              ;CYL # REGISTER
 4726  014300  012737  100000  003402      MOV     #BIT15,TEMP5
 4727
 4728  014306                      1$:
 4729  014306  104415                      SCOP1
 4730  014310  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
 4731
 4732  014314  004737  045522              JSR     PC,SUBCLR
 4733  014320  104024                      ERROR   24              ;CERR AFTER SCLR
 4734
 4735
 4736  014322  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
 4737  014330  013765  001222  000010      MOV     $UNIT,RKCS2(R5)
 4738  014336  012765  000013  000000      MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
 4739                                                              ;RESET CYL DIFF/OFFSET & CYL ADDR REG
 4740                                                              ;IN RKMR2 & RKMR3 RESP.
 4741  014344  013737  001414  003372      MOV     T10,TEMP1
 4742  014352  004737  043612              JSR     PC,FRDY         ;FIND RDY
 4743  014356  104124                      ERROR   124             ;RDY NOT SET AFTER RECAL CMD
 4744
 4745  014360  012765  000001  000026      MOV     #1,RKMR1(R5)    ;SELECT WORD 1
```

```
4746  014366  004737  045150                    JSR     PC,GSTAT
4747  014372  032737  020000  003362            BIT     #D.RTZ,HMR2
4748  014400  001001                            BNE     64$
4749  014402  104244                            ERROR   244             ;RTZ NOT SET DURING RECAL CMD
4750  014404  013737  001414  003374  64$:      MOV     T10,TEMP2       ;SETUP TIMEOUT
4751  014412  004737  044126                    JSR     PC,FATT1        ;FIND ATTN
4752  014416  104055                            ERROR   55              ;NO ATTN AFTER RECAL CMD
4753
4754  014420  012737  050340  003424            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4755  014426  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
4756  014432  012737  001720  003430            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4757  014440  012737  000001  003432            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4758  014446  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
4759  014452  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4760  014460  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4761
4762  014466  004737  044334                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4763  014472  000007                            .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
4764  014474  104221                            ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
4765  014476  104275                            ERROR   275             ;MSG B0 ERROR
4766  014500  104222                            ERROR   222             ;MSG A1 ERROR
4767  014502  104276                            ERROR   276             ;MSG B1 ERROR
4768
4769  014504  005737  001362                    TST     CYLDIF          ;SEE IF MSG A2=0
4770  014510  001401                            BEQ     65$             ;BR IF YES
4771  014512  104047                            ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
4772  014514  005737  001364  65$:              TST     CYLADD          ;SEE IF MSG B2=0
4773  014520  001401                            BEQ     66$             ;BR IF YES
4774  014522  104050                            ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
4775  014524                  66$:
4776
4777  014524  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
4778  014532  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
4779  014540  012765  000005  000000            MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4780  014546  013737  001414  003372            MOV     T10,TEMP1
4781  014554  004737  043612                    JSR     PC,FRDY         ;FIND RDY
4782  014560  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4783  014562  004737  044074                    JSR     PC,TSTATN       ;TEST FOR ATTN
4784  014566  000401                            BR      67$
4785  014570  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4786  014572                  67$:
4787
4788  014572  012737  010340  003424            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
4789  014600  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
4790  014604  012737  001720  003430            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4791  014612  012737  000001  003432            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4792  014620  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
4793  014624  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4794  014632  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4795
4796  014640  004737  044334                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4797  014644  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4798  014646  104273                            ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4799  014650  104265                            ERROR   265             ;MSG B0 ERROR
4800  014652  104274                            ERROR   274             ;MSG A1 ERROR
4801  014654  104266                            ERROR   266             ;MSG B1 ERROR
```

```
4802
4803
4804  014656  104415                  SCOP1
4805  014660  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
4806
4807  014664  004737  045522          JSR     PC,SUBCLR
4808  014670  104024                  ERROR   24              ;CERR AFTER SCLR
4809
4810  014672  005237  001462          INC     BYPCERR         ;DO NOT TEST CERR IN GSTAT1
4811  014676  012765  000020  000026  MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
4812  014704  010065  000020          MOV     R0,RKDC(R5)     ;CYL ADDR
4813  014710  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD.
4814  014716  013737  001414  003372  MOV     T10,TEMP1
4815  014724  004737  043612          JSR     PC,FRDY         ;FIND RDY
4816  014730  104122                  ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
4817  014732  004737  044074          JSR     PC,TSTATN       ;TEST FOR ATTN
4818  014736  104125                  ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
4819  014740  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
4820  014746  012737  001200  003426  MOV     #<D.FLT!D.PAR>,E.B0
4821  014754  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4822  014762  012737  000001  003432  MOV     #1,E.B1
4823  014770  010037  003434          MOV     R0,E.A2
4824  014774  010037  003436          MOV     R0,E.B2
4825  015000  052737  000002  003436  BIS     #2,E.B2         ;ADD MSG ID
4826
4827  015006  004737  044334          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4828  015012  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4829  015014  104110                  ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
4830  015016  104111                  ERROR   111             ;MSG B0 ERROR
4831  015020  104146                  ERROR   146             ;MSG A1 ERROR
4832  015022  104147                  ERROR   147             ;MSG B1 ERROR
4833
4834  015024  020037  001364          CMP     R0,CYLADD
4835  015030  001401                  BEQ     2$
4836  015032  104043                  ERROR   43              ;CYL ADDR IN B2 NOT=RKDC
4837
4838  015034  020037  001362  2$:     CMP     R0,CYLDIF
4839  015040  001401                  BEQ     3$
4840  015042  104044                  ERROR   44              ;CYL DIFF IN A2 NOT=RKDC
4841
4842  015044  005037  001462  3$:     CLR     BYPCERR         ;ALLOW CHECKING FOR ANY CERR IN GSTAT1
4843  015050  006137  003402          ROL     TEMP5           ;SET CARRY ONLY ONCE
4844  015054  006100                  ROL     R0              ;SELECT NEXT MAJOR CYL
4845  015056  020027  001000          CMP     R0,#1000        ;ALL MAJOR CYL DONE?
4846  015062  001001                  BNE     4$              ;BRANCH IF NO
4847  015064  000402                  BR      TST16           ;;GO TO  NEXT TST
4848  015066  000137  014306  4$:     JMP     1$
4849
4850                          ;;*****************************************************************
4851                          ;*TEST 16       STATIC CYL DIFF & CYL ADDR REG TEST-PART 2
4852                          ;*
4853                          ;*      THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
4854                          ;*      DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
4855                          ;*      ALL CYLS FROM EVERY OTHER CYL. (N SQUARE SEEKS).
4856                          ;*      IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.
4857                          ;*
```

```
4858                                    ;;*****************************************************************
4859  015072  000004                    TST16:  SCOPE
4860  015074  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
4861  015102  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
4862
4863  015106  005737  001340                    TST     BYPT16
4864  015112  001404                            BEQ     13$
4865  015114  104401  056002                    TYPE    ,MSG9           ;BYPASSING TEST 16
4866  015120  000137  016362                    JMP     12$
4867
4868  015124  104401  055744            13$:    TYPE    ,MSG8           ;PLEASE WAIT, LONG TEST
4869
4870  015130  005037  001350                    CLR     FRCYL           ;FROM CYL
4871  015134  005037  001352                    CLR     TOCYL           ;TO CYL
4872  015140  005037  001354                    CLR     CCYL            ;CURRENT CYL
4873  015144  005037  001356                    CLR     PCYL            ;PREV CYL
4874
4875  015150  104415                            SCOP1
4876  015152  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
4877
4878  015156  004737  045522                    JSR     PC,SUBCLR
4879  015162  104024                            ERROR   24              ;CERR AFTER SCLR
4880
4881
4882  015164  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
4883  015172  013765  001222  000010            MOV     $UNIT,RKCS2(R5)
4884  015200  012765  000013  000000            MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
4885                                                                    ;RESET CYL DIFF/OFFSET & CYL ADDR REG
4886                                                                    ;IN RKMR2 & RKMR3 RESP.
4887  015206  013737  001414  003372            MOV     T10,TEMP1
4888  015214  004737  043612                    JSR     PC,FRDY         ;FIND RDY
4889  015220  104124                            ERROR   124             ;RDY NOT SET AFTER RECAL CMD
4890
4891  015222  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4892  015230  004737  045150                    JSR     PC,GSTAT
4893  015234  032737  020000  003362            BIT     #D.RTZ,HMR2
4894  015242  001001                            BNE     64$
4895  015244  104244                            ERROR   244             ;RTZ NOT SET DURING RECAL CMD
4896  015246  013737  001414  003374    64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
4897  015254  004737  044126                    JSR     PC,FATT1        ;FIND ATTN
4898  015260  104055                            ERROR   55              ;NO ATTN AFTER RECAL CMD
4899
4900  015262  012737  050340  003424            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4901  015270  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
4902  015274  012737  001720  003430            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4903  015302  012737  000001  003432            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4904  015310  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
4905  015314  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4906  015322  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4907
4908  015330  004737  044334                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4909  015334  000007                            .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
4910  015336  104221                            ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
4911  015340  104275                            ERROR   275             ;MSG B0 ERROR
4912  015342  104222                            ERROR   222             ;MSG A1 ERROR
4913  015344  104276                            ERROR   276             ;MSG B1 ERROR
```

```
4914
4915  015346  005737  001362            TST     CYLDIF          ;SEE IF MSG A2=0
4916  015352  001401                    BEQ     65$             ;BR IF YES
4917  015354  104047                    ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
4918  015356  005737  001364    65$:    TST     CYLADD          ;SEE IF MSG B2=0
4919  015362  001401                    BEQ     66$             ;BR IF YES
4920  015364  104050                    ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
4921  015366                    66$:
4922
4923  015366  012765  100000  000000    MOV     #CCLR,RKCS1(R5)
4924  015374  013765  001222  000010    MOV     $UNIT,RKCS2(R5) ;DRIVE#
4925  015402  012765  000005  000000    MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4926  015410  013737  001414  003372    MOV     T10,TEMP1
4927  015416  004737  043612            JSR     PC,FRDY         ;FIND RDY
4928  015422  104151                    ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4929  015424  004737  044074            JSR     PC,TSTATN       ;TEST FOR ATTN
4930  015430  000401                    BR      67$
4931  015432  104154                    ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4932  015434                    67$:
4933
4934  015434  012737  010340  003424    MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
4935  015442  005037  003426            CLR     E.B0            ;EXPECTED MSG B0
4936  015446  012737  001720  003430    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4937  015454  012737  000001  003432    MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4938  015462  005037  003434            CLR     E.A2            ;EXPECTED MSG A2
4939  015466  012737  000002  003436    MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4940  015474  012737  000003  003442    MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4941
4942  015502  004737  044334            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4943  015506  000003                    .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4944  015510  104273                    ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4945  015512  104265                    ERROR   265             ;MSG B0 ERROR
4946  015514  104274                    ERROR   274             ;MSG A1 ERROR
4947  015516  104266                    ERROR   266             ;MSG B1 ERROR
4948
4949
4950  015520  104415                    SCOP1
4951  015522  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
4952
4953  015526  004737  045522            JSR     PC,SUBCLR
4954  015532  104024                    ERROR   24              ;CERR AFTER SCLR
4955
4956
4957  015534  012765  000020  000026  1$:     MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
4958  015542  013765  001352  000020    MOV     TOCYL,RKDC(R5)  ;SET TO CYL ADDR
4959  015550  013737  001352  001354    MOV     TOCYL,CCYL      ;CURRENT CYL
4960  015556  013737  001354  003376    MOV     CCYL,TEMP3
4961  015564  013737  001356  003400    MOV     PCYL,TEMP4      ;PREV CYL
4962  015572  163737  003376  003400    SUB     TEMP3,TEMP4
4963  015600  100002                    BPL     2$              ;BR IF TEMP4 IS POS
4964  015602  005437  003400            NEG     TEMP4
4965  015606  013737  003400  001360  2$:     MOV     TEMP4,CALDIF
4966  015614  013737  001354  001356    MOV     CCYL,PCYL
4967  015622  012765  000017  000000    MOV     #SEEK,RKCS1(R5) ;SEEK CMD.
4968  015630  013737  001414  003372    MOV     T10,TEMP1
4969  015636  004737  043612            JSR     PC,FRDY         ;FIND RDY
```

```
 4970  015642  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
 4971  015644  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4972  015650  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
 4973  015652  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
 4974  015660  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
 4975  015666  004737  046112                  JSR     PC,RDCYLA       ;READ CYL ADDR
 4976  015672  023737  001352  001364          CMP     TOCYL,CYLADD    ;SEE IF TO CYL ECHOED OK
 4977  015700  001401                          BEQ     3$
 4978  015702  104045                          ERROR   45              ;MR3 NOT=RKDC
 4979
 4980  015704  004737  046026          3$:     JSR     PC,RDCYLD       ;READ CYL DIFF
 4981  015710  023737  001360  001362          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF CORRECT
 4982  015716  001401                          BEQ     4$
 4983  015720  104046                          ERROR   46              ;CYL DIFF IN RKMR2 INCORRECT
 4984
 4985  015722                          4$:
 4986
 4987  015722  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4988  015730  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4989  015736  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 4990  015744  013737  001414  003372          MOV     T10,TEMP1
 4991  015752  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 4992  015756  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4993  015760  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4994  015764  000401                          BR      68$
 4995  015766  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4996  015770                          68$:
 4997
 4998
 4999  015770  104415                  SCOP1
 5000  015772  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 5001
 5002  015776  004737  045522                  JSR     PC,SUBCLR
 5003  016002  104024                          ERROR   24              ;CERR AFTER SCLR
 5004
 5005  016004  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
 5006  016012  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;SET RETURN CYL ADDR
 5007  016020  013737  001350  001354          MOV     FRCYL,CCYL
 5008  016026  013737  001354  003376          MOV     CCYL,TEMP3
 5009  016034  013737  001356  003400          MOV     PCYL,TEMP4
 5010  016042  163737  003376  003400          SUB     TEMP3,TEMP4
 5011  016050  100002                          BPL     5$              ;BR IF TEMP4 IS POS
 5012  016052  005437  003400                  NEG     TEMP4
 5013  016056  013737  003400  001360  5$:     MOV     TEMP4,CALDIF
 5014  016064  013737  001354  001356          MOV     CCYL,PCYL
 5015  016072  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
 5016  016100  013737  001414  003372          MOV     T10,TEMP1
 5017  016106  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 5018  016112  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
 5019  016114  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
 5020  016120  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
 5021  016122  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
 5022  016130  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
 5023  016136  004737  046112                  JSR     PC,RDCYLA       ;READ CYL ADDR
 5024  016142  023737  001350  001364          CMP     FRCYL,CYLADD    ;SEE IF RETURN CYL ECHOED OK
 5025  016150  001401                          BEQ     6$
```

```
5026  016152  104241                          ERROR   241             ;MR3 NOT=RKDC
5027
5028  016154  023737  001352  001350  6$:     CMP     TOCYL,FRCYL     ;SEE IF TO=FROM
5029  016162  001022                          BNE     10$             ;DO NORMAL TEST IF NO
5030  016164  005737  001352                  TST     TOCYL           ;SEE IF=0
5031  016170  001007                          BNE     9$
5032  016172  004737  046026                  JSR     PC,RDCYLD       ;CYL DIFF S/B 0 ON CYL 0
5033  016176  005737  001362                  TST     CYLDIF
5034  016202  001421                          BEQ     7$
5035  016204  104242                          ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
5036  016206  000417                          BR      7$
5037
5038  016210  004737  046026          9$:     JSR     PC,RDCYLD       ;CYL DIFF/OFFSET SHOULD NOT
5039  016214  023727  001362  000001          CMP     CYLDIF,#1       ;CHANGE IN SEEK TO SELF
5040  016222  001411                          BEQ     7$              ;SHOULD = 1 IN THIS TEST
5041  016224  104263                          ERROR   263             ;CYL DIFF IN RKMR2 DID NOT REMAIN = 0
5042  016226  000407                          BR      7$
5043  016230  004737  046026          10$:    JSR     PC,RDCYLD       ;READ CYL DIFF
5044  016234  023737  001360  001362          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF OK
5045  016242  001401                          BEQ     7$
5046  016244  104242                          ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
5047
5048  016246                          7$:
5049
5050  016246  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5051  016254  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5052  016262  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5053  016270  013737  001414  003372          MOV     T10,TEMP1
5054  016276  004737  043612                  JSR     PC,FRDY         ;FIND RDY
5055  016302  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5056  016304  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
5057  016310  000401                          BR      69$
5058  016312  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5059  016314                          69$:
5060
5061
5062  016314  005237  001352                  INC     TOCYL
5063  016320  023727  001352  000633          CMP     TOCYL,#411.     ;SEE IF SCANNED ALL CYLS
5064  016326  001402                          BEQ     8$              ;BR IF YES
5065  016330  000137  015534                  JMP     1$              ;ELSE REPEAT
5066
5067  016334  005237  001350          8$:     INC     FRCYL
5068  016340  023727  001350  000633          CMP     FRCYL,#411.     ;SEE IF ALL DONE
5069  016346  001405                          BEQ     TST17           ;;GO TO NEXT TST
5070  016350  013737  001350  001352          MOV     FRCYL,TOCYL     ;FRCYL ALWAYS = OR > TOCYL
5071  016356  000137  015534                  JMP     1$              ;ELSE REPEAT
5072  016362                          12$:
5073
5074
5075                                  ;;*****************************************************************
5076                                  ;*TEST 17       HEAD REGISTER TEST
5077                                  ;*
5078                                  ;*      THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
5079                                  ;*      VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
5080                                  ;*      HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.
5081                                  ;*
```

```
                                ;*      SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK CMDS,
                                ;*      SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
                                ;*      ILLEGAL ADDRESS. IF NOT,THIS MEANS THAT CHANGING HEAD ADDRESSES
                                ;*      ARE NOT TIED TO SEEK CMDS
                                ;*
                                ;**********************************************************************
016362  000004                TST17:  SCOPE
016364  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
016372  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

016376  005000                        CLR     R0              ;HEAD #
016400                        1$:
016400  104415                        SCOP1
016402  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

016406  004737  045522                JSR     PC,SUBCLR
016412  104024                        ERROR   24              ;CERR AFTER SCLR


016414  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
016422  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
016430  012765  000013  000000        MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                                                ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                                                ;IN RKMR2 & RKMR3 RESP.

016436  013737  001414  003372        MOV     T10,TEMP1
016444  004737  043612                JSR     PC,FRDY         ;FIND RDY
016450  104124                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD

016452  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
016460  004737  045150                JSR     PC,GSTAT
016464  032737  020000  003362        BIT     #D.RTZ,HMR2
016472  001001                        BNE     64$
016474  104244                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
016476  013737  001414  003374  64$:  MOV     T10,TEMP2       ;SETUP TIMEOUT
016504  004737  044126                JSR     PC,FATT1        ;FIND ATTN
016510  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD

016512  012737  050340  003424        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
016520  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
016524  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
016532  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
016540  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
016544  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
016552  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

016560  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
016564  000007                        .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
016566  104221                        ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
016570  104275                        ERROR   275             ;MSG B0 ERROR
016572  104222                        ERROR   222             ;MSG A1 ERROR
016574  104276                        ERROR   276             ;MSG B1 ERROR

016576  005737  001362                TST     CYLDIF          ;SEE IF MSG A2=0
016602  001401                        BEQ     65$             ;BR IF YES
016604  104047                        ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
```

```
5138  016606  005737  001364          65$:    TST     CYLADD          ;SEE IF MSG B2=0
5139  016612  001401                          BEQ     66$             ;BR IF YES
5140  016614  104050                          ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
5141  016616                          66$:
5142
5143  016616  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5144  016624  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5145  016632  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5146  016640  013737  001414  003372          MOV     T10,TEMP1
5147  016646  004737  043612                  JSR     PC,FRDY         ;FIND RDY
5148  016652  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5149  016654  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
5150  016660  000401                          BR      67$
5151  016662  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5152  016664                          67$:
5153
5154  016664  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5155  016672  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
5156  016676  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5157  016704  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5158  016712  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5159  016716  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5160  016724  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5161
5162  016732  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5163  016736  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5164  016740  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5165  016742  104265                          ERROR   265             ;MSG B0 ERROR
5166  016744  104274                          ERROR   274             ;MSG A1 ERROR
5167  016746  104266                          ERROR   266             ;MSG B1 ERROR
5168
5169
5170  016750  023727  001432  000001          CMP     HEADA,#1        ;FOR HEAD 0, B3=1
5171  016756  001401                          BEQ     3$
5172  016760  104053                          ERROR   53              ;RECAL DID NOT RESET HEAD REG IN B3.
5173
5174  016762                          3$:
5175  016762  104415                          SCOP1
5176  016764  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5177
5178  016770  004737  045522                  JSR     PC,SUBCLR
5179  016774  104024                          ERROR   24              ;CERR AFTER SCLR
5180
5181  016776  000300                          SWAB    R0
5182  017000  010065  000006                  MOV     R0,RKDA(R5)     ;HEAD #
5183  017004  000300                          SWAB    R0
5184
5185  017006  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
5186  017014  013737  001414  003372          MOV     T10,TEMP1
5187  017022  004737  043612                  JSR     PC,FRDY         ;FIND RDY
5188  017026  104156                          ERROR   156             ;NO RDY AFTER SEEK TO SELF
5189  017030  004737  044074                  JSR     PC,TSTATN
5190  017034  104157                          ERROR   157             ;NO ATTN AFTER SEEK TO SELF
5191  017036  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
5192  017044  020027  000003                  CMP     R0,#3
5193  017050  001403                          BEQ     4$              ;BR FOR HEAD 3
```

```
5194  017052  005037  003426                CLR     E.B0            ;FOR HEADS 0,1,2
5195  017056  000403                        BR      5$
5196  017060  012737  002240  003426  4$:   MOV     #<D.SKI!D.FLT!D.IDAE>,E.B0      ;FOR HEAD 3
5197  017066  012737  001720  003430  5$:   MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5198  017074  012737  000001  003432        MOV     #1,E.B1
5199  017102  005037  003434                CLR     E.A2
5200  017106  012737  000002  003436        MOV     #2,E.B2
5201  017114  005700                        TST     R0              ;SEE IF HEAD 0
5202  017116  001004                        BNE     6$              ;BR IF NO
5203  017120  012737  001003  003442        MOV     #<BIT9!3>,E.B3  ;LOAD EXPECTED B3 FOR HEAD 0 & MSG ID
5204  017126  000412                        BR      8$
5205  017130  020027  000001          6$:   CMP     R0,#1           ;SEE IF HEAD 1
5206  017134  001004                        BNE     7$              ;BR IF NO
5207  017136  012737  002003  003442        MOV     #<BIT10!3>,E.B3 ;B3 FOR HEAD 1
5208  017144  000403                        BR      8$
5209  017146  012737  004003  003442  7$:   MOV     #<BIT11!3>,E.B3 ;B3 FOR HEAD 2
5210  017154                          8$:
5211
5212  017154  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5213  017160  000007                        .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
5214  017162  104114                        ERROR   114             ;MSG A0 ERROR AFTER LOAD HEAD REG & SEEK CMD
5215  017164  104115                        ERROR   115             ;MSG B0 ERROR
5216  017166  104322                        ERROR   322             ;MSG A1 ERROR
5217  017170  104323                        ERROR   323             ;MSG B1 ERROR
5218
5219  017172  005737  001362                TST     CYLDIF          ;SEE IF MSG A2=0
5220  017176  001401                        BEQ     68$             ;BR IF YES
5221  017200  104324                        ERROR   324             ;MSG A2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
5222  017202  005737  001364          68$:  TST     CYLADD          ;SEE IF MSG B2=0
5223  017206  001401                        BEQ     69$             ;BR IF YES
5224  017210  104325                        ERROR   325             ;MSG B2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
5225  017212                          69$:
5226
5227  017212  020027  000003                CMP     R0,#3
5228  017216  001412                        BEQ     9$              ;BR IF HEAD 3
5229
5230  017220  005037  003372                CLR     TEMP1
5231  017224  116037  003324  003372        MOVB    ATTN(R0),TEMP1
5232  017232  023737  003372  001432        CMP     TEMP1,HEADA     ;FOR RKDA=HEAD 0, HEAD=1 IN B3
5233                                                                ;FOR RKDA=HEAD 1, HEAD=2 IN B3
5234                                                                ;FOR RKDA=HEAD 2, HEAD=4 IN B3
5235  017240  001401                        BEQ     9$
5236  017242  104054                        ERROR   54              ;HEAD DECODE IN B3 INCORRECT
5237
5238
5239  017244  005200                  9$:   INC     R0
5240  017246  020027  000004                CMP     R0,#4           ;0 THRU 3 DONE?
5241  017252  001402                        BEQ     10$             ;BR IF YES
5242  017254  000137  016400                JMP     1$              ;ELSE REPEAT
5243
5244  017260                          10$:
5245
5246  017260  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
5247  017266  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
5248  017274  012765  000013  000000        MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
5249  017302  013737  001414  003372        MOV     T10,TEMP1
```

```
5250  017310  004737  043612          JSR     PC,FRDY         ;FIND RDY
5251  017314  104124                  ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
5252
5253  017316  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5254  017324  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5255  017332  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5256  017340  013737  001414  003372  MOV     T10,TEMP1
5257  017346  004737  043612          JSR     PC,FRDY         ;FIND RDY
5258  017352  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5259  017354  004737  044074          JSR     PC,TSTATN       ;TEST FOR ATTN
5260  017360  000401                  BR      71$
5261  017362  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5262  017364                  71$:
5263
5264
5265  017364  004737  045150          JSR     PC,GSTAT
5266  017370  032737  000040  003364  BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
5267  017376  001401                  BEQ     70$             ;BR IF YES
5268  017400  104155                  ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
5269
5270  017402  012765  100000  000000  70$:    MOV     #CCLR,RKCS1(R5)
5271  017410  013737  001412  003374  MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
5272  017416  004737  044126          JSR     PC,FATT1
5273  017422  104055                  ERROR   55              ;NO ATTN AFTER RECAL CMD
5274
5275
5276                          ;;*****************************************************************
5277                          ;*TEST 20       SEEK TO CYL 0
5278                          ;*
5279                          ;*      TESTS THE ABILITY TO DO A SEEK CMD.
5280                          ;*      VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
5281                          ;*      STATUS BITS.  VERIFIES CMD COMPLETION BETWEEN 10-15USEC.
5282                          ;*      READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.
5283                          ;*
5284                          ;;*****************************************************************
5285  017424  000004          TST20:  SCOPE
5286  017426  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
5287  017434  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
5288
5289  017440  004737  045522          JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
5290  017444  104024                  ERROR   24              ;CERR AFTER SCLR
5291  017446  004737  046112          JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
5292  017452  005737  001364          TST     CYLADD
5293  017456  001401                  BEQ     1$
5294  017460  104130                  ERROR   130             ;CYL ADDR NOT CLEARED AFTER SCLR
5295  017462                  1$:
5296  017462  104415                  SCOP1
5297  017464  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
5298
5299  017470  004737  045522          JSR     PC,SUBCLR
5300  017474  104024                  ERROR   24              ;CERR AFTER SCLR
5301
5302  017476  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD: SEEK TO SELF
5303  017504  012737  000005  003372  MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
5304  017512  004737  043612          JSR     PC,FRDY         ;FIND RDY & GET STATUS
5305  017516  104131                  ERROR   131             ;RDY NOT SET AFTER SEEK CMD
```

```
5306  017520  012737  000005  003372          MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
5307
5308  017526  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
5309  017532  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
5310  017534  032737  100000  003334          BIT     #CERR,HCS1
5311  017542  001401                          BEQ     64$
5312  017544  104210                          ERROR   210             ;CERR AFTER SEEK CMD
5313  017546                          64$:
5314
5315  017546  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5316  017554  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
5317  017560  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5318  017566  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5319  017574  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5320  017600  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5321  017606  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5322
5323  017614  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5324  017620  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5325  017622  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
5326  017624  104134                          ERROR   134             ;MSG B0 ERROR
5327  017626  104135                          ERROR   135             ;MSG A1 ERROR
5328  017630  104136                          ERROR   136             ;MSG B1 ERROR
5329  017632  005737  001362                  TST     CYLDIF
5330  017636  001401                          BEQ     65$
5331  017640  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
5332
5333  017642                          65$:
5334
5335  017642  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5336  017650  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5337  017656  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5338  017664  013737  001414  003372          MOV     T10,TEMP1
5339  017672  004737  043612                  JSR     PC,FRDY         ;FIND RDY
5340  017676  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5341  017700  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
5342  017704  000401                          BR      66$
5343  017706  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5344  017710                          66$:
5345
5346  017710  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5347  017716  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
5348  017722  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5349  017730  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5350  017736  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5351  017742  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5352  017750  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5353
5354  017756  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5355  017762  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5356  017764  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5357  017766  104265                          ERROR   265             ;MSG B0 ERROR
5358  017770  104274                          ERROR   274             ;MSG A1 ERROR
5359  017772  104266                          ERROR   266             ;MSG B1 ERROR
5360
5361  017774  005737  001364                  TST     CYLADD
```

```
5362  020000  001401                        BEQ     TST21           ;GO TO NEXT TEST
5363  020002  104140                        ERROR   140             ;CYL ADDR IN B2 NOT CLEARED AFT SEEK CMD.
5364
5365                                ;;*****************************************************************
5366                                ;*TEST 21       TEST SECTOR COUNT REG. IN MSG B3
5367                                ;;*****************************************************************
5368  020004  000004                TST21:  SCOPE
5369  020006  012737  000001  001174        MOV     #1,$TIMES       ;DO 1 ITERATION
5370  020014  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5371
5372  020020  004737  045522                JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
5373  020024  104024                        ERROR   24              ;CERR AFTER SCLR
5374  020026  012737  020156  001176        MOV     #2$,$ESCAPE     ;GO TO NEXT TEST IF ANY ERROR DETECTED
5375
5376  020034  012737  000025  001400        MOV     #21.,SECNT      ;22 SECTOR FORMAT TEST
5377
5378  020042  004737  045652                JSR     PC,FS022                ;FIND SECTOR 0
5379  020046  104142                        ERROR   142             ;SECTOR 0 NOT FOUND BY TIMEOUT
5380
5381  020050  005037  001402                CLR PSEC                ;PREVIOUS SECTOR
5382  020054  004737  045736        64$:    JSR     PC,FNS22                ;FIND NEXT SECTOR
5383  020060  104143                        ERROR   143             ;DIFFERENT SECTOR, NOT FOUND BY TIMEOUT
5384  020062  013737  001402  001404        MOV     PSEC,ESEC
5385  020070  062737  000001  001404        ADD     #1,ESEC         ;SETUP EXPECTED SECTOR
5386  020076  013737  001406  001402        MOV     SECTOR,PSEC     ;UPDATE PREV SECTOR
5387  020104  004737  045602                JSR     PC,RDSEC                ;READ SECTOR
5388  020110  023737  001406  001402        CMP     SECTOR,PSEC
5389  020116  001407                        BEQ     65$             ;BR IF READ SAME TWICE
5390  020120  004737  045602                JSR     PC,RDSEC
5391  020124  023737  001406  001402        CMP     SECTOR,PSEC
5392  020132  001401                        BEQ     65$             ;TRY 1 MORE TIME
5393  020134  104144                        ERROR   144             ;MSG B3 ERROR, SECTOR REG UNSTABLE
5394                                                                ;MAY BE DURING SECTOR PULSE TIME
5395  020136  023737  001406  001404  65$:   CMP     SECTOR,ESEC
5396  020144  001401                        BEQ     66$
5397  020146  104145                        ERROR   145             ;MSG B3 ERROR BETWEEN SECTOR COUNTS
5398  020150  005337  001400        66$:    DEC     SECNT
5399  020154  001337                        BNE     64$             ;BR IF SECTOR COUNT NOT DONE
5400
5401
5402  020156  005037  001176        2$:     CLR     $ESCAPE
5403
5404                                ;;*****************************************************************
5405                                ;*TEST 22       DETECT OUTER LIMIT
5406                                ;*
5407                                ;*      THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYL 0
5408                                ;*      BY DETECTING OUTER LIMIT AS THE ADJACENT CYL.
5409                                ;*      AN ERROR IN THIS TEST INDICATES:
5410                                ;*
5411                                ;*              A.  HEADS WERE NOT ON CYL 0
5412                                ;*      AND/OR  B.  COULD NOT SEEK IN REVERSE DIRECTION.
5413                                ;*
5414                                ;;*****************************************************************
5415  020162  000004                TST22:  SCOPE
5416  020164  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
5417  020172  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
```

```
5418
5419  020176  004737  045522          JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
5420  020202  104024                  ERROR   24              ;CERR AFTER SCLR
5421
5422  020204  005037  001410          CLR     LPFLG
5423  020210  005237  001462          INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
5424  020214  005237  003316          INC     UNLD            ;USED FOR VALID HALT
5425
5426  020220  012765  000020  000026  MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
5427  020226  012765  000001  000020  MOV     #1,RKDC(R5)     ;CYL 1
5428  020234  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD
5429  020242  013737  001414  003372  MOV     T10,TEMP1
5430  020250  004737  043612          JSR     PC,FRDY         ;FIND RDY
5431  020254  104122                  ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
5432  020256  004737  044074          JSR     PC,TSTATN       ;TEST FOR ATTN
5433  020262  104125                  ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
5434  020264  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
5435  020272  012737  001200  003426  MOV     #<D.FLT!D.PAR>,E.B0
5436  020300  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5437  020306  012737  000001  003432  MOV     #1,E.B1
5438
5439  020314  004737  044334          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5440  020320  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5441  020322  104110                  ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
5442  020324  104111                  ERROR   111             ;MSG B0 ERROR
5443  020326  104146                  ERROR   146             ;MSG A1 ERROR
5444  020330  104147                  ERROR   147             ;MSG B1 ERROR
5445
5446  020332  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5447  020340  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5448  020346  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5449  020354  013737  001414  003372  MOV     T10,TEMP1
5450  020362  004737  043612          JSR     PC,FRDY         ;FIND RDY
5451  020366  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5452  020370  004737  044074          JSR     PC,TSTATN       ;TEST FOR ATTN
5453  020374  000401                  BR      64$
5454  020376  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5455  020400                  64$:
5456
5457  020400  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5458  020406  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5459  020412  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5460  020420  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5461  020426  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5462  020432  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5463  020440  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5464
5465  020446  004737  044334          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5466  020452  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5467  020454  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5468  020456  104265                  ERROR   265             ;MSG B0 ERROR
5469  020460  104274                  ERROR   274             ;MSG A1 ERROR
5470  020462  104266                  ERROR   266             ;MSG B1 ERROR
5471
5472
5473  020464  012765  000000  000020  MOV     #0,RKDC(R5)     ;CYL 0
```

```
5474  020472  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL 0
5475  020500  013737  001414  003372          MOV     T10,TEMP1
5476  020506  004737  043612                  JSR     PC,FRDY         ;FIND RDY
5477  020512  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
5478  020514  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5479  020522  004737  045150                  JSR     PC,GSTAT
5480  020526  004737  046356                  JSR     PC,FLIM         ;FIND LIMIT DETECT
5481  020532  104160                          ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
5482
5483  020534  032737  040000  003362          BIT     #D.UNLD,HMR2
5484  020542  001003                          BNE     1$
5485  020544  104305                          ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
5486  020546  000137  021260                  JMP     30$             ;BYPASS REST OF TEST
5487
5488  020552  012737  021170  001176  1$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
5489  020560  012737  070140  003424          MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
5490  020566  012737  002200  003426          MOV     #<D.SKI!D.FLT>,E.B0
5491  020574  012737  045720  003430          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5492  020602  012737  030001  003432          MOV     #<D.LIMD!D.NMOV!1>,E.B1
5493
5494  020610  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5495  020614  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5496  020616  104161                          ERROR   161             ;MSG A0 ERROR AFTER OUTER LIMIT DETECT
5497  020620  104162                          ERROR   162             ;MSG B0 ERROR
5498  020622  104163                          ERROR   163             ;MSG A1 ERROR
5499  020624  104164                          ERROR   164             ;MSG B1 ERROR
5500
5501  020626  004737  044074                  JSR     PC,TSTATN
5502  020632  104165                          ERROR   165             ;NO ATTN AFTER OUTER LIMIT DETECT
5503  020634  005037  001462                  CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1
5504
5505  020640  004737  045522                  JSR     PC,SUBCLR       ;SUBSYS CLR
5506  020644  104024                          ERROR   24              ;CERR AFTER SCLR
5507  020646  013737  001414  003374          MOV     T10,TEMP2       ;SET UP TIMEOUT
5508  020654  004737  046434                  JSR     PC,FHDHM        ;FIND HEAD HOME
5509  020660  104166                          ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
5510  020662  004737  046510                  JSR     PC,FLOAD        ;FIND LOAD HEADS
5511  020666  104167                          ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
5512  020670  013737  001416  003374          MOV     T100,TEMP2      ;SETUP TIMEOUT
5513  020676  004737  044126                  JSR     PC,FATT1        ;FIND ATTN
5514  020702  104067                          ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
5515  020704  005037  001176          2$:     CLR     $ESCAPE
5516  020710  005037  003316                  CLR     UNLD            ;CLEAR FLAG
5517
5518  020714  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5519  020722  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
5520  020726  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5521  020734  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5522  020742  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5523  020746  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5524  020754  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5525
5526  020762  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5527  020766  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5528  020770  104063                          ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOADING
5529  020772  104064                          ERROR   64              ;MSG B0 ERROR
```

```
5530  020774  104065                        ERROR   65              ;MSG A1 ERROR
5531  020776  104066                        ERROR   66              ;MSG B1 ERROR
5532
5533  021000  005737  001362                TST     CYLDIF          ;SEE IF MSG A2=0
5534  021004  001401                        BEQ     65$             ;BR IF YES
5535  021006  104175                        ERROR   175             ;MSG A2 NOT CLEARED AT END OF HEAD LOADING
5536  021010  005737  001364        65$:    TST     CYLADD          ;SEE IF MSG B2=0
5537  021014  001401                        BEQ     66$             ;BR IF YES
5538  021016  104176                        ERROR   176             ;MSG B2 NOT CLEARED AT END OF HEAD LOADING
5539  021020                        66$:
5540
5541  021020  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
5542  021026  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
5543  021034  012765  000005  000000        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5544  021042  013737  001414  003372        MOV     T10,TEMP1
5545  021050  004737  043612                JSR     PC,FRDY         ;FIND RDY
5546  021054  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5547  021056  004737  044074                JSR     PC,TSTATN       ;TEST FOR ATTN
5548  021062  000401                        BR      67$
5549  021064  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5550  021066                        67$:
5551
5552  021066  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5553  021074  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
5554  021100  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5555  021106  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5556  021114  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
5557  021120  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5558  021126  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5559
5560  021134  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5561  021140  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5562  021142  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5563  021144  104265                        ERROR   265             ;MSG B0 ERROR
5564  021146  104274                        ERROR   274             ;MSG A1 ERROR
5565  021150  104266                        ERROR   266             ;MSG B1 ERROR
5566
5567  021152  004737  047526                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5568  021156  000440                        BR      TST23           ;;GO TO NEXT TEST
5569                                                                ;RETURN HERE IF SW 14 IS SET OR
5570                                                                ;SW 8 WITH SWR <7:0> APPLY
5571
5572
5573  021160  005037  001176        10$:    CLR     $ESCAPE
5574  021164  000177  157716                JMP     @$LPADR
5575  021170                        20$:
5576
5577  021170  004737  045522                JSR     PC,SUBCLR
5578  021174  104024                        ERROR   24              ;CERR AFTER SCLR
5579
5580  021176  012765  000011  000000        MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
5581  021204  013737  001414  003372        MOV     T10,TEMP1       ;SET TIMEOUT
5582  021212  004737  043612                JSR     PC,FRDY         ;FIND RDY
5583  021216  104121                        ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
5584
5585  021220  013737  001420  003374        MOV     T500,TEMP2      ;SETUP  TIMEOUT
```

```
5586  021226  004737  044126              JSR     PC,FATT1        ;FIND ATTN
5587  021232  104067                      ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
5588
5589  021234  005037  003316              CLR     UNLD
5590  021240  005237  001410              INC     LPFLG
5591  021244  032777  001000  157666      BIT     #SW9,@SWR       ;LOOP ON ERROR?
5592  021252  001342                      BNE     10$             ;YES, RECONDITION DRIVE
5593  021254  000137  020704              JMP     2$              ;RETURN TO MAINLINE
5594  021260                      30$:
5595
5596
5597
5598
5599
5600                              ;;*****************************************************************
5601                              ;*TEST 23       BASIC WRITE/READ HEADER & HEAD SWITCHING TEST
5602                              ;*
5603                              ;*      THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
5604                              ;*      ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
5605                              ;*      UNIQUE. 22 SECTOR FORMAT IS USED
5606                              ;*
5607                              ;*      I.E.    TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
5608                              ;*              TRACK 1: 0101 FOR ALL SECTOR HEADERS
5609                              ;*              TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS
5610                              ;*
5611                              ;;*****************************************************************
5612  021260  000004              TST23:  SCOPE
5613  021262  012737  000001  001174      MOV     #1,$TIMES       ;;DO 1 ITERATION
5614  021270  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
5615  021274  005737  001460              TST     LIMERR          ;CHK FOR LIMIT ERROR
5616  021300  001403                      BEQ     5$              ;BR IF NO
5617  021302  104170                      ERROR   170             ;FATAL ERROR
5618  021304  000137  043076              JMP     $EOP            ;ABORT BAL OF TESTS
5619  021310                      5$:
5620
5621  021310  005237  003320              INC     BADHDR          ;USED FOR VALID HALT
5622  021314  005037  001430              CLR     HEAD            ;HEAD CTR
5623
5624  021320  104415                      SCOP1
5625  021322  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
5626
5627  021326  004737  045522              JSR     PC,SUBCLR
5628  021332  104024                      ERROR   24              ;CERR AFTER SCLR
5629
5630  021334  052765  000020  000010  1$: BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
5631  021342  012765  001470  000004      MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
5632  021350  012765  177676  000002      MOV     #-66.,RKWC(R5)  ;WORD COUNT.
5633  021356  000337  001430              SWAB    HEAD
5634  021362  013765  001430  000006      MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
5635  021370  000337  001430              SWAB    HEAD
5636
5637  021374  013700  001430              MOV     HEAD,R0
5638  021400  006300                      ASL     R0              ;DOUBLE R0
5639  021402  016037  001444  001470      MOV     DATA0(R0),HDTAB ;SETUP HEADER WORD FOR RKBA
5640
5641  021410  012765  000027  000000      MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
```

```
5642  021416  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5643  021424  004737  043612                  JSR     PC,FRDY         ;FIND RDY
5644  021430  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5645  021432  004737  045150                  JSR     PC,GSTAT        ;GET FRESH STATUS
5646  021436  032737  100000  003334          BIT     #CERR,HCS1
5647  021444  001405                          BEQ     64$
5648  021446  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
5649  021450  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5650  021454  000137  043076                  JMP     $EOP            ;ABORT DRIVE
5651  021460                          64$:
5652
5653  021460  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5654  021466  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
5655  021472  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5656  021500  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5657  021506  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5658  021512  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5659  021520  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5660
5661  021526  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5662  021532  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5663  021534  104277                          ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
5664  021536  104267                          ERROR   267             ;MSG B0 ERROR
5665  021540  104300                          ERROR   300             ;MSG A1 ERROR
5666  021542  104270                          ERROR   270             ;MSG B1 ERROR
5667
5668
5669  021544  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5670  021550  001401                          BEQ     65$             ;BR IF YES
5671  021552  104303                          ERROR   303             ;MSG A2 NOT CLEARED AFTER WRITE HEADER CMD
5672  021554  005737  001364          65$:    TST     CYLADD          ;SEE IF MSG B2=0
5673  021560  001401                          BEQ     66$             ;BR IF YES
5674  021562  104304                          ERROR   304             ;MSG B2 NOT CLEARED AFTER WRITE HEADER CMD
5675  021564                          66$:
5676
5677  021564  005237  001430                  INC     HEAD
5678  021570  023727  001430  000003          CMP     HEAD,#3
5679  021576  001256                          BNE     1$
5680
5681  021600  005037  001430                  CLR     HEAD            ;HEAD CTR
5682  021604  104415                          SCOP1
5683  021606  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5684
5685  021612  004737  045522                  JSR     PC,SUBCLR
5686  021616  104024                          ERROR   24              ;CERR AFTER SCLR
5687
5688
5689
5690  021620  000337  001430          2$:     SWAB    HEAD
5691  021624  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
5692  021632  000337  001430                  SWAB    HEAD
5693
5694  021636  012700  001674                  MOV     #RHTAB,R0
5695  021642  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
5696  021650  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5697  021656  004737  043612                  JSR     PC,FRDY         ;FIND RDY
```

```
5698  021662  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
5699  021664  032737  100000  003334        BIT     #CERR,HCS1
5700  021672  001405                        BEQ     67$
5701  021674  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
5702  021676  104401  056333                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5703  021702  000137  043076                JMP     $EOP            ;ABORT DRIVE
5704
5705  021706  016520  000024        67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5706  021712  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
5707  021716  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
5708
5709
5710  021722  032765  100000  000010        BIT     #DLT,RKCS2(R5)
5711  021730  001407                        BEQ     68$
5712  021732  004737  045150                JSR     PC,GSTAT
5713  021736  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
5714  021740  104401  056333                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5715  021744  000137  043076                JMP     $EOP            ;ABORT DRIVE
5716  021750                        68$:
5717
5718  021750  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5719  021756  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
5720  021762  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5721  021770  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5722  021776  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
5723  022002  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5724  022010  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5725
5726  022016  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5727  022022  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5728  022024  104301                        ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
5729  022026  104271                        ERROR   271             ;MSG B0 ERROR
5730  022030  104302                        ERROR   302             ;MSG A1 ERROR
5731  022032  104272                        ERROR   272             ;MSG B1 ERROR
5732
5733
5734  022034  005737  001362                TST     CYLDIF          ;SEE IF MSG A2=0
5735  022040  001401                        BEQ     69$             ;BR IF YES
5736  022042  104172                        ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5737  022044  005737  001364        69$:    TST     CYLADD          ;SEE IF MSG B2=0
5738  022050  001401                        BEQ     70$             ;BR IF YES
5739  022052  104264                        ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5740  022054                        70$:
5741  022054  000337  001430                SWAB    HEAD
5742  022060  013765  001430  000006        MOV     HEAD,RKDA(R5)   ;RESTORE RKDA
5743  022066  000337  001430                SWAB    HEAD
5744
5745  022072  012701  001674                MOV     #RHTAB,R1
5746
5747  022076  005037  001442                CLR     WDCNT           ;HEADER WORD COUNT
5748  022102  013700  001430                MOV     HEAD,R0
5749  022106  006300                        ASL     R0              ;DOUBLE R0
5750  022110  016037  001444  003372        MOV     DATA0(R0),TEMP1 ;GET THE 'SHOULD BE' DATA
5751  022116  012137  001454        3$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
5752  022122  023737  001454  003372        CMP     HDWD,TEMP1
5753  022130  001401                        BEQ     4$
```

```
5754  022132  104202                        ERROR   202             ;READ HEADER MISMATCH
5755  022134  005237  001442          4$:   INC     WDCNT
5756  022140  023727  001442  000003        CMP     WDCNT,#3        ;DO ONLY 1 SECTOR
5757  022146  001363                        BNE     3$
5758
5759  022150  005237  001430                INC     HEAD
5760  022154  023727  001430  000003        CMP     HEAD,#3         ;ALL 3 HEADS DONE?
5761  022162  001402                        BEQ     TST24           ;;GO TO NXT TST IF YES
5762  022164  000137  021620                JMP     2$              ;ELSE REPEAT
5763
5764                                ;;*****************************************************************
5765                                ;*TEST 24       BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS
5766                                ;*
5767                                ;*      USING HEAD 0, WRITE & READ 20 SECTOR HEADERS BY WRITING ALL
5768                                ;*      1'S AS HEADERS.  ATTEMPT TO FIND SECTORS 20 & 21.  VERIFY
5769                                ;*      THEY ARE NO LONGER THERE BY READING 22 SECTORS AND NOT
5770                                ;*      FINDING 0'S AS DATA FROM THE PREVIOUS TEST.
5771                                ;*
5772                                ;;*****************************************************************
5773  022170  000004                TST24:  SCOPE
5774  022172  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
5775  022200  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5776
5777  022204  004737  045522                JSR     PC,SUBCLR
5778  022210  104024                        ERROR   24              ;CERR AFTER SCLR
5779  022212  052765  000020  000010        BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
5780  022220  012765  001450  000004        MOV     #DATA1,RKBA(R5) ;XFER 1'S ONLY
5781  022226  012765  177704  000002        MOV     #-60.,RKWC(R5)  ;WORD COUNT
5782
5783
5784  022234  012765  010027  000000        MOV     #<CFMT!WRHEAD>,RKCS1(R5)        ;WRITE HEADER CMD
5785  022242  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP TIMEOUT
5786  022250  004737  043612                JSR     PC,FRDY         ;FIND RDY
5787  022254  104200                        ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5788  022256  004737  045150                JSR     PC,GSTAT        ;GET FRESH STATUS
5789  022262  032737  100000  003334        BIT     #CERR,HCS1
5790  022270  001405                        BEQ     64$
5791  022272  104201                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
5792  022274  104401  056333                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5793  022300  000137  043076                JMP     $EOP            ;ABORT DRIVE
5794  022304                        64$:
5795
5796  022304  012765  010001  000000        MOV     #<CFMT!SELDRV>,RKCS1(R5)        ;GET 20 SECTOR STATUS
5797  022312  013737  001414  003372        MOV     T10,TEMP1
5798  022320  004737  043612                JSR     PC,FRDY         ;FIND RDY
5799  022324  104117                        ERROR   117             ;NO RDY AFTER SELDRV CMD
5800  022326  032737  001000  003362        BIT     #D.FORM,HMR2
5801  022334  001001                        BNE     1$
5802  022336  104312                        ERROR   312             ;FORMAT NOT SET AFTER WRITE HDR CMD
5803
5804  022340                        1$:
5805
5806  022340  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5807  022346  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
5808  022352  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5809  022360  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
```

```
5810  022366  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
5811  022372  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5812  022400  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5813
5814  022406  004737  044334                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5815  022412  000000                            .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5816  022414  104277                            ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
5817  022416  104267                            ERROR   267             ;MSG B0 ERROR
5818  022420  104300                            ERROR   300             ;MSG A1 ERROR
5819  022422  104270                            ERROR   270             ;MSG B1 ERROR
5820
5821  022424  005037  001400                    CLR     SECNT           ;SECTOR COUNT
5822  022430                          2$:
5823  022430  104415                            SCOP1
5824  022432  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
5825
5826  022436  004737  045522                    JSR     PC,SUBCLR
5827  022442  104024                            ERROR   24              ;CERR AFTER SCLR
5828
5829
5830  022444  012700  001674                    MOV     #RHTAB,R0
5831  022450  012765  010025  000000            MOV     #<CFMT!RDHEAD>,RKCS1(R5)        ;READ HEADER CMD
5832  022456  013737  001426  003372            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5833  022464  004737  043612                    JSR     PC,FRDY         ;FIND RDY
5834  022470  104171                            ERROR   171             ;NO RDY AFTER READ HEADER CMD
5835  022472  032737  100000  003334            BIT     #CERR,HCS1
5836  022500  001405                            BEQ     65$
5837  022502  104174                            ERROR   174             ;CERR AFTER READ HEADER CMD
5838  022504  104401  056333                    TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5839  022510  000137  043076                    JMP     $EOP            ;ABORT DRIVE
5840
5841  022514  016520  000024          65$:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5842  022520  016520  000024                    MOV     RKDB(R5),(R0)+  ;2'ND WORD
5843  022524  016520  000024                    MOV     RKDB(R5),(R0)+  ;3'RD WORD
5844
5845
5846  022530  032765  100000  000010            BIT     #DLT,RKCS2(R5)
5847  022536  001407                            BEQ     66$
5848  022540  004737  045150                    JSR     PC,GSTAT
5849  022544  104173                            ERROR   173             ;DLT AFTER READ HEADER CMD
5850  022546  104401  056333                    TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5851  022552  000137  043076                    JMP     $EOP            ;ABORT DRIVE
5852  022556                          66$:
5853
5854  022556  012765  010001  000000            MOV     #<CFMT!SELDRV>,RKCS1(R5)
5855  022564  013737  001414  003372            MOV     T10,TEMP1
5856  022572  004737  043612                    JSR     PC,FRDY         ;FIND RDY
5857  022576  104117                            ERROR   117             ;NO RDY AFTER SELDRV CMD
5858  022600  032737  001000  003362            BIT     #D.FORM,HMR2
5859  022606  001001                            BNE     6$
5860  022610  104313                            ERROR   313             ;FORMAT NOT SET AFTER READ HDR CMD
5861
5862  022612                          6$:
5863
5864  022612  012737  010340  003424            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5865  022620  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
```

```
5866  022624  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5867  022632  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5868  022640  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5869  022644  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5870  022652  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5871
5872  022660  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5873  022664  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5874  022666  104301                          ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
5875  022670  104271                          ERROR   271             ;MSG B0 ERROR
5876  022672  104302                          ERROR   302             ;MSG A1 ERROR
5877  022674  104272                          ERROR   272             ;MSG B1 ERROR
5878
5879  022676  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5880  022702  001401                          BEQ     67$             ;BR IF YES
5881  022704  104172                          ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5882  022706  005737  001364          67$:    TST     CYLADD          ;SEE IF MSG B2=0
5883  022712  001401                          BEQ     68$             ;BR IF YES
5884  022714  104264                          ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5885  022716                          68$:
5886  022716  012701  001674                  MOV     #RHTAB,R1
5887
5888  022722  005037  001442          3$:     CLR     WDCNT           ;HEADER WORD COUNT
5889  022726  013737  001450  003372          MOV     DATA1,TEMP1     ;GET 'SHOULD BE' DATA
5890  022734  012137  001454          4$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
5891  022740  023737  001454  003372          CMP     HDWD,TEMP1      ;MATCH OK?
5892  022746  001401                          BEQ     5$              ;BR IF YES
5893  022750  104202                          ERROR   202             ;READ HEADER MISMATCH
5894  022752  005237  001442          5$:     INC     WDCNT
5895  022756  023727  001442  000003          CMP     WDCNT,#3        ;JUST 1 SECTOR AND 1 HEAD
5896  022764  001363                          BNE     4$
5897
5898
5899
5900
5901                                  ;;*****************************************************************
5902                                  ;*TEST 25       WRITE & READ HEADERS CYL 0, HEAD 0
5903                                  ;;*****************************************************************
5904  022766  000004                  TST25:  SCOPE
5905  022770  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5906  022776  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5907
5908  023002  004737  045522                  JSR     PC,SUBCLR
5909  023006  104024                          ERROR   24              ;CERR AFTER SCLR
5910
5911  023010  005237  001464                  INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
5912
5913  023014  012765  001470  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
5914  023022  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT.
5915  023030  012737  000000  001352          MOV     #0,TOCYL
5916
5917  023036  013737  001352  001366          MOV     TOCYL,CALADD    ;SETUP
5918  023044  012737  000000  001430          MOV     #0,HEAD         ;TO FILL
5919  023052  012737  000000  001436          MOV     #0,FORMAT       ;HEADER
5920  023060  004737  046632                  JSR     PC,FHDTAB       ;TABLE
5921
```

```
 5922  023064  012765  000000  000020          MOV     #0,RKDC(R5)     ;CYL#
 5923
 5924  023072  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
 5925  023100  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
 5926  023106  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 5927  023112  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
 5928  023114  004737  045150                  JSR     PC,GSTAT        ;GET FRESH STATUS
 5929  023120  032737  100000  003334          BIT     #CERR,HCS1
 5930  023126  001405                          BEQ     64$
 5931  023130  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
 5932  023132  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 5933  023136  000137  043076                  JMP     $EOP            ;ABORT DRIVE
 5934  023142                          64$:
 5935
 5936  023142  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
 5937  023150  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
 5938  023154  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 5939  023162  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 5940  023170  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
 5941  023174  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 5942  023202  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5943
 5944  023210  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 5945  023214  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5946  023216  104277                          ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
 5947  023220  104267                          ERROR   267             ;MSG B0 ERROR
 5948  023222  104300                          ERROR   300             ;MSG A1 ERROR
 5949  023224  104270                          ERROR   270             ;MSG B1 ERROR
 5950
 5951  023226  005037  001400                  CLR     SECNT           ;SECTOR COUNT
 5952  023232  104415                          SCOP1
 5953  023234  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 5954
 5955  023240  004737  045522                  JSR     PC,SUBCLR
 5956  023244  104024                          ERROR   24              ;CERR AFTER SCLR
 5957
 5958  023246  012765  000000  000020          MOV     #0,RKDC(R5)     ;CYL #
 5959
 5960  023254  012700  001674                  MOV     #RHTAB,R0
 5961
 5962  023260  012765  000025  000000  65$:     MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
 5963  023266  013737  001420  003372          MOV     T500,TEMP1      ;SETUP  TIMEOUT
 5964  023274  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 5965  023300  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
 5966  023302  032737  100000  003334          BIT     #CERR,HCS1
 5967  023310  001405                          BEQ     66$
 5968  023312  104174                          ERROR   174             ;CERR AFTER READ HEADER  CMD
 5969  023314  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 5970  023320  000137  043076                  JMP     $EOP            ;ABORT DRIVE
 5971
 5972  023324  016520  000024          66$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 5973  023330  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
 5974  023334  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
 5975
 5976  023340  032765  100000  000010          BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
 5977  023346  001407                          BEQ     67$
```

```
5978  023350  004737  045150            JSR     PC,GSTAT
5979  023354  104173                    ERROR   173             ;DATA LATE ON READ HEADER
5980  023356  104401  056333            TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5981  023362  000137  043076            JMP     $EOP            ;ABORT DRIVE
5982
5983  023366  020027  002100    67$:    CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
5984  023372  001332                    BNE     65$             ;BR IF NO
5985
5986  023374  004737  047154            JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
5987                                                            ;BEGINS WITH SECTOR 0
5988  023400  005037  001442            CLR     WDCNT           ;WORD COUNT
5989  023404  012700  002100            MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
5990  023410  012701  001470            MOV     #HDTAB,R1       ;CALC HEADER TABLE
5991
5992  023414  012037  001454    68$:    MOV     (R0)+,HDWD
5993  023420  012137  003372            MOV     (R1)+,TEMP1
5994  023424  023737  001454  003372    CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
5995  023432  001401                    BEQ     69$             ;BR IF COMPARE
5996  023434  104202                    ERROR   202             ;READ HEADER MISMATCH
5997
5998  023436  005237  001442    69$:    INC     WDCNT
5999  023442  023727  001442  000102    CMP     WDCNT,#66.      ;ALL WORDS DONE?
6000  023450  001361                    BNE     68$             ;BR IF NO
6001
6002
6003  023452  005037  001464            CLR BYPFMT              ;ALLOW CORRECT FORMATTING
6004
6005
6006
6007
6008                                    ;;*****************************************************************
6009                                    ;*TEST 26       SEEK FROM CYL 0 TO 1 & READ HEADERS
6010                                    ;*
6011                                    ;*      THIS TEST CHECKS MSG A & B WORDS 0,1,2 FOR CORRECT STATUS AFTER RDY
6012                                    ;*      IS RECEIVED FROM A SEEK CMD TO DETERMINE
6013                                    ;*      THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1.
6014                                    ;*      AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS.
6015                                    ;*      CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR
6016                                    ;*      IN MSG B2 IS VERIFIED TO BE 1.
6017                                    ;*
6018                                    ;*      HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE
6019                                    ;*      DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.
6020                                    ;*
6021                                    ;*****************************************************************
6022  023456  000004            TST26:  SCOPE
6023  023460  012737  000001  001174    MOV     #1,$TIMES       ;;DO 1 ITERATION
6024  023466  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
6025  023472  004737  045522            JSR     PC,SUBCLR
6026  023476  104024                    ERROR   24              ;CERR AFTER SCLR
6027  023500  005037  001350            CLR     FRCYL
6028  023504  012737  000001  001352    MOV     #1,TOCYL
6029  023512  012737  000001  001360    MOV     #1,CALDIF
6030  023520  012765  000001  000020    MOV     #1,RKDC(R5)     ;SET FOR CYL 1
6031  023526  012737  024542  001176    MOV     #10$,$ESCAPE
6032
6033  023534  012765  000017  000000    MOV     #SEEK,RKCS1(R5) ;SEEK CMD
```

```
6034  023542  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
6035  023550  004737  043612                  JSR     PC,FRDY         ;FIND RDY
6036  023554  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
6037  023556  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
6038  023564  005037  003426                  CLR     E.B0
6039  023570  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6040  023576  012737  000001  003432          MOV     #1,E.B1
6041
6042  023604  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6043  023610  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6044  023612  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
6045  023614  104204                          ERROR   204             ;MSG B0 ERROR
6046  023616  104205                          ERROR   205             ;MSG A1 ERROR
6047  023620  104206                          ERROR   206             ;MSG B1 ERROR
6048
6049  023622  023727  001362  000001          CMP     CYLDIF,#1
6050  023630  001401                          BEQ     1$
6051  023632  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK CMD.
6052
6053
6054  023634  012737  024562  001176  1$:     MOV     #12$,$ESCAPE
6055  023642  013737  001422  003372          MOV     T2500,TEMP1     ;SETUP TIMEOUT
6056
6057
6058  023650  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
6059  023654  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6060  023656  032737  100000  003334          BIT     #CERR,HCS1
6061  023664  001401                          BEQ     64$
6062  023666  104210                          ERROR   210             ;CERR AFTER SEEK CMD
6063  023670                          64$:
6064
6065  023670  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
6066  023676  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
6067  023702  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6068  023710  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6069  023716  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
6070  023722  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6071  023730  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6072
6073  023736  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6074  023742  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6075  023744  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6076  023746  104134                          ERROR   134             ;MSG B0 ERROR
6077  023750  104135                          ERROR   135             ;MSG A1 ERROR
6078  023752  104136                          ERROR   136             ;MSG B1 ERROR
6079  023754  005737  001362                  TST     CYLDIF
6080  023760  001401                          BEQ     65$
6081  023762  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6082
6083  023764                          65$:
6084
6085  023764  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6086  023772  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6087  024000  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6088  024006  013737  001414  003372          MOV     T10,TEMP1
6089  024014  004737  043612                  JSR     PC,FRDY         ;FIND RDY
```

```
6090  024020  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6091  024022  004737  044074                 JSR     PC,TSTATN       ;TEST FOR ATTN
6092  024026  000401                         BR      66$
6093  024030  104154                         ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6094  024032                          66$:
6095
6096  024032  012737  010340  003424         MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
6097  024040  005037  003426                 CLR     E.B0            ;EXPECTED MSG B0
6098  024044  012737  001720  003430         MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6099  024052  012737  000001  003432         MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6100  024060  005037  003434                 CLR     E.A2            ;EXPECTED MSG A2
6101  024064  012737  000002  003436         MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6102  024072  012737  000003  003442         MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6103
6104  024100  004737  044334                 JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6105  024104  000003                         .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6106  024106  104273                         ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6107  024110  104265                         ERROR   265             ;MSG B0 ERROR
6108  024112  104274                         ERROR   274             ;MSG A1 ERROR
6109  024114  104266                         ERROR   266             ;MSG B1 ERROR
6110
6111  024116  005737  001364                 TST     CYLADD
6112  024122  023727  001364  000001         CMP     CYLADD,#1
6113  024130  001401                         BEQ     2$
6114  024132  104207                         ERROR   207             ;CYL ADDR INCORRECT AFTER SEEK CMD
6115
6116
6117  024134                          2$:
6118  024134  104415                         SCOP1
6119  024136  012706  001100                 MOV     #STACK,SP       ;RESTORE STK PTR
6120
6121  024142  004737  045522                 JSR     PC,SUBCLR
6122  024146  104024                         ERROR   24              ;CERR AFTER SCLR
6123
6124  024150  005037  001176                 CLR     $ESCAPE
6125  024154  012765  000001  000020         MOV     #1,RKDC(R5)     ;CYL #
6126
6127  024162  012700  001674                 MOV     #RHTAB,R0
6128  024166  012765  000025  000000         MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
6129  024174  013737  001426  003372         MOV     T50000,TEMP1    ;SETUP TIMEOUT
6130  024202  004737  043612                 JSR     PC,FRDY         ;FIND RDY
6131  024206  104171                         ERROR   171             ;NO RDY AFTER READ HEADER CMD
6132  024210  032737  100000  003334         BIT     #CERR,HCS1
6133  024216  001405                         BEQ     67$
6134  024220  104174                         ERROR   174             ;CERR AFTER READ HEADER CMD
6135  024222  104401  056333                 TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
6136  024226  000137  043076                 JMP     $EOP            ;ABORT DRIVE
6137
6138  024232  016520  000024          67$:   MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6139  024236  016520  000024                 MOV     RKDB(R5),(R0)+  ;2'ND WORD
6140  024242  016520  000024                 MOV     RKDB(R5),(R0)+  ;3'RD WORD
6141
6142
6143  024246  032765  100000  000010         BIT     #DLT,RKCS2(R5)
6144  024254  001407                         BEQ     68$
6145  024256  004737  045150                 JSR     PC,GSTAT
```

```
6146  024262  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
6147  024264  104401  056333                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6148  024270  000137  043076                JMP     $EOP            ;ABORT DRIVE
6149  024274                        68$:
6150
6151
6152  024274  005737  001674                TST     RHTAB           ;CHECK 1'ST WORD ONLY:CYL#
6153  024300  001001                        BNE     3$
6154  024302  104211                        ERROR   211             ;CYL 0 HEADER ON CYL 1
6155
6156  024304  013737  001674  001454  3$:   MOV     RHTAB,HDWD
6157  024312  012737  000001  003372        MOV     #1,TEMP1
6158  024320  023737  001454  003372        CMP     HDWD,TEMP1
6159  024326  001401                        BEQ     4$
6160  024330  104202                        ERROR   202             ;READ CYL WORD HEADER ERROR
6161  024332                        4$:
6162  024332  004737  047526                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6163  024336  000521                        BR      TST27           ;;GO TO NEXT TEST
6164                                                                ;RETURN HERE IF SW 14 IS SET OR
6165                                                                ;SW 8 WITH SWR <7:0> APPLY
6166  024340  004737  045522        6$:     JSR     PC,SUBCLR
6167  024344  104024                        ERROR   24              ;CERR AFTER SCLR
6168
6169  024346  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6170  024354  013737  001414  003372        MOV     T10,TEMP1       ;SETUP TIMEOUT
6171  024362  004737  043612                JSR     PC,FRDY         ;FIND RDY
6172  024366  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
6173
6174  024370  013737  001426  003372        MOV     T50000,TEMP1
6175  024376  004737  044222                JSR     PC,FATT2        ;FIND ATTN
6176  024402  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
6177  024404  032737  100000  003334        BIT     #CERR,HCS1
6178  024412  001401                        BEQ     69$
6179  024414  104210                        ERROR   210             ;CERR AFTER SEEK CMD.
6180
6181  024416  004737  045522        69$:    JSR     PC,SUBCLR
6182  024422  104024                        ERROR   24              ;CERR AFTER SCLR
6183
6184
6185  024424  012737  050340  003424        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
6186  024432  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
6187  024436  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6188  024444  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6189  024452  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
6190  024456  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6191  024464  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6192
6193  024472  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6194  024476  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6195  024500  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6196  024502  104134                        ERROR   134             ;MSG B0 ERROR
6197  024504  104135                        ERROR   135             ;MSG A1 ERROR
6198  024506  104136                        ERROR   136             ;MSG B1 ERROR
6199  024510  005737  001364                TST     CYLADD
6200  024514  001401                        BEQ     7$
6201  024516  104043                        ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
```

```
6202
6203  024520                          7$:
6204  024520  005037  001176                  CLR     $ESCAPE
6205  024524  005737  001410                  TST     LPFLG
6206  024530  001402                          BEQ     70$
6207  024532  000177  154352                  JMP     @$LPERR         ;SW 9 WAS SET.
6208  024536  000177  154344          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6209
6210  024542                          10$:
6211  024542  005237  001410                  INC     LPFLG
6212  024546  032777  001000  154364        BIT     #SW9,@SWR       ;LOOP ON ERROR?
6213  024554  001271                        BNE     6$              ;YES, RECONDITION DRIVE
6214  024556  000137  023634                JMP     1$              ;RETURN TO MAINLINE
6215  024562                          12$:
6216  024562  005237  001410                  INC     LPFLG
6217  024566  032777  001000  154344        BIT     #SW9,@SWR       ;LOOP ON ERROR?
6218  024574  001261                        BNE     6$              ;YES, RECONDITION DRIVE
6219  024576  000137  024134                JMP     2$              ;RETURN TO MAINLINE
6220
6221
6222
6223
6224                                  ;;*****************************************************************
6225                                  ;*TEST 27       WRITE & READ HEADERS CYL 1, HEAD 0
6226                                  ;;*****************************************************************
6227  024602  000004                  TST27:  SCOPE
6228  024604  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
6229  024612  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
6230
6231  024616  004737  045522                JSR     PC,SUBCLR
6232  024622  104024                        ERROR   24              ;CERR AFTER SCLR
6233
6234  024624  005237  001464                INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
6235
6236  024630  012765  001470  000004        MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
6237  024636  012765  177676  000002        MOV     #-66.,RKWC(R5)  ;WORD COUNT.
6238  024644  012737  000001  001352        MOV     #1,TOCYL
6239
6240  024652  013737  001352  001366        MOV     TOCYL,CALADD    ;SETUP
6241  024660  012737  000000  001430        MOV     #0,HEAD         ;TO FILL
6242  024666  012737  000000  001436        MOV     #0,FORMAT       ;HEADER
6243  024674  004737  046632                JSR     PC,FHDTAB       ;TABLE
6244
6245  024700  012765  000001  000020        MOV     #1,RKDC(R5)     ;CYL#
6246
6247  024706  012765  000027  000000        MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
6248  024714  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP TIMEOUT
6249  024722  004737  043612                JSR     PC,FRDY         ;FIND RDY
6250  024726  104200                        ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
6251  024730  004737  045150                JSR     PC,GSTAT        ;GET FRESH STATUS
6252  024734  032737  100000  003334        BIT     #CERR,HCS1
6253  024742  001405                        BEQ     64$
6254  024744  104201                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
6255  024746  104401  056333                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6256  024752  000137  043076                JMP     $EOP            ;ABORT DRIVE
6257  024756                          64$:
```

```
6258
6259  024756  012737  010340  003424          MOV   #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0    ;EXPECTED MSG A0
6260  024764  005037  003426                  CLR   E.B0                ;EXPECTED MSG B0
6261  024770  012737  001720  003430          MOV   #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
6262  024776  012737  000001  003432          MOV   #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
6263  025004  005037  003434                  CLR   E.A2                ;EXPECTED MSG A2
6264  025010  012737  000002  003436          MOV   #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
6265  025016  012737  000003  003442          MOV   #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
6266
6267  025024  004737  044334                  JSR   PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
6268  025030  000003                          .WORD T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
6269  025032  104277                          ERROR 277                 ;MSG A0 ERROR AFTER WRITE HEADER CMD
6270  025034  104267                          ERROR 267                 ;MSG B0 ERROR
6271  025036  104300                          ERROR 300                 ;MSG A1 ERROR
6272  025040  104270                          ERROR 270                 ;MSG B1 ERROR
6273
6274  025042  005037  001400                  CLR   SECNT               ;SECTOR COUNT
6275  025046  104415                          SCOP1
6276  025050  012706  001100                  MOV   #STACK,SP           ;RESTORE STK PTR
6277
6278  025054  004737  045522                  JSR   PC,SUBCLR
6279  025060  104024                          ERROR 24                  ;CERR AFTER SCLR
6280
6281  025062  012765  000001  000020          MOV   #1,RKDC(R5)         ;CYL #
6282
6283  025070  012700  001674                  MOV   #RHTAB,R0
6284
6285  025074  012765  000025  000000  65$:    MOV   #RDHEAD,RKCS1(R5)           ;READ HEADER CMD
6286  025102  013737  001420  003372          MOV   T500,TEMP1          ;SETUP  TIMEOUT
6287  025110  004737  043612                  JSR   PC,FRDY             ;FIND RDY
6288  025114  104171                          ERROR 171                 ;NO RDY AFTER READ HEADER CMD
6289  025116  032737  100000  003334          BIT   #CERR,HCS1
6290  025124  001405                          BEQ   66$
6291  025126  104174                          ERROR 174                 ;CERR AFTER READ HEADER  CMD
6292  025130  104401  056333                  TYPE  ,MSG18              ;ABORTING BALANCE OF TESTS
6293  025134  000137  043076                  JMP   $EOP                ;ABORT DRIVE
6294
6295  025140  016520  000024          66$:    MOV   RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
6296  025144  016520  000024                  MOV   RKDB(R5),(R0)+      ;2'ND WORD
6297  025150  016520  000024                  MOV   RKDB(R5),(R0)+      ;3'RD WORD
6298
6299  025154  032765  100000  000010          BIT   #DLT,RKCS2(R5)      ;SEE IF DATA LATE
6300  025162  001407                          BEQ   67$
6301  025164  004737  045150                  JSR   PC,GSTAT
6302  025170  104173                          ERROR 173                 ;DATA LATE ON READ HEADER
6303  025172  104401  056333                  TYPE  ,MSG18              ;ABORT BALANCE OF TESTS
6304  025176  000137  043076                  JMP   $EOP                ;ABORT DRIVE
6305
6306  025202  020027  002100          67$:    CMP   R0,#RHTAB+132.      ;ALL 66 WORDS DONE?
6307  025206  001332                          BNE   65$                 ;BR IF NO
6308
6309  025210  004737  047154                  JSR   PC,SORT             ;SORT RHTAB INTO SRTTAB SO THAT IT
6310                                                                    ;BEGINS WITH SECTOR 0
6311  025214  005037  001442                  CLR   WDCNT               ;WORD COUNT
6312  025220  012700  002100                  MOV   #SRTTAB,R0          ;ACTUAL HEADER TABLE
6313  025224  012701  001470                  MOV   #HDTAB,R1           ;CALC HEADER TABLE
```

```
6314
6315  025230  012037  001454          68$:    MOV     (R0)+,HDWD
6316  025234  012137  003372                  MOV     (R1)+,TEMP1
6317  025240  023737  001454  003372          CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
6318  025246  001401                          BEQ     69$             ;BR IF COMPARE
6319  025250  104202                          ERROR   202             ;READ HEADER MISMATCH
6320
6321  025252  005237  001442          69$:    INC     WDCNT
6322  025256  023727  001442  000102          CMP     WDCNT,#66.      ;ALL WORDS DONE?
6323  025264  001361                          BNE     68$             ;BR IF NO
6324
6325
6326  025266  005037  001464                  CLR BYPFMT              ;ALLOW CORRECT FORMATTING
6327
6328
6329
6330                                  ;;*****************************************************************
6331                                  ;*TEST 30       TEST RECALIBRATE CMD & READ HEADERS
6332                                  ;*
6333                                  ;*      THIS TEST DOES A RECALIBRATE & READS HEADERS.
6334                                  ;*      IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS
6335                                  ;*      TEST DID NOT OVERWRITE CYL 0 HEADERS.
6336                                  ;*      AN ERROR IN THIS TEST INDICATES THAT HEADS:
6337                                  ;*
6338                                  ;*
6339                                  ;*              A. MOVED TO A CYL OTHER THAN 1
6340                                  ;*      OR      B. DID NOT GET BACK TO CYL 0
6341                                  ;*
6342                                  ;*****************************************************************
6343  025272  000004          TST30:  SCOPE
6344  025274  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6345  025302  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6346
6347  025306  004737  045522                  JSR     PC,SUBCLR
6348  025312  104024                          ERROR   24              ;CERR AFTER SCLR
6349  025314  012737  000001  001350          MOV     #1,FRCYL        ;PARAMETERS
6350  025322  005037  001352                  CLR     TOCYL           ;FOR
6351  025326  012737  000001  001360          MOV     #1,CALDIF       ;ERROR TYPEOUTS
6352  025334  012737  026360  001176          MOV     #10$,$ESCAPE
6353  025342  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
6354  025350  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
6355  025356  004737  043612                  JSR     PC,FRDY         ;FIND RDY
6356  025362  104124                          ERROR   124             ;NO RDY AFTER RECAL CMD
6357  025364  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6358  025372  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6359  025400  004737  045150                  JSR     PC,GSTAT
6360  025404  032737  020000  003362          BIT     #D.RTZ,HMR2
6361  025412  001001                          BNE     1$
6362  025414  104307                          ERROR   307             ;RTZ NOT SET DURING RECAL CMD
6363
6364  025416  012737  030140  003424  1$:     MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
6365  025424  005037  003426                  CLR     E.B0
6366  025430  012737  025720  003430          MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6367  025436  012737  000001  003432          MOV     #1,E.B1
6368
6369  025444  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
```

```
                                        .WORD   T.A2!0!0                ;& MSGS SPECIFIED HERE
                                        ERROR   213             ;MSG A0 ERROR DURING RECAL CMD
                                        ERROR   214             ;MSG B0 ERROR
                                        ERROR   215             ;MSG A1 ERROR
                                        ERROR   216             ;MSG B1 ERROR
                                        TST     CYLDIF
                                        BEQ     2$
                                        ERROR   217             ;CYL DIFF INCORRECT DURING RECAL CMD.

                        2$:             MOV     #12$,$ESCAPE
                                        MOV     #-1,TEMP1       ;SETUP TIMEOUT
                                        JSR     PC,FATT2        ;FIND ATTN
                                        ERROR   55              ;NO ATTN AFTER RECAL CMD
                                        BIT     #CERR,HCS1
                                        BEQ     3$
                                        ERROR   220             ;CERR AFTER RECAL CMD
                        3$:

                                        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
                                        CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
                                        CLR     E.A2            ;EXPECTED MSG A2
                                        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
                                        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
                                        ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
                                        ERROR   275             ;MSG B0 ERROR
                                        ERROR   222             ;MSG A1 ERROR
                                        ERROR   276             ;MSG B1 ERROR

                                        TST     CYLDIF          ;SEE IF MSG A2=0
                                        BEQ     64$             ;BR IF YES
                                        ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
                        64$:            TST     CYLADD          ;SEE IF MSG B2=0
                                        BEQ     65$             ;BR IF YES
                                        ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
                        65$:

                                        MOV     #CCLR,RKCS1(R5)
                                        MOV     $UNIT,RKCS2(R5) ;DRIVE#
                                        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
                                        MOV     T10,TEMP1
                                        JSR     PC,FRDY         ;FIND RDY
                                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                                        JSR     PC,TSTATN       ;TEST FOR ATTN
                                        BR      66$
                                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                        66$:

                                        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
                                        CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
```

```
6426  025726  005037  003434                CLR     E.A2                    ;EXPECTED MSG A2
6427  025732  012737  000002  003436        MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
6428  025740  012737  000003  003442        MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
6429
6430  025746  004737  044334                JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
6431  025752  000003                        .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
6432  025754  104273                        ERROR   273                     ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6433  025756  104265                        ERROR   265                     ;MSG B0 ERROR
6434  025760  104274                        ERROR   274                     ;MSG A1 ERROR
6435  025762  104266                        ERROR   266                     ;MSG B1 ERROR
6436
6437  025764  004737  045522        4$:     JSR     PC,SUBCLR
6438  025770  104024                        ERROR   24                      ;CERR AFTER SCLR
6439
6440  025772  005037  001176                CLR     $ESCAPE
6441
6442  025776  012700  001674                MOV     #RHTAB,R0
6443  026002  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
6444  026010  013737  001426  003372        MOV     T50000,TEMP1            ;SETUP  TIMEOUT
6445  026016  004737  043612                JSR     PC,FRDY                 ;FIND RDY
6446  026022  104171                        ERROR   171                     ;NO RDY AFTER READ HEADER CMD
6447  026024  032737  100000  003334        BIT     #CERR,HCS1
6448  026032  001405                        BEQ     67$
6449  026034  104174                        ERROR   174                     ;CERR AFTER READ HEADER CMD
6450  026036  104401  056333                TYPE    ,MSG18                  ;ABORT BALANCE OF TESTS
6451  026042  000137  043076                JMP     $EOP                    ;ABORT DRIVE
6452
6453  026046  016520  000024        67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6454  026052  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
6455  026056  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
6456
6457
6458  026062  032765  100000  000010        BIT     #DLT,RKCS2(R5)
6459  026070  001407                        BEQ     68$
6460  026072  004737  045150                JSR     PC,GSTAT
6461  026076  104173                        ERROR   173                     ;DLT AFTER READ HEADER CMD
6462  026100  104401  056333                TYPE    ,MSG18                  ;ABORTING BALANCE OF TESTS
6463  026104  000137  043076                JMP     $EOP                    ;ABORT DRIVE
6464  026110                        68$:
6465
6466  026110  023727  001674  000001        CMP     RHTAB,#1                ;CHECK WORD 0, CYL # ONLY
6467  026116  001001                        BNE     5$
6468  026120  104240                        ERROR   240                     ;CYL 1 HEADERS ON CYL 0
6469
6470  026122  005737  001674        5$:     TST     RHTAB
6471  026126  001401                        BEQ     6$
6472  026130  104202                        ERROR   202                     ;READ CYL WORD HEADER ERROR
6473  026132                        6$:
6474  026132  004737  047526                JSR     PC,SWTST                ;SEE IF SW 14 OR 8 IS SET
6475  026136  000530                        BR      TST31                   ;;GO TO NEXT TEST
6476                                                                        ;RETURN HERE IF SW 14 IS SET OR
6477                                                                        ;SW 8 WITH SWR <7:0> APPLY
6478  026140  004737  045522        8$:     JSR     PC,SUBCLR
6479  026144  104024                        ERROR   24                      ;CERR AFTER SCLR
6480  026146  012765  000001  000020        MOV     #1,RKDC(R5)             ;RECONDITION BACK TO CYL 1
6481
```

```
6482  026154  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6483  026162  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
6484  026170  004737  043612                  JSR     PC,FRDY         ;FIND RDY
6485  026174  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
6486
6487  026176  013737  001426  003372          MOV     T50000,TEMP1
6488  026204  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
6489  026210  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6490  026212  032737  100000  003334          BIT     #CERR,HCS1
6491  026220  001401                          BEQ     69$
6492  026222  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
6493
6494  026224  004737  045522          69$:    JSR     PC,SUBCLR
6495  026230  104024                          ERROR   24              ;CERR AFTER SCLR
6496
6497
6498
6499  026232  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
6500  026240  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
6501  026244  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
6502  026252  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6503  026260  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
6504  026264  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6505  026272  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6506  026300  012737  000022  003436          MOV     #<BIT4!2>,E.B2  ;EXPECTED MSG B2 & ID FOR CYL 1
6507
6508  026306  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6509  026312  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6510  026314  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6511  026316  104134                          ERROR   134             ;MSG B0 ERROR
6512  026320  104135                          ERROR   135             ;MSG A1 ERROR
6513  026322  104136                          ERROR   136             ;MSG B1 ERROR
6514  026324  023727  001364  000001          CMP     CYLADD,#1
6515  026332  001401                          BEQ     9$
6516  026334  104043                          ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
6517  026336                          9$:
6518  026336  005037  001176                  CLR     $ESCAPE
6519  026342  005737  001410                  TST     LPFLG
6520  026346  001402                          BEQ     70$
6521  026350  000177  152534                  JMP     @$LPERR         ;SW 9 WAS SET.
6522  026354  000177  152526          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6523  026360                          10$:
6524  026360  005237  001410                  INC     LPFLG
6525  026364  032777  001000  152546          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6526  026372  001262                          BNE     8$              ;YES, RECONDITION DRIVE
6527  026374  000137  025472                  JMP     2$              ;RETURN TO MAINLINE
6528  026400                          12$:
6529  026400  005237  001410                  INC     LPFLG
6530  026404  032777  001000  152526          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6531  026412  001252                          BNE     8$              ;YES, RECONDITION DRIVE
6532  026414  000137  025764                  JMP     4$              ;RETURN TO MAINLINE
6533                          ;;*****************************************************************
6534                          ;*TEST 31        SINGLE INCREMENT SEEKS TO CYL 410
6535                          ;*
6536                          ;*      THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO CYL 410
6537                          ;*      WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY
```

```
                                        ;*      DESTROY DATA.
                                        ;*
                                        ;:*****************************************************************
                                        ;TST31: SCOPE
026420  000004
026422  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
026430  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
026434  004737  045522                  JSR     PC,SUBCLR
026440  104024                          ERROR   24              ;CERR AFTER SCLR
026442  005037  001350                  CLR     FRCYL           ;FROM CYL
026446  012737  000001  001352          MOV     #1,TOCYL        ;TO CYL
026454  012737  000001  001360          MOV     #1,CALDIF       ;CALCULATED DIFF.
026462                                  1$:
026462  104415                          SCOP1
026464  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
026470  004737  045522                  JSR     PC,SUBCLR
026474  104024                          ERROR   24              ;CERR AFTER SCLR
026476  012737  027302  001176          MOV     #10$,$ESCAPE
026504  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;CYL TO SEEK TO
026512  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
026520  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
026526  004737  043612                  JSR     PC,FRDY         ;FIND RDY
026532  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
026534  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
026542  005037  003426                  CLR     E.B0
026546  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
026554  012737  000001  003432          MOV     #1,E.B1
026562  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
026566  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
026570  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
026572  104204                          ERROR   204             ;MSG B0 ERROR
026574  104205                          ERROR   205             ;MSG A1 ERROR
026576  104206                          ERROR   206             ;MSG B1 ERROR
026600  023727  001362  000001          CMP     CYLDIF,#1
026606  001401                          BEQ     2$
026610  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
026612  012737  027322  001176  2$:     MOV     #12$,$ESCAPE
026620  013737  001422  003372          MOV     T2500,TEMP1     ;SETUP TIMEOUT
026626  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
026632  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
026634  032737  100000  003334          BIT     #CERR,HCS1
026642  001401                          BEQ     64$
026644  104210                          ERROR   210             ;CERR AFTER SEEK CMD
026646                                  64$:
026646  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
026654  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
026660  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
```

```
6594  026666  012737  000001  003432          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
6595  026674  005037  003434                  CLR     E.A2             ;EXPECTED MSG A2
6596  026700  012737  000002  003436          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
6597  026706  012737  000003  003442          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
6598
6599  026714  004737  044334                  JSR     PC,CHKMSG        ;CHECK MSGS A0,B0,A1,B1
6600  026720  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
6601  026722  104133                          ERROR   133              ;MSG A0 ERROR AFTER SEEK CMD
6602  026724  104134                          ERROR   134              ;MSG B0 ERROR
6603  026726  104135                          ERROR   135              ;MSG A1 ERROR
6604  026730  104136                          ERROR   136              ;MSG B1 ERROR
6605  026732  005737  001362                  TST     CYLDIF
6606  026736  001401                          BEQ     65$
6607  026740  104137                          ERROR   137              ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6608
6609  026742                          65$:
6610
6611  026742  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6612  026750  013765  001222  000010          MOV     $UNIT,RKCS2(R5)  ;DRIVE#
6613  026756  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)         ;DRIVE CLEAR CMD
6614  026764  013737  001414  003372          MOV     T10,TEMP1
6615  026772  004737  043612                  JSR     PC,FRDY          ;FIND RDY
6616  026776  104151                          ERROR   151              ;NO RDY AFTER DRIVE CLEAR CMD
6617  027000  004737  044074                  JSR     PC,TSTATN        ;TEST FOR ATTN
6618  027004  000401                          BR      66$
6619  027006  104154                          ERROR   154              ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6620  027010                          66$:
6621
6622  027010  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
6623  027016  005037  003426                  CLR     E.B0             ;EXPECTED MSG B0
6624  027022  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1         ;EXPECTED A1
6625  027030  012737  000001  003432          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
6626  027036  005037  003434                  CLR     E.A2             ;EXPECTED MSG A2
6627  027042  012737  000002  003436          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
6628  027050  012737  000003  003442          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
6629
6630  027056  004737  044334                  JSR     PC,CHKMSG        ;CHECK MSGS A0,B0,A1,B1
6631  027062  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
6632  027064  104273                          ERROR   273              ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6633  027066  104265                          ERROR   265              ;MSG B0 ERROR
6634  027070  104274                          ERROR   274              ;MSG A1 ERROR
6635  027072  104266                          ERROR   266              ;MSG B1 ERROR
6636
6637  027074  023737  001364  001352          CMP     CYLADD,TOCYL
6638  027102  001401                          BEQ     3$
6639  027104  104207                          ERROR   207              ;CYL ADDR IN RKMR2 NOT=RKDC
6640
6641  027106  023727  001352  000632  3$:     CMP     TOCYL,#410.      ;ALL CYL DONE?
6642  027114  001407                          BEQ     4$               ;BR IF YES
6643  027116  005237  001350                  INC     FRCYL            ;ELSE DO ANOTHER
6644  027122  005237  001352                  INC     TOCYL
6645  027126  001402                          BEQ     4$               ;BR IF YES
6646  027130  000137  026462                  JMP     1$
6647
6648  027134                          4$:
6649  027134  004737  047526                  JSR     PC,SWTST         ;SEE IF SW 14 OR 8 IS SET
```

```
6650  027140  000500                        BR      TST32           ;;GO TO NEXT TEST
6651                                                                  ;RETURN HERE IF SW 14 IS SET OR
6652                                                                  ;SW 8 WITH SWR <7:0> APPLY
6653
6654
6655
6656  027142                        6$:
6657
6658  027142  004737  045522              JSR     PC,SUBCLR
6659  027146  104024                      ERROR   24              ;CERR AFTER SCRL
6660
6661  027150  013765  001352  000020 67$:  MOV     TOCYL,RKDC(R5)  ;CYL#
6662
6663  027156  012765  000017  000000      MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6664  027164  013737  001414  003372      MOV     T10,TEMP1       ;SETUP TIMEOUT
6665  027172  004737  043612              JSR     PC,FRDY         ;FIND RDY
6666  027176  104131                      ERROR   131             ;NO RDY AFTER SEEK CMD.
6667
6668  027200  013737  001426  003372      MOV     T50000,TEMP1
6669  027206  004737  044222              JSR     PC,FATT2        ;FIND ATTN
6670  027212  104132                      ERROR   132             ;NO ATTN AFTER SEEK CMD
6671  027214  032737  100000  003334      BIT     #CERR,HCS1
6672  027222  001401                      BEQ     69$
6673  027224  104210                      ERROR   210             ;CERR AFTER SEEK CMD.
6674
6675  027226  004737  045522        69$:  JSR     PC,SUBCLR
6676  027232  104024                      ERROR   24              ;CERR AFTER SCLR
6677
6678  027234  023727  001352  000000      CMP     TOCYL,#0        ;ALL CYL DONE?
6679  027242  001403                      BEQ     68$             ;BR IF YES
6680  027244  005337  001352              DEC     TOCYL           ;ELSE DO ANOTHER
6681  027250  000737                      BR      67$
6682
6683  027252  004737  045522        68$:  JSR     PC,SUBCLR
6684  027256  104024                      ERROR   24              ;CERR AFTER SCLR
6685
6686  027260  005037  001176              CLR     $ESCAPE
6687  027264  005737  001410              TST     LPFLG
6688  027270  001402                      BEQ     70$
6689  027272  000177  151612              JMP     @$LPERR         ;SW 9 WAS SET.
6690  027276  000177  151604        70$:  JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6691
6692
6693
6694  027302                        10$:
6695  027302  005237  001410              INC     LPFLG
6696  027306  032777  001000  151624      BIT     #SW9,@SWR       ;LOOP ON ERROR?
6697  027314  001312                      BNE     6$              ;YES, RECONDITION DRIVE
6698  027316  000137  026612              JMP     2$              ;RETURN TO MAINLINE
6699
6700  027322                        12$:
6701  027322  005237  001410              INC     LPFLG
6702  027326  032777  001000  151604      BIT     #SW9,@SWR       ;LOOP ON ERROR?
6703  027334  001302                      BNE     6$              ;YES, RECONDITION DRIVE
6704  027336  000137  027134              JMP     4$              ;RETURN TO MAINLINE
6705
```

```
                                ;;*****************************************************************
                                ;*TEST 32       READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
                                ;*
                                ;*      THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
                                ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
                                ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
                                ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
                                ;*
                                ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED FOR BAD
                                ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
                                ;*
                                ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
                                ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
                                ;*      A MSG WILL BE TYPED INDICATING THAT ALL
                                ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
                                ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
                                ;*
                                ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
                                ;*
                                ;*       THIS IS THE FIRST TEST WHERE THE READ DATA CMD IS PERFORMED
                                ;*
                                ;;*****************************************************************
027342  000004                  TST32:  SCOPE
027344  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
027352  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
027356  004737  045522                  JSR     PC,SUBCLR
027362  104024                  1$:     ERROR   24              ;CERR AFTER SCLR
027364  005037  003374                  CLR     TEMP2           ;SECTOR CTR
027370  005037  003376                  CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
                                                                ;1=22 SECTOR SOFTWARE DETECTED TABLE
                                                                ;2=DONE
027374  012737  002304  003400          MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE INFO
027402  013765  003400  000004          MOV     TEMP4,RKBA(R5)
027410  012737  001000  003402          MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
027416  013765  003402  000006          MOV     TEMP5,RKDA(R5)
027424  012765  000632  000020  1$:     MOV     #410.,RKDC(R5)  ;CYL 410
027432  012765  177400  000002          MOV     #-256.,RKWC(R5) ;LOAD WORD CT
027440  012765  000021  000000          MOV     #RDDATA,RKCS1(R5);READ DATA CMD
027446  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
027454  004737  043612                  JSR     PC,FRDY         ;FIND RDY
027460  104226                  1$:     ERROR   226             ;NO RDY AFTER READ DATA CMD
027462  004737  045150                  JSR     PC,GSTAT        ;GET FRESH STATUS
027466  032737  100000  003334          BIT     #CERR,HCS1
027474  001470                          BEQ     8$
027476  104227                  1$:     ERROR   227             ;CERR AFTER READ DATA CMD
027500  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
027506  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
027512  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
027520  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
027526  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
027532  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
027540  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
```

```
6762  027546  004737  044334          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6763  027552  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6764  027554  104051                  ERROR   51              ;MSG A0 ERROR AFTER READ DATA CMD
6765  027556  104052                  ERROR   52              ;MSG B0 ERROR
6766  027560  104112                  ERROR   112             ;MSG A1 ERROR
6767  027562  104113                  ERROR   113             ;MSG B1 ERROR
6768
6769  027564  004737  045522          JSR     PC,SUBCLR
6770  027570  104024                  ERROR   24              ;CERR AFTER SUBCLR
6771
6772  027572  005237  003374          INC     TEMP2
6773  027576  023727  003374  000005  CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
6774  027604  001007                  BNE     5$
6775  027606  005737  003376          TST     TEMP3
6776  027612  001002                  BNE     2$
6777  027614  104233                  ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
6778  027616  000414                  BR      3$
6779  027620  104230          2$:     ERROR   230             ;CANT READ SECTORS 10,12,14,16,18,20
6780  027622  000412                  BR      3$
6781
6782  027624  013765  003400  000004  5$:     MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
6783  027632  062737  000002  003402  ADD     #2,TEMP5        ;SETUP TO READ 2 SECTORS FROM LAST
6784  027640  013765  003402  000006  MOV     TEMP5,RKDA(R5)
6785  027646  000666                  BR      1$
6786
6787  027650  005237  001456          3$:     INC     BSERR           ;SET BSE FLAG
6788  027654  000454                  BR      TST33           ;;GO TO NEXT TEST
6789
6790  027656  005737  002312          8$:     TST     BSE22H+6        ;TEST CARTRIDGE TYPE
6791  027662  001404                  BEQ     9$              ;BRANCH IF DATA CARTRIDGE
6792  027664  104235                  ERROR   235             ;ALIGNMENT CARTRIDGE USED
6793  027666  005237  001456          INC     BSERR           ;SET BSE ERROR FLAG
6794  027672  000426                  BR      10$
6795
6796  027674  005237  003376          9$:     INC     TEMP3
6797  027700  023727  003376  000001  CMP     TEMP3,#1
6798  027706  001020                  BNE     10$
6799  027710  005037  003374          CLR     TEMP2
6800  027714  012737  054472  003400  MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
6801  027722  013765  003400  000004  MOV     TEMP4,RKBA(R5)
6802  027730  012737  001012  003402  MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12
6803  027736  013765  003402  000006  MOV     TEMP5,RKDA(R5)
6804  027744  000137  027424          JMP     1$              ;REPEAT
6805
6806  027750  005737  001216          10$:    TST     $PASS
6807  027754  001014                  BNE     TST33           ;;GO TO NEXT TST IF NOT 1'ST PASS
6808  027756  104401  056304          TYPE    ,MSG17          ;CART SERIAL #
6809  027762  012746  002304          MOV     #BSE22H,-(SP)
6810  027766  004737  053676          JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
6811  027772  004737  054246          JSR     PC,$SUPRS       ;TYPE SERIAL #
6812  027776  104401  001205          TYPE    ,$CRLF
6813  030002  104401  001205          TYPE    ,$CRLF
6814
6815                                  ;;*****************************************************************
6816                                  ;*TEST 33        DETECT INNER LIMIT
6817                                  ;*
```

```
                                        ;*      THIS TEST VERIFIES THAT THE LAST CYL IN THE ABOVE
                                        ;*      TEST WAS 410 BY DETECTING INNER LIMIT AS THE ADJACENT CYL.
                                        ;*      IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON CYL 410
                                        ;*      & THAT BSE INFO IS NOT VALID.  THE FORMAT PACK TEST
                                        ;*      & ALL READ-WRITE TESTS ARE BYPASSED
                                        ;*      TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
                                        ;*      SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.
                                        ;*
                                        ;**************************************************************
030006  000004                  TST33:  SCOPE
030010  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
030016  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

030022  004737  045522                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
030026  104024                          ERROR   24              ;CERR AFTER SCLR

030030  005037  001410                  CLR     LPFLG
030034  005237  001462                  INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
030040  005237  003316                  INC     UNLD            ;USED FOR VALID HALT

030044  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
030052  012765  000631  000020          MOV     #409.,RKDC(R5)  ;CYL 409.
030060  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
030066  013737  001414  003372          MOV     T10,TEMP1
030074  004737  043612                  JSR     PC,FRDY         ;FIND RDY
030100  104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
030102  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
030106  104125                          ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
030110  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
030116  012737  001200  003426          MOV     #<D.FLT!D.PAR>,E.B0
030124  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
030132  012737  000001  003432          MOV     #1,E.B1

030140  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
030144  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
030146  104110                          ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
030150  104111                          ERROR   111             ;MSG B0 ERROR
030152  104146                          ERROR   146             ;MSG A1 ERROR
030154  104147                          ERROR   147             ;MSG B1 ERROR

030156  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
030164  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
030172  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
030200  013737  001414  003372          MOV     T10,TEMP1
030206  004737  043612                  JSR     PC,FRDY         ;FIND RDY
030212  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
030214  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
030220  000401                          BR      64$
030222  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
030224                          64$:

030224  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
030232  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
030236  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
030244  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
030252  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
```

```
6874  030256  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6875  030264  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6876
6877  030272  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6878  030276  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6879  030300  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6880  030302  104265                          ERROR   265             ;MSG B0 ERROR
6881  030304  104274                          ERROR   274             ;MSG A1 ERROR
6882  030306  104266                          ERROR   266             ;MSG B1 ERROR
6883
6884
6885  030310  012765  000632  000020          MOV     #410.,RKDC(R5)  ;CYL 410.
6886  030316  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL 410.
6887  030324  013737  001414  003372          MOV     T10,TEMP1
6888  030332  004737  043612                  JSR     PC,FRDY         ;FIND RDY
6889  030336  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
6890  030340  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6891  030346  004737  045150                  JSR     PC,GSTAT
6892  030352  004737  046356                  JSR     PC,FLIM         ;FIND LIMIT DETECT
6893  030356  104160                          ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
6894
6895  030360  032737  040000  003362          BIT     #D.UNLD,HMR2
6896  030366  001003                          BNE     1$
6897  030370  104305                          ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
6898  030372  000137  031234                  JMP     30$             ;BYPASS REST OF TEST
6899
6900  030376  012737  031144  001176  1$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
6901  030404  012737  070140  003424          MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
6902  030412  012737  002200  003426          MOV     #<D.SKI!D.FLT>,E.B0
6903  030420  012737  045720  003430          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6904  030426  012737  030001  003432          MOV     #<D.LIMD!D.NMOV!1>,E.B1
6905
6906  030434  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6907  030440  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6908  030442  104161                          ERROR   161             ;MSG A0 ERROR AFTER INNER LIMIT DETECT
6909  030444  104162                          ERROR   162             ;MSG B0 ERROR
6910  030446  104163                          ERROR   163             ;MSG A1 ERROR
6911  030450  104164                          ERROR   164             ;MSG B1 ERROR
6912
6913  030452  004737  044074                  JSR     PC,TSTATN
6914  030456  104165                          ERROR   165             ;NO ATTN AFTER INNER LIMIT DETECT
6915  030460  005037  001462                  CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1
6916
6917  030464  004737  045522                  JSR     PC,SUBCLR       ;SUBSYS CLR
6918  030470  104024                          ERROR   24              ;CERR AFTER SCLR
6919  030472  013737  001414  003374          MOV     T10,TEMP2       ;SET UP TIMEOUT
6920  030500  004737  046434                  JSR     PC,FHDHM        ;FIND HEAD HOME
6921  030504  104166                          ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
6922  030506  004737  046510                  JSR     PC,FLOAD        ;FIND LOAD HEADS
6923  030512  104167                          ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
6924  030514  013737  001416  003374          MOV     T100,TEMP2      ;SETUP TIMEOUT
6925  030522  004737  044126                  JSR     PC,FATT1        ;FIND ATTN
6926  030526  104067                          ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
6927  030530  005037  001176          2$:     CLR     $ESCAPE
6928  030534  005037  003316                  CLR     UNLD            ;CLEAR FLAG
6929
```

```
 6930  030540  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 6931  030546  005037  003426                  CLR     E.B0                    ;EXPECTED MSG B0
 6932  030552  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6933  030560  012737  000001  003432          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
 6934  030566  005037  003434                  CLR     E.A2                    ;EXPECTED MSG A2
 6935  030572  012737  000002  003436          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
 6936  030600  012737  000003  003442          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
 6937
 6938  030606  004737  044334                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
 6939  030612  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
 6940  030614  104063                          ERROR   63                      ;MSG A0 ERROR AT END OF HEAD LOADING
 6941  030616  104064                          ERROR   64                      ;MSG B0 ERROR
 6942  030620  104065                          ERROR   65                      ;MSG A1 ERROR
 6943  030622  104066                          ERROR   66                      ;MSG B1 ERROR
 6944
 6945  030624  005737  001362                  TST     CYLDIF                  ;SEE IF MSG A2=0
 6946  030630  001401                          BEQ     65$                     ;BR IF YES
 6947  030632  104175                          ERROR   175                     ;MSG A2 NOT CLEARED AT END OF HEAD LOADING
 6948  030634  005737  001364          65$:    TST     CYLADD                  ;SEE IF MSG B2=0
 6949  030640  001401                          BEQ     66$                     ;BR IF YES
 6950  030642  104176                          ERROR   176                     ;MSG B2 NOT CLEARED AT END OF HEAD LOADING
 6951  030644                          66$:
 6952
 6953  030644  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6954  030652  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 6955  030660  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 6956  030666  013737  001414  003372          MOV     T10,TEMP1
 6957  030674  004737  043612                  JSR     PC,FRDY                 ;FIND RDY
 6958  030700  104151                          ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
 6959  030702  004737  044074                  JSR     PC,TSTATN               ;TEST FOR ATTN
 6960  030706  000401                          BR      67$
 6961  030710  104154                          ERROR   154                     ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 6962  030712                          67$:
 6963
 6964  030712  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 6965  030720  005037  003426                  CLR     E.B0                    ;EXPECTED MSG B0
 6966  030724  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6967  030732  012737  000001  003432          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
 6968  030740  005037  003434                  CLR     E.A2                    ;EXPECTED MSG A2
 6969  030744  012737  000002  003436          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
 6970  030752  012737  000003  003442          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
 6971
 6972  030760  004737  044334                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
 6973  030764  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
 6974  030766  104273                          ERROR   273                     ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 6975  030770  104265                          ERROR   265                     ;MSG B0 ERROR
 6976  030772  104274                          ERROR   274                     ;MSG A1 ERROR
 6977  030774  104266                          ERROR   266                     ;MSG B1 ERROR
 6978
 6979  030776  004737  047526                  JSR     PC,SWTST                ;SEE IF SW 14 OR 8 IS SET
 6980  031002  000514                          BR      TST34                   ;;GO TO NEXT TEST
 6981                                                                          ;RETURN HERE IF SW 14 IS SET OR
 6982                                                                          ;SW 8 WITH SWR <7:0> APPLY
 6983
 6984
 6985  031004                          10$:
```

```

 031004  004737  045522                    JSR     PC,SUBCLR
 031010  104024                            ERROR   24                 ;CERR AFTER SCRL

 031012  013765  001352  000020    68$:    MOV     TOCYL,RKDC(R5)     ;CYL#

 031020  012765  000017  000000            MOV     #SEEK,RKCS1(R5)    ;SEEK CMD TO RECONDITION DRIVE.
 031026  013737  001414  003372            MOV     T10,TEMP1          ;SETUP TIMEOUT
 031034  004737  043612                    JSR     PC,FRDY            ;FIND RDY
 031040  104131                            ERROR   131                ;NO RDY AFTER SEEK CMD.

 031042  013737  001426  003372            MOV     T50000,TEMP1
 031050  004737  044222                    JSR     PC,FATT2           ;FIND ATTN
 031054  104132                            ERROR   132                ;NO ATTN AFTER SEEK CMD
 031056  032737  100000  003334            BIT     #CERR,HCS1
 031064  001401                            BEQ     70$
 031066  104210                            ERROR   210                ;CERR AFTER SEEK CMD.

 031070  004737  045522            70$:    JSR     PC,SUBCLR
 031074  104024                            ERROR   24                 ;CERR AFTER SCLR

 031076  023727  001352  000632            CMP     TOCYL,#410.        ;ALL CYL DONE?
 031104  001403                            BEQ     69$                ;BR IF YES
 031106  005237  001352                    INC     TOCYL              ;ELSE DO ANOTHER
 031112  000737                            BR      68$

 031114  004737  045522            69$:    JSR     PC,SUBCLR
 031120  104024                            ERROR   24                 ;CERR AFTER SCLR

 031122  005037  001176                    CLR     $ESCAPE
 031126  005737  001410                    TST     LPFLG
 031132  001402                            BEQ     71$
 031134  000177  147750                    JMP     @$LPERR            ;SW 9 WAS SET.
 031140  000177  147742            71$:    JMP     @$LPADR            ;SW 14 OR 8 WAS SET

 031144                            20$:

 031144  004737  045522                    JSR     PC,SUBCLR
 031150  104024                            ERROR   24                 ;CERR AFTER SCLR

 031152  012765  000011  000000            MOV     #SRTSPL,RKCS1(R5)          ;START SPINDLE CMD
 031160  013737  001414  003372            MOV     T10,TEMP1          ;SET TIMEOUT
 031166  004737  043612                    JSR     PC,FRDY            ;FIND RDY
 031172  104121                            ERROR   121                ;RDY NOT FOUND AFTER ST SPIN CMD.

 031174  013737  001420  003374            MOV     T500,TEMP2         ;SETUP  TIMEOUT
 031202  004737  044126                    JSR     PC,FATT1           ;FIND ATTN
 031206  104067                            ERROR   67                 ;NO ATTN AFTER ST SPIN CMD.

 031210  005037  003316                    CLR     UNLD
 031214  005237  001410                    INC     LPFLG
 031220  032777  001000  147712            BIT     #SW9,@SWR          ;LOOP ON ERROR?
 031226  001266                            BNE     10$                ;YES, RECONDITION DRIVE
 031230  000137  030530                    JMP     2$                 ;RETURN TO MAINLINE
 031234                            30$:

```

```
031234                                  FORM:
                                        ;;*****************************************************************
                                        ;*TEST 34       FORMAT PACK
                                        ;*
                                        ;*      THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
                                        ;*      DOING 1 CYL INCREMENTAL SEEKS
                                        ;*      FROM 0 TO 410 WITH WRITE HEADER CMDS (ALL TRACKS).
                                        ;*      HEADERS WILL BE READ IN THE NEXT TEST
                                        ;*
                                        ;;*****************************************************************
031234  000004                  TST34:  SCOPE
031236  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION

031244  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
031250  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
031254  001402                          BEQ     22$             ;BR IF NO
031256  104401  056417                  TYPE    ,MSG20          ;RUNNING MODIFIED VERSION OF TEST

031262  005737  001460          22$:    TST     LIMERR          ;CHECK IF FOUND LIMIT DETECT ERROR
031266  001403                          BEQ     1$
031270  104170                          ERROR   170             ;FATAL ERROR
031272  000137  043076                  JMP     $EOP            ;ABORT BAL OF TESTS
031276  005737  001456          1$:     TST     BSERR           ;CHECK IF FOUND BSE INFO OK
031302  001403                          BEQ     2$
031304  104177                          ERROR   177             ;FORMAT TEST BYPASSED-BSE ERROR
031306  000137  033356                  JMP     13$

031312  004737  045522          2$:     JSR     PC,SUBCLR
031316  104024                          ERROR   24              ;CERR AFTER SCLR

031320  104401  056117                  TYPE    ,MSG12          ;FORMATTING PACK, PLEASE WAIT

031324  005037  001352                  CLR     TOCYL

031330  013737  001352  001366          MOV     TOCYL,CALADD    ;SETUP
031336  012737  000000  001430          MOV     #0,HEAD         ;TO FILL
031344  012737  000000  001436          MOV     #0,FORMAT       ;HEADER
031352  004737  046632                  JSR     PC,FHDTAB       ;TABLE

031356  012765  001470  000004  9$:     MOV     #HDTAB,RKBA(R5) ;THIS SECTION
031364  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;OF CODE
031372  000337  001430                  SWAB    HEAD            ;IS TO RESTORE STANDARD FORMAT
031376  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;TO CYL 0
031404  000337  001430                  SWAB    HEAD            ;HEAD 0,1 & 2

031410  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
031416  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
031424  004737  043612                  JSR     PC,FRDY         ;FIND RDY
031430  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
031432  004737  045150                  JSR     PC,GSTAT        ;GET FRESH STATUS
031436  032737  100000  003334          BIT     #CERR,HCS1
031444  001405                          BEQ     64$
031446  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
031450  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
```

```
7098  031454  000137  043076                  JMP     $EOP            ;ABORT DRIVE
7099  031460                          64$:
7100
7101  031460  005237  001430                  INC     HEAD
7102  031464  023727  001430  000003          CMP     HEAD,#3
7103  031472  001403                          BEQ     11$             ;BR IF ALL HEADS DONE
7104
7105  031474  004737  046632                  JSR     PC,FHDTAB
7106  031500  000726                          BR      9$
7107
7108  031502  012737  000001  001366  11$:    MOV     #1,CALADD       ;SETUP
7109  031510  005037  001430                  CLR     HEAD            ;FOR
7110  031514  005037  001436                  CLR     FORMAT          ;FHDTAB ROUTINE
7111
7112  031520  012737  000001  001360          MOV     #1,CALDIF       ;SETUP
7113  031526  005037  001350                  CLR     FRCYL           ;FOR
7114  031532  012737  000001  001352          MOV     #1,TOCYL        ;ERROR REPORT
7115                                                                  ;START FORMATTING CYL 1 TO 410 HERE
7116
7117
7118  031540                          3$:
7119  031540  104415                          SCOP1
7120  031542  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7121
7122  031546  004737  045522                  JSR     PC,SUBCLR
7123  031552  104024                          ERROR   24              ;CERR AFTER SCLR
7124
7125  031554  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
7126  031560  001404                          BEQ     18$             ;BR IF NO
7127  031562  012737  033316  001176          MOV     #16$,$ESCAPE
7128  031570  000403                          BR      19$
7129  031572  012737  033006  001176  18$:    MOV     #10$,$ESCAPE
7130  031600  013765  001366  000020  19$:    MOV     CALADD,RKDC(R5) ;CYL #
7131  031606  000337  001430                  SWAB    HEAD
7132  031612  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;HEAD #
7133  031620  000337  001430                  SWAB    HEAD
7134
7135  031624  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7136  031632  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
7137  031640  004737  043612                  JSR     PC,FRDY         ;FIND RDY
7138  031644  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7139  031646  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.AO ;EXPECTED AO
7140  031654  005037  003426                  CLR     E.BO
7141  031660  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7142  031666  012737  000001  003432          MOV     #1,E.B1
7143
7144  031674  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS AO,BO,A1,B1
7145  031700  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7146  031702  104203                          ERROR   203             ;MSG AO ERROR DURING SEEK CMD
7147  031704  104204                          ERROR   204             ;MSG BO ERROR
7148  031706  104205                          ERROR   205             ;MSG A1 ERROR
7149  031710  104206                          ERROR   206             ;MSG B1 ERROR
7150
7151  031712  023727  001362  000001          CMP     CYLDIF,#1
7152  031720  001401                          BEQ     4$
7153  031722  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
```

```
7154
7155  031724  005737  001342          4$:     TST     MODTST          ;SEE IF MODULE TESTING
7156  031730  001404                          BEQ     20$             ;BR IF NO
7157  031732  012737  033336  001176          MOV     #17$,$ESCAPE
7158  031740  000403                          BR      21$
7159
7160  031742  012737  033026  001176  20$:    MOV     #12$,$ESCAPE
7161  031750  012737  004704  003372  21$:    MOV     #2500.,TEMP1    ;SETUP TIMEOUT
7162
7163  031756  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
7164  031762  -104132                         ERROR   132             ;NO ATTN AFTER SEEK CMD
7165  031764  032737  100000  003334          BIT     #CERR,HCS1
7166  031772  001401                          BEQ     65$
7167  031774  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7168  031776                          65$:
7169
7170  031776  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7171  032004  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7172  032010  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7173  032016  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7174  032024  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7175  032030  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7176  032036  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7177
7178  032044  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7179  032050  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7180  032052  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7181  032054  104134                          ERROR   134             ;MSG B0 ERROR
7182  032056  104135                          ERROR   135             ;MSG A1 ERROR
7183  032060  104136                          ERROR   136             ;MSG B1 ERROR
7184  032062  005737  001362                  TST     CYLDIF
7185  032066  001401                          BEQ     66$
7186  032070  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7187
7188  032072                          66$:
7189
7190  032072  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7191  032100  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7192  032106  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7193  032114  013737  001414  003372          MOV     T10,TEMP1
7194  032122  004737  043612                  JSR     PC,FRDY         ;FIND RDY
7195  032126  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7196  032130  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
7197  032134  000401                          BR      67$
7198  032136  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7199  032140                          67$:
7200
7201  032140  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
7202  032146  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7203  032152  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7204  032160  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7205  032166  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7206  032172  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7207  032200  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7208
7209  032206  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
```

```
7210  032212  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7211  032214  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7212  032216  104265                          ERROR   265             ;MSG B0 ERROR
7213  032220  104274                          ERROR   274             ;MSG A1 ERROR
7214  032222  104266                          ERROR   266             ;MSG B1 ERROR
7215
7216  032224  023737  001364  001366          CMP     CYLADD,CALADD
7217  032232  001401                          BEQ     5$
7218  032234  104232                          ERROR   232             ;CYL ADDR IN RKMR2 NOT=RKDC
7219
7220  032236                          5$:
7221  032236  104415                          SCOP1
7222  032240  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7223
7224  032244  004737  045522                  JSR     PC,SUBCLR
7225  032250  104024                          ERROR   24              ;CERR AFTER SCLR
7226
7227  032252  005037  001176                  CLR     $ESCAPE
7228  032256  004737  046632                  JSR     PC,FHDTAB       ;FILL HEADER TABLE
7229  032262  000337  001430                  SWAB    HEAD
7230  032266  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;SET TRACK #
7231  032274  000337  001430                  SWAB    HEAD
7232  032300  012765  001470  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
7233  032306  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD CT
7234  032314  013765  001366  000020          MOV     CALADD,RKDC(R5) ;CYL #
7235
7236
7237  032322  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
7238  032330  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
7239  032336  004737  043612                  JSR     PC,FRDY         ;FIND RDY
7240  032342  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
7241  032344  004737  045150                  JSR     PC,GSTAT        ;GET FRESH STATUS
7242  032350  032737  100000  003334          BIT     #CERR,HCS1
7243  032356  001405                          BEQ     68$
7244  032360  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
7245  032362  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7246  032366  000137  043076                  JMP     $EOP            ;ABORT DRIVE
7247  032372                          68$:
7248
7249  032372  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7250  032400  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7251  032404  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7252  032412  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7253  032420  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7254  032424  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7255  032432  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7256
7257  032440  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7258  032444  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7259  032446  104277                          ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
7260  032450  104267                          ERROR   267             ;MSG B0 ERROR
7261  032452  104300                          ERROR   300             ;MSG A1 ERROR
7262  032454  104270                          ERROR   270             ;MSG B1 ERROR
7263
7264
7265  032456  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
```

```
7266  032462  001402                        BEQ     23$             ;BR IF NO
7267  032464  000137  033046                JMP     14$             ;ELSE RESTORE HEADERS ONLY
7268
7269  032470  005237  001430        23$:    INC     HEAD
7270  032474  023727  001430  000002        CMP     HEAD,#2
7271  032502  001006                        BNE     6$
7272  032504  023727  001366  000632        CMP     CALADD,#410.    ;HEAD 2, SEE IF CYL 410
7273  032512  001002                        BNE     6$              ;DO NOT WRITE ON CYL 410 HEAD 2
7274  032514  000137  032746                JMP     7$
7275
7276  032520  023727  001430  000003  6$:   CMP     HEAD,#3         ;ALL HEADS DONE?
7277  032526  001243                        BNE     5$              ;BR IF NO
7278  032530  005037  001430                CLR     HEAD            ;ALL HEADS ON CYL DONE
7279  032534  005237  001366                INC     CALADD          ;GO TO NEXT CYL
7280  032540  005237  001350                INC     FRCYL           ;FOR ERROR REPORT
7281  032544  005237  001352                INC     TOCYL           ;FOR ERROR REPORT
7282  032550  005737  003322                TST     HPEND           ;SEE IF HALT PENDING
7283  032554  001002                        BNE     24$             ;BR IF YES
7284  032556  000137  031540                JMP     3$              ;ELSE KEEP FORMATTING
7285
7286  032562  005037  003322        24$:    CLR     HPEND           ;CLEAR FOR FUTURE FORMATTING
7287  032566  005037  003320                CLR     BADHDR          ;HEADERS NOW OK
7288  032572  000137  047576                JMP     STOP            ;GO & HALT THE CPU
7289
7290  032576  005037  001366        8$:     CLR     CALADD
7291  032602  005037  001350                CLR     FRCYL
7292
7293  032606  004737  045522                JSR     PC,SUBCLR
7294  032612  104024                        ERROR   24              ;CERR AFTER SCRL
7295
7296  032614  013765  001352  000020  69$:   MOV     TOCYL,RKDC(R5)  ;CYL#
7297
7298  032622  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7299  032630  013737  001414  003372        MOV     T10,TEMP1       ;SETUP TIMEOUT
7300  032636  004737  043612                JSR     PC,FRDY         ;FIND RDY
7301  032642  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
7302
7303  032644  013737  001426  003372        MOV     T50000,TEMP1
7304  032652  004737  044222                JSR     PC,FATT2        ;FIND ATTN
7305  032656  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
7306  032660  032737  100000  003334        BIT     #CERR,HCS1
7307  032666  001401                        BEQ     71$
7308  032670  104210                        ERROR   210             ;CERR AFTER SEEK CMD.
7309
7310  032672  004737  045522        71$:    JSR     PC,SUBCLR
7311  032676  104024                        ERROR   24              ;CERR AFTER SCLR
7312
7313  032700  023727  001352  000000        CMP     TOCYL,#0        ;ALL CYL DONE?
7314  032706  001403                        BEQ     70$             ;BR IF YES
7315  032710  005337  001352                DEC     TOCYL           ;ELSE DO ANOTHER
7316  032714  000737                        BR      69$
7317
7318  032716  004737  045522        70$:    JSR     PC,SUBCLR
7319  032722  104024                        ERROR   24              ;CERR AFTER SCLR
7320
7321  032724  005037  001176                CLR     $ESCAPE
```

```
7322  032730  005737  001410          TST     LPFLG
7323  032734  001402                  BEQ     72$
7324  032736  000177  146146          JMP     @$LPERR         ;SW 9 WAS SET.
7325  032742  000177  146140  72$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7326
7327
7328  032746  004737  050126  7$:     JSR     PC,HPEN         ;SEE IF HALT PENDING
7329  032752  000137  032562          JMP     24$             ;RET HERE IF YES & EXIT
7330                                                          ;ELSE RET HERE
7331  032756  004737  047526          JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7332  032762  000575                  BR      TST35           ;;GO TO NEXT TEST
7333                                                          ;RETURN HERE IF SW 14 IS SET OR
7334                                                          ;SW 8 WITH SWR <7:0> APPLY
7335  032764  005037  001176          CLR     $ESCAPE
7336  032770  005737  001410          TST     LPFLG
7337  032774  001402                  BEQ     73$
7338  032776  000177  146106          JMP     @$LPERR         ;SW 9 WAS SET.
7339  033002  000177  146100  73$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7340
7341  033006                  10$:
7342  033006  005237  001410          INC     LPFLG
7343  033012  032777  001000  146120  BIT     #SW9,@SWR       ;LOOP ON ERROR?
7344  033020  001266                  BNE     8$              ;YES, RECONDITION DRIVE
7345  033022  000137  031724          JMP     4$              ;RETURN TO MAINLINE
7346
7347  033026                  12$:
7348  033026  005237  001410          INC     LPFLG
7349  033032  032777  001000  146100  BIT     #SW9,@SWR       ;LOOP ON ERROR?
7350  033040  001256                  BNE     8$              ;YES, RECONDITION DRIVE
7351  033042  000137  032236          JMP     5$              ;RETURN TO MAINLINE
7352
7353
7354  033046  005237  001430  14$:    INC     HEAD
7355  033052  023727  001430  000003  CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
7356  033060  001402                  BEQ     15$             ;BR IF YES TO GO BACK TO CYL 0
7357  033062  000137  032236          JMP     5$              ;ELSE REPEAT FOR NEXT HEAD
7358
7359  033066  005065  000006  15$:    CLR     RKDA(R5)        ;SEEK TO CYL 0 & READ HEADERS
7360  033072  005037  001352          CLR     TOCYL           ;TO RECONDITION DRIVE
7361
7362  033076  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7363  033104  013737  001414  003372  MOV     T10,TEMP1       ;SETUP TIMEOUT
7364  033112  004737  043612          JSR     PC,FRDY         ;FIND RDY
7365  033116  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD.
7366
7367  033120  013737  001426  003372  MOV     T50000,TEMP1
7368  033126  004737  044222          JSR     PC,FATT2        ;FIND ATTN
7369  033132  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
7370  033134  032737  100000  003334  BIT     #CERR,HCS1
7371  033142  001401                  BEQ     74$
7372  033144  104210                  ERROR   210             ;CERR AFTER SEEK CMD.
7373
7374  033146  004737  045522  74$:    JSR     PC,SUBCLR
7375  033152  104024                  ERROR   24              ;CERR AFTER SCLR
7376
7377
```

```
7378
7379  033154  012700  001674                MOV     #RHTAB,R0
7380  033160  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7381  033166  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7382  033174  004737  043612                JSR     PC,FRDY         ;FIND RDY
7383  033200  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
7384  033202  032737  100000  003334        BIT     #CERR,HCS1
7385  033210  001405                        BEQ     76$
7386  033212  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
7387  033214  104401  056333                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7388  033220  000137  043076                JMP     $EOP            ;ABORT DRIVE
7389
7390  033224  016520  000024        76$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7391  033230  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
7392  033234  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
7393
7394
7395  033240  032765  100000  000010        BIT     #DLT,RKCS2(R5)
7396  033246  001407                        BEQ     77$
7397  033250  004737  045150                JSR     PC,GSTAT
7398  033254  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
7399  033256  104401  056333                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7400  033262  000137  043076                JMP     $EOP            ;ABORT DRIVE
7401  033266                        77$:
7402
7403  033266  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7404  033274  001401                        BEQ     75$             ;BR IF SAME
7405  033276  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
7406  033300                        75$:
7407
7408  033300  004737  050126                JSR     PC,HPEN         ;SEE IF HALT PENDING
7409  033304  000137  032562                JMP     24$             ;RET HERE IF YES
7410                                                                ;ELSE RET HERE & EXIT
7411  033310  004737  047526                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7412  033314  000420                        BR      TST35           ;;GO TO NEXT TEST
7413                                                                ;RETURN HERE IF SW 14 IS SET OR
7414                                                                ;SW 8 WITH SWR <7:0> APPLY
7415  033316                        16$:
7416  033316  005237  001410                INC     LPFLG
7417  033322  032777  001000  145610        BIT     #SW9,@SWR       ;LOOP ON ERROR?
7418  033330  001256                        BNE     15$             ;YES, RECONDITION DRIVE
7419  033332  000137  031724                JMP     4$              ;RETURN TO MAINLINE
7420  033336                        17$:
7421  033336  005237  001410                INC     LPFLG
7422  033342  032777  001000  145570        BIT     #SW9,@SWR       ;LOOP ON ERROR?
7423  033350  001246                        BNE     15$             ;YES, RECONDITION DRIVE
7424  033352  000137  032236                JMP     5$              ;RETURN TO MAINLINE
7425
7426
7427  033356                        13$:
7428
7429
7430
7431                                ;;*****************************************************************
7432                                ;*TEST 35       DECREMENT FROM CYL 410 TO 0 & READ HEADERS
7433                                ;*
```

```
                                ;*      THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
                                ;*      SINGLE CYL INCREMENTAL SEEKS.
                                ;*
                                ;;*********************************************************************
033356  000004                TST35:  SCOPE
033360  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
033366  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

033372  104401  056646                TYPE    ,MSG22          ;FORMATTING FINISHED
033376  005737  001342                TST     MODTST          ;SEE IF MODULE TESTING
033402  001404                        BEQ     5$              ;BR IF NO
033404  104401  056566                TYPE    ,MSG21          ;BYP TESTS 36,40,41
033410  000137  034452                JMP     13$
033414  012737  000632  001350  5$:   MOV     #410.,FRCYL     ;FROM CYL
033422  012737  000631  001352        MOV     #409.,TOCYL     ;TO CYL

033430                          1$:
033430  104415                        SCOP1
033432  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

033436  004737  045522                JSR     PC,SUBCLR
033442  104024                        ERROR   24              ;CERR AFTER SCLR

033444  012737  034412  001176        MOV     #10$,$ESCAPE
033452  013765  001352  000020        MOV     TOCYL,RKDC(R5)  ;CYL #

033460  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
033466  013737  001414  003372        MOV     T10,TEMP1       ;SETUP  TIMEOUT
033474  004737  043612                JSR     PC,FRDY         ;FIND RDY
033500  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
033502  012737  030140  003424        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
033510  005037  003426                CLR     E.B0
033514  012737  005720  003430        MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
033522  012737  000001  003432        MOV     #1,E.B1

033530  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
033534  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
033536  104203                        ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
033540  104204                        ERROR   204             ;MSG B0 ERROR
033542  104205                        ERROR   205             ;MSG A1 ERROR
033544  104206                        ERROR   206             ;MSG B1 ERROR

033546  023727  001362  000001        CMP     CYLDIF,#1
033554  001401                        BEQ     2$
033556  104212                        ERROR   212             ;CYL DIFF INCORRECT DURING SEEK

033560  012737  034432  001176  2$:   MOV     #12$,$ESCAPE
033566  012737  004704  003372        MOV     #2500.,TEMP1    ;SETUP TIMEOUT

033574  004737  044222                JSR     PC,FATT2        ;FIND ATTN
033600  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
033602  032737  100000  003334        BIT     #CERR,HCS1
033610  001401                        BEQ     64$
033612  104210                        ERROR   210             ;CERR AFTER SEEK CMD
033614                          64$:
```

```
7490  033614  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7491  033622  005037  003426                  CLR     E.B0                    ;EXPECTED MSG B0
7492  033626  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7493  033634  012737  000001  003432          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
7494  033642  005037  003434                  CLR     E.A2                    ;EXPECTED MSG A2
7495  033646  012737  000002  003436          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
7496  033654  012737  000003  003442          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
7497
7498  033662  004737  044334                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
7499  033666  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
7500  033670  104133                          ERROR   133                     ;MSG A0 ERROR AFTER SEEK CMD
7501  033672  104134                          ERROR   134                     ;MSG B0 ERROR
7502  033674  104135                          ERROR   135                     ;MSG A1 ERROR
7503  033676  104136                          ERROR   136                     ;MSG B1 ERROR
7504  033700  005737  001362                  TST     CYLDIF
7505  033704  001401                          BEQ     65$
7506  033706  104137                          ERROR   137                     ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7507
7508  033710                          65$:
7509
7510  033710  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7511  033716  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7512  033724  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)                ;DRIVE CLEAR CMD
7513  033732  013737  001414  003372          MOV     T10,TEMP1
7514  033740  004737  043612                  JSR     PC,FRDY                 ;FIND RDY
7515  033744  104151                          ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
7516  033746  004737  044074                  JSR     PC,TSTATN               ;TEST FOR ATTN
7517  033752  000401                          BR      66$
7518  033754  104154                          ERROR   154                     ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7519  033756                          66$:
7520
7521  033756  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7522  033764  005037  003426                  CLR     E.B0                    ;EXPECTED MSG B0
7523  033770  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7524  033776  012737  000001  003432          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
7525  034004  005037  003434                  CLR     E.A2                    ;EXPECTED MSG A2
7526  034010  012737  000002  003436          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
7527  034016  012737  000003  003442          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
7528
7529  034024  004737  044334                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
7530  034030  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
7531  034032  104273                          ERROR   273                     ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7532  034034  104265                          ERROR   265                     ;MSG B0 ERROR
7533  034036  104274                          ERROR   274                     ;MSG A1 ERROR
7534  034040  104266                          ERROR   266                     ;MSG B1 ERROR
7535
7536  034042  023737  001364  001352          CMP     CYLADD,TOCYL
7537  034050  001401                          BEQ     3$
7538  034052  104207                          ERROR   207                     ;CYL ADDR IN RKMR3 NOT = RKDC
7539
7540  034054                          3$:
7541  034054  104415                          SCOP1
7542  034056  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR
7543
7544  034062  004737  045522                  JSR     PC,SUBCLR
7545  034066  104024                          ERROR   24                      ;CERR AFTER SCLR
```

```
7546
7547  034070  005037  001176                CLR     $ESCAPE
7548
7549  034074  013765  001352  000020        MOV     TOCYL,RKDC(R5)  ;CYL #
7550
7551
7552  034102  012700  001674                MOV     #RHTAB,R0
7553  034106  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7554  034114  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7555  034122  004737  043612                JSR     PC,FRDY         ;FIND RDY
7556  034126  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
7557  034130  032737  100000  003334        BIT     #CERR,HCS1
7558  034136  001405                        BEQ     68$
7559  034140  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
7560  034142  104401  056333                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7561  034146  000137  043076                JMP     $EOP            ;ABORT DRIVE
7562
7563  034152  016520  000024        68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7564  034156  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
7565  034162  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
7566
7567
7568  034166  032765  100000  000010        BIT     #DLT,RKCS2(R5)
7569  034174  001407                        BEQ     69$
7570  034176  004737  045150                JSR     PC,GSTAT
7571  034202  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
7572  034204  104401  056333                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7573  034210  000137  043076                JMP     $EOP            ;ABORT DRIVE
7574  034214                        69$:
7575
7576  034214  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7577  034222  001401                        BEQ     67$             ;BR IF SAME
7578  034224  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
7579  034226                        67$:
7580
7581  034226  005337  001350                DEC     FRCYL
7582  034232  001404                        BEQ     4$
7583  034234  005337  001352                DEC     TOCYL
7584  034240  000137  033430                JMP     1$
7585
7586  034244                        4$:
7587  034244  004737  047526                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7588  034250  000500                        BR      TST36           ;;GO TO NEXT TEST
7589                                                                ;RETURN HERE IF SW 14 IS SET OR
7590                                                                ;SW 8 WITH SWR <7:0> APPLY
7591
7592
7593  034252                        6$:
7594
7595  034252  004737  045522                JSR     PC,SUBCLR
7596  034256  104024                        ERROR   24              ;CERR AFTER SCRL
7597
7598  034260  013765  001352  000020  70$:  MOV     TOCYL,RKDC(R5)  ;CYL#
7599
7600  034266  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7601  034274  013737  001414  003372        MOV     T10,TEMP1       ;SETUP TIMEOUT
```

```
 7602  034302  004737  043612            JSR     PC,FRDY         ;FIND RDY
 7603  034306  104131                    ERROR   131             ;NO RDY AFTER SEEK CMD.
 7604
 7605  034310  013737  001426  003372    MOV     T50000,TEMP1
 7606  034316  004737  044222            JSR     PC,FATT2        ;FIND ATTN
 7607  034322  104132                    ERROR   132             ;NO ATTN AFTER SEEK CMD
 7608  034324  032737  100000  003334    BIT     #CERR,HCS1
 7609  034332  001401                    BEQ     72$
 7610  034334  104210                    ERROR   210             ;CERR AFTER SEEK CMD.
 7611
 7612  034336  004737  045522    72$:    JSR     PC,SUBCLR
 7613  034342  104024                    ERROR   24              ;CERR AFTER SCLR
 7614
 7615  034344  023727  001352  000632    CMP     TOCYL,#410.     ;ALL CYL DONE?
 7616  034352  001403                    BEQ     71$             ;BR IF YES
 7617  034354  005237  001352            INC     TOCYL           ;ELSE DO ANOTHER
 7618  034360  000737                    BR      70$
 7619
 7620  034362  004737  045522    71$:    JSR     PC,SUBCLR
 7621  034366  104024                    ERROR   24              ;CERR AFTER SCLR
 7622
 7623  034370  005037  001176            CLR     $ESCAPE
 7624  034374  005737  001410            TST     LPFLG
 7625  034400  001402                    BEQ     73$
 7626  034402  000177  144502            JMP     @$LPERR         ;SW 9 WAS SET.
 7627  034406  000177  144474    73$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 7628
 7629
 7630
 7631  034412                    10$:
 7632  034412  005237  001410            INC     LPFLG
 7633  034416  032777  001000  144514    BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7634  034424  001312                    BNE     6$              ;YES, RECONDITION DRIVE
 7635  034426  000137  033560            JMP     2$              ;RETURN TO MAINLINE
 7636
 7637  034432                    12$:
 7638  034432  005237  001410            INC     LPFLG
 7639  034436  032777  001000  144474    BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7640  034444  001302                    BNE     6$              ;YES, RECONDITION DRIVE
 7641  034446  000137  034054            JMP     3$              ;RETURN TO MAINLINE
 7642
 7643  034452                    13$:
 7644
 7645
 7646                            ;;*****************************************************************
 7647                            ;*TEST 36       SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS
 7648                            ;*
 7649                            ;*      THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLS & READS HEADERS.
 7650                            ;*      IT THEN SEEKS CYL 0 & READS HEADERS.
 7651                            ;*
 7652                            ;*      MAJOR CYLS ARE: 1  (DECIMAL)  =  1 (OCTAL)
 7653                            ;*                      2                2
 7654                            ;*                      4                4
 7655                            ;*                      8                10
 7656                            ;*                      16               20
 7657                            ;*                      32               40
```

```
                                ;*                              64              100
                                ;*                              128             200
                                ;*                              256             400
                                ;*
                                ;;*****************************************************************
                        TST36:  SCOPE
034452  000004
034454  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
034462  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

034466  012737  000000  001350          MOV     #0,FRCYL        ;SETUP FROM CYL
034474  012737  000001  001352          MOV     #1,TOCYL        ;SETUP TO CYL

034502                          1$:
034502  104415                  SCOP1
034504  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

034510  004737  045522                  JSR     PC,SUBCLR
034514  104024                  ERROR   24                      ;CERR AFTER SCLR

034516  012737  036276  001176          MOV     #10$,$ESCAPE
034524  013737  001350  003376          MOV     FRCYL,TEMP3     ;SETUP
034532  013737  001352  003400          MOV     TOCYL,TEMP4     ;CYL DIFF
034540  163737  003376  003400          SUB     TEMP3,TEMP4     ;FOR
034546  013737  003400  001360          MOV     TEMP4,CALDIF    ;ERROR PRINTOUT

034554  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #

034562  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
034570  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
034576  004737  043612                  JSR     PC,FRDY         ;FIND RDY
034602  104131                  ERROR   131                     ;NO RDY AFTER SEEK CMD
034604  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
034612  005037  003426                  CLR     E.B0
034616  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
034624  012737  000001  003432          MOV     #1,E.B1

034632  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
034636  000003                  .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
034640  104203                  ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
034642  104204                  ERROR   204             ;MSG B0 ERROR
034644  104205                  ERROR   205             ;MSG A1 ERROR
034646  104206                  ERROR   206             ;MSG B1 ERROR

034650  012737  036316  001176  2$:     MOV     #12$,$ESCAPE
034656  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT

034664  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
034670  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
034672  032737  100000  003334          BIT     #CERR,HCS1
034700  001401                  BEQ     64$
034702  104210                  ERROR   210             ;CERR AFTER SEEK CMD
034704                          64$:

034704  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
034712  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
034716  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
```

```
7714  034724  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7715  034732  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7716  034736  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7717  034744  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7718
7719  034752  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7720  034756  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7721  034760  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7722  034762  104134                          ERROR   134             ;MSG B0 ERROR
7723  034764  104135                          ERROR   135             ;MSG A1 ERROR
7724  034766  104136                          ERROR   136             ;MSG B1 ERROR
7725  034770  005737  001362                  TST     CYLDIF
7726  034774  001401                          BEQ     65$
7727  034776  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7728
7729  035000                          65$:
7730
7731  035000  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7732  035006  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7733  035014  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7734  035022  013737  001414  003372          MOV     T10,TEMP1
7735  035030  004737  043612                  JSR     PC,FRDY         ;FIND RDY
7736  035034  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7737  035036  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
7738  035042  000401                          BR      66$
7739  035044  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7740  035046                          66$:
7741
7742  035046  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
7743  035054  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7744  035060  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
7745  035066  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7746  035074  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7747  035100  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7748  035106  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7749
7750  035114  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7751  035120  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7752  035122  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7753  035124  104265                          ERROR   265             ;MSG B0 ERROR
7754  035126  104274                          ERROR   274             ;MSG A1 ERROR
7755  035130  104266                          ERROR   266             ;MSG B1 ERROR
7756
7757  035132  023737  001364  001352          CMP     CYLADD,TOCYL
7758  035140  001401                          BEQ     3$
7759  035142  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
7760
7761  035144                          3$:
7762  035144  104415                          SCOP1
7763  035146  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7764
7765  035152  004737  045522                  JSR     PC,SUBCLR
7766  035156  104024                          ERROR   24              ;CERR AFTER SCLR
7767
7768  035160  005037  001176                  CLR     $ESCAPE
7769  035164  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
```

```
7770
7771
7772  035172  012700  001674              MOV     #RHTAB,R0
7773  035176  012765  000025  000000      MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7774  035204  013737  001426  003372      MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7775  035212  004737  043612              JSR     PC,FRDY         ;FIND RDY
7776  035216  104171                      ERROR   171             ;NO RDY AFTER READ HEADER CMD
7777  035220  032737  100000  003334      BIT     #CERR,HCS1
7778  035226  001405                      BEQ     68$
7779  035230  104174                      ERROR   174             ;CERR AFTER READ HEADER CMD
7780  035232  104401  056333              TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7781  035236  000137  043076              JMP     $EOP            ;ABORT DRIVE
7782
7783  035242  016520  000024      68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7784  035246  016520  000024              MOV     RKDB(R5),(R0)+  ;2'ND WORD
7785  035252  016520  000024              MOV     RKDB(R5),(R0)+  ;3'RD WORD
7786
7787
7788  035256  032765  100000  000010      BIT     #DLT,RKCS2(R5)
7789  035264  001407                      BEQ     69$
7790  035266  004737  045150              JSR     PC,GSTAT
7791  035272  104173                      ERROR   173             ;DLT AFTER READ HEADER CMD
7792  035274  104401  056333              TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7793  035300  000137  043076              JMP     $EOP            ;ABORT DRIVE
7794  035304                      69$:
7795
7796  035304  023737  001674  001352      CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7797  035312  001401                      BEQ     67$             ;BR IF SAME
7798  035314  104310                      ERROR   310             ;READ CYL WORD HEADER ERROR
7799  035316                      67$:
7800
7801
7802  035316  104415                      SCOP1
7803  035320  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
7804
7805  035324  004737  045522              JSR     PC,SUBCLR
7806  035330  104024                      ERROR   24              ;CERR AFTER SCLR
7807
7808  035332  012737  036336  001176      MOV     #14$,$ESCAPE
7809  035340  013765  001350  000020      MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
7810  035346  013737  001350  001354      MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROUTINE
7811
7812  035354  012765  000017  000000      MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7813  035362  013737  001414  003372      MOV     T10,TEMP1       ;SETUP  TIMEOUT
7814  035370  004737  043612              JSR     PC,FRDY         ;FIND RDY
7815  035374  104131                      ERROR   131             ;NO RDY AFTER SEEK CMD
7816  035376  012737  030140  003424      MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7817  035404  005037  003426              CLR     E.B0
7818  035410  012737  005720  003430      MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7819  035416  012737  000001  003432      MOV     #1,E.B1
7820
7821  035424  004737  044334              JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7822  035430  000003                      .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7823  035432  104203                      ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7824  035434  104204                      ERROR   204             ;MSG B0 ERROR
7825  035436  104205                      ERROR   205             ;MSG A1 ERROR
```

```
7826  035440  104206                          ERROR   206             ;MSG B1 ERROR
7827
7828
7829  035442  012737  036356  001176  4$:     MOV     #16$,$ESCAPE
7830  035450  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7831
7832  035456  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
7833  035462  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7834  035464  032737  100000  003334          BIT     #CERR,HCS1
7835  035472  001401                          BEQ     70$
7836  035474  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7837  035476                          70$:
7838
7839  035476  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7840  035504  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7841  035510  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7842  035516  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7843  035524  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7844  035530  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7845  035536  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7846
7847  035544  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7848  035550  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7849  035552  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7850  035554  104134                          ERROR   134             ;MSG B0 ERROR
7851  035556  104135                          ERROR   135             ;MSG A1 ERROR
7852  035560  104136                          ERROR   136             ;MSG B1 ERROR
7853  035562  005737  001362                  TST     CYLDIF
7854  035566  001401                          BEQ     71$
7855  035570  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7856
7857  035572                          71$:
7858
7859  035572  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7860  035600  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7861  035606  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7862  035614  013737  001414  003372          MOV     T10,TEMP1
7863  035622  004737  043612                  JSR     PC,FRDY         ;FIND RDY
7864  035626  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7865  035630  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
7866  035634  000401                          BR      72$
7867  035636  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7868  035640                          72$:
7869
7870  035640  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7871  035646  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7872  035652  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7873  035660  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7874  035666  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7875  035672  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7876  035700  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7877
7878  035706  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7879  035712  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7880  035714  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7881  035716  104265                          ERROR   265             ;MSG B0 ERROR
```

```
7882  035720  104274                          ERROR   274             ;MSG A1 ERROR
7883  035722  104266                          ERROR   266             ;MSG B1 ERROR
7884
7885  035724  023737  001364  001350          CMP     CYLADD,FRCYL
7886  035732  001401                          BEQ     5$
7887  035734  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7888
7889  035736                          5$:
7890  035736  104415                          SCOP1
7891  035740  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7892
7893  035744  004737  045522                  JSR     PC,SUBCLR
7894  035750  104024                          ERROR   24              ;CERR AFTER SCLR
7895
7896  035752  005037  001176                  CLR     $ESCAPE
7897  035756  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;CYL #
7898
7899
7900  035764  012700  001674                  MOV     #RHTAB,R0
7901  035770  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7902  035776  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7903  036004  004737  043612                  JSR     PC,FRDY         ;FIND RDY
7904  036010  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
7905  036012  032737  100000  003334          BIT     #CERR,HCS1
7906  036020  001405                          BEQ     74$
7907  036022  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
7908  036024  104401  056333                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7909  036030  000137  043076                  JMP     $EOP            ;ABORT DRIVE
7910
7911  036034  016520  000024          74$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7912  036040  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
7913  036044  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
7914
7915
7916  036050  032765  100000  000010          BIT     #DLT,RKCS2(R5)
7917  036056  001407                          BEQ     75$
7918  036060  004737  045150                  JSR     PC,GSTAT
7919  036064  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
7920  036066  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7921  036072  000137  043076                  JMP     $EOP            ;ABORT DRIVE
7922  036076                          75$:
7923
7924  036076  023737  001674  001350          CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
7925  036104  001401                          BEQ     73$             ;BR IF SAME
7926  036106  104311                          ERROR   311             ;READ CYL WORD HEADER ERROR
7927  036110                          73$:
7928
7929
7930  036110  023727  001352  000400          CMP     TOCYL,#400      ;ALL CYL DONE?
7931  036116  001404                          BEQ     6$              ;BR IF YES
7932  036120  006337  001352                  ASL     TOCYL           ;ELSE DO ANOTHER
7933  036124  000137  034502                  JMP     1$
7934  036130                          6$:
7935  036130  004737  047526                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7936  036134  000520                          BR      TST37           ;;GO TO NEXT TEST
7937                                                                  ;RETURN HERE IF SW 14 IS SET OR
```

```
                                                                ;SW 8 WITH SWR <7:0> APPLY
036136                                  8$:
                                        JSR     PC,SUBCLR
036136  004737  045522                ERROR   24                ;CERR AFTER SCRL
036142  104024
                                76$:    MOV     TOCYL,RKDC(R5)  ;CYL#
036144  013765  001352  000020
036152  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
036160  013737  001414  003372        MOV     T10,TEMP1       ;SETUP TIMEOUT
036166  004737  043612                JSR     PC,FRDY         ;FIND RDY
036172  104131                      ERROR   131             ;NO RDY AFTER SEEK CMD.
036174  013737  001426  003372        MOV     T50000,TEMP1
036202  004737  044222                JSR     PC,FATT2        ;FIND ATTN
036206  104132                      ERROR   132             ;NO ATTN AFTER SEEK CMD
036210  032737  100000  003334        BIT     #CERR,HCS1
036216  001401                      BEQ     78$
036220  104210                      ERROR   210             ;CERR AFTER SEEK CMD.
036222  004737  045522                78$:    JSR     PC,SUBCLR
036226  104024                      ERROR   24              ;CERR AFTER SCLR
036230  023727  001352  000000        CMP     TOCYL,#0        ;ALL CYL DONE?
036236  001403                      BEQ     77$             ;BR IF YES
036240  005337  001352                DEC     TOCYL           ;ELSE DO ANOTHER
036244  000737                      BR      76$
036246  004737  045522                77$:    JSR     PC,SUBCLR
036252  104024                      ERROR   24              ;CERR AFTER SCLR
036254  005037  001176        CLR     $ESCAPE
036260  005737  001410        TST     LPFLG
036264  001402                BEQ     79$
036266  000177  142616        JMP     @$LPERR         ;SW 9 WAS SET.
036272  000177  142610        79$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
036276                          10$:
036276  005237  001410        INC     LPFLG
036302  032777  001000  142630  BIT     #SW9,@SWR       ;LOOP ON ERROR?
036310  001312                BNE     8$              ;YES, RECONDITION DRIVE
036312  000137  034650        JMP     2$              ;RETURN TO MAINLINE
036316                          12$:
036316  005237  001410        INC     LPFLG
036322  032777  001000  142610  BIT     #SW9,@SWR       ;LOOP ON ERROR?
036330  001302                BNE     8$              ;YES, RECONDITION DRIVE
036332  000137  035144        JMP     3$              ;RETURN TO MAINLINE
036336                          14$:
036336  005237  001410        INC     LPFLG
036342  032777  001000  142570  BIT     #SW9,@SWR       ;LOOP ON ERROR?
036350  001272                BNE     8$              ;YES, RECONDITION DRIVE
036352  000137  035442        JMP     4$              ;RETURN TO MAINLINE
036356                          16$:
036356  005237  001410        INC     LPFLG
036362  032777  001000  142550  BIT     #SW9,@SWR       ;LOOP ON ERROR?
036370  001262                BNE     8$              ;YES, RECONDITION DRIVE
```

```
 7994  036372  000137  035736                    JMP     5$              ;RETURN TO MAINLINE
 7995
 7996
 7997
 7998                                            ;;*****************************************************************
 7999                                            ;*TEST 37       SEEK TO ALL CYLS FROM 0 & READ HEADERS
 8000                                            ;;*****************************************************************
 8001  036376  000004                            TST37:  SCOPE
 8002  036400  012737  000001  001174                    MOV     #1,$TIMES       ;;DO 1 ITERATION
 8003  036406  012706  001100                            MOV     #STACK,SP       ;RESTORE STK PTR
 8004
 8005  036412  005737  001342                            TST     MODTST          ;SEE IF MODULE TESTING
 8006  036416  001402                            BEQ     DOSEEK          ;BR IF NO
 8007  036420  000137  042352                    JMP     CYLINV          ;ELSE BYPASS TESTS 40 & 41
 8008  036424                                    DOSEEK:
 8009
 8010  036424  012737  000000  001350                    MOV     #0,FRCYL        ;SETUP FROM CYL
 8011  036432  012737  000001  001352                    MOV     #1,TOCYL        ;SETUP TO CYL
 8012
 8013  036440                                    1$:
 8014  036440  104415                                    SCOP1
 8015  036442  012706  001100                            MOV     #STACK,SP       ;RESTORE STK PTR
 8016
 8017  036446  004737  045522                            JSR     PC,SUBCLR
 8018  036452  104024                                    ERROR   24              ;CERR AFTER SCLR
 8019
 8020  036454  012737  040234  001176                    MOV     #10$,$ESCAPE
 8021  036462  013737  001350  003376                    MOV     FRCYL,TEMP3     ;SETUP
 8022  036470  013737  001352  003400                    MOV     TOCYL,TEMP4     ;CYL DIFF
 8023  036476  163737  003376  003400                    SUB     TEMP3,TEMP4     ;FOR
 8024  036504  013737  003400  001360                    MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
 8025
 8026  036512  013765  001352  000020                    MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
 8027
 8028  036520  012765  000017  000000                    MOV     #SEEK,RKCS1(R5) ;SEEK CMD
 8029  036526  013737  001414  003372                    MOV     T10,TEMP1       ;SETUP  TIMEOUT
 8030  036534  004737  043612                            JSR     PC,FRDY         ;FIND RDY
 8031  036540  104131                                    ERROR   131             ;NO RDY AFTER SEEK CMD
 8032  036542  012737  030140  003424                    MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 8033  036550  005037  003426                            CLR     E.B0
 8034  036554  012737  003720  003430                    MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 8035  036562  012737  000001  003432                    MOV     #1,E.B1
 8036
 8037  036570  004737  044334                            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8038  036574  000003                                    .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8039  036576  104203                                    ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 8040  036600  104204                                    ERROR   204             ;MSG B0 ERROR
 8041  036602  104205                                    ERROR   205             ;MSG A1 ERROR
 8042  036604  104206                                    ERROR   206             ;MSG B1 ERROR
 8043
 8044  036606  012737  040254  001176  2$:       MOV     #12$,$ESCAPE
 8045  036614  013737  001426  003372                    MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 8046
 8047  036622  004737  044222                            JSR     PC,FATT2        ;FIND ATTN
 8048  036626  104132                                    ERROR   132             ;NO ATTN AFTER SEEK CMD
 8049  036630  032737  100000  003334                    BIT     #CERR,HCS1
```

```
8050  036636  001401                        BEQ     64$
8051  036640  104210                        ERROR   210             ;CERR AFTER SEEK CMD
8052  036642                          64$:
8053
8054  036642  012737  050340  003424        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8055  036650  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
8056  036654  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8057  036662  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8058  036670  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
8059  036674  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8060  036702  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8061
8062  036710  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8063  036714  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8064  036716  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
8065  036720  104134                        ERROR   134             ;MSG B0 ERROR
8066  036722  104135                        ERROR   135             ;MSG A1 ERROR
8067  036724  104136                        ERROR   136             ;MSG B1 ERROR
8068  036726  005737  001362                TST     CYLDIF
8069  036732  001401                        BEQ     65$
8070  036734  104137                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8071
8072  036736                          65$:
8073
8074  036736  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
8075  036744  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
8076  036752  012765  000005  000000        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8077  036760  013737  001414  003372        MOV     T10,TEMP1
8078  036766  004737  043612                JSR     PC,FRDY         ;FIND RDY
8079  036772  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8080  036774  004737  044074                JSR     PC,TSTATN       ;TEST FOR ATTN
8081  037000  000401                        BR      66$
8082  037002  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8083  037004                          66$:
8084
8085  037004  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
8086  037012  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
8087  037016  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8088  037024  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8089  037032  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
8090  037036  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8091  037044  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8092
8093  037052  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8094  037056  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8095  037060  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8096  037062  104265                        ERROR   265             ;MSG B0 ERROR
8097  037064  104274                        ERROR   274             ;MSG A1 ERROR
8098  037066  104266                        ERROR   266             ;MSG B1 ERROR
8099
8100  037070  023737  001364  001352        CMP     CYLADD,TOCYL
8101  037076  001401                        BEQ     3$
8102  037100  104207                        ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
8103
8104  037102                          3$:
8105  037102  104415                        SCOP1
```

```
 8106  037104  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 8107
 8108  037110  004737  045522                  JSR     PC,SUBCLR
 8109  037114  104024                          ERROR   24              ;CERR AFTER SCLR
 8110
 8111  037116  005037  001176                  CLR     $ESCAPE
 8112  037122  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
 8113
 8114
 8115  037130  012700  001674                  MOV     #RHTAB,R0
 8116  037134  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
 8117  037142  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 8118  037150  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 8119  037154  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
 8120  037156  032737  100000  003334          BIT     #CERR,HCS1
 8121  037164  001405                          BEQ     68$
 8122  037166  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
 8123  037170  104401  056333                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 8124  037174  000137  043076                  JMP     $EOP            ;ABORT DRIVE
 8125
 8126  037200  016520  000024          68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 8127  037204  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
 8128  037210  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
 8129
 8130
 8131  037214  032765  100000  000010          BIT     #DLT,RKCS2(R5)
 8132  037222  001407                          BEQ     69$
 8133  037224  004737  045150                  JSR     PC,GSTAT
 8134  037230  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
 8135  037232  104401  056333                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 8136  037236  000137  043076                  JMP     $EOP            ;ABORT DRIVE
 8137  037242                          69$:
 8138
 8139  037242  023737  001674  001352          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
 8140  037250  001401                          BEQ     67$             ;BR IF SAME
 8141  037252  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
 8142  037254                          67$:
 8143
 8144
 8145  037254  104415                          SCOP1
 8146  037256  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 8147
 8148  037262  004737  045522                  JSR     PC,SUBCLR
 8149  037266  104024                          ERROR   24              ;CERR AFTER SCLR
 8150
 8151  037270  012737  040274  001176          MOV     #14$,$ESCAPE
 8152  037276  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
 8153  037304  013737  001350  001354          MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROUTINE
 8154
 8155  037312  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
 8156  037320  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
 8157  037326  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 8158  037332  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 8159  037334  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 8160  037342  005037  003426                  CLR     E.B0
 8161  037346  012737  005720  003430          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
```

```
 8162  037354  012737  000001  003432          MOV     #1,E.B1
 8163
 8164  037362  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8165  037366  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8166  037370  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 8167  037372  104204                          ERROR   204             ;MSG B0 ERROR
 8168  037374  104205                          ERROR   205             ;MSG A1 ERROR
 8169  037376  104206                          ERROR   206             ;MSG B1 ERROR
 8170
 8171
 8172  037400  012737  040314  001176  4$:     MOV     #16$,$ESCAPE
 8173  037406  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 8174
 8175  037414  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
 8176  037420  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 8177  037422  032737  100000  003334          BIT     #CERR,HCS1
 8178  037430  001401                          BEQ     70$
 8179  037432  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 8180  037434                          70$:
 8181
 8182  037434  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 8183  037442  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
 8184  037446  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 8185  037454  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 8186  037462  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
 8187  037466  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 8188  037474  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 8189
 8190  037502  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8191  037506  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8192  037510  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 8193  037512  104134                          ERROR   134             ;MSG B0 ERROR
 8194  037514  104135                          ERROR   135             ;MSG A1 ERROR
 8195  037516  104136                          ERROR   136             ;MSG B1 ERROR
 8196  037520  005737  001362                  TST     CYLDIF
 8197  037524  001401                          BEQ     71$
 8198  037526  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 8199
 8200  037530                          71$:
 8201
 8202  037530  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8203  037536  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 8204  037544  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 8205  037552  013737  001414  003372          MOV     T10,TEMP1
 8206  037560  004737  043612                  JSR     PC,FRDY         ;FIND RDY
 8207  037564  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 8208  037566  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
 8209  037572  000401                          BR      72$
 8210  037574  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 8211  037576                          72$:
 8212
 8213  037576  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 8214  037604  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
 8215  037610  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 8216  037616  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 8217  037624  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
```

```
8218  037630  012737  000002  003436          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
8219  037636  012737  000003  003442          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
8220
8221  037644  004737  044334                  JSR     PC,CHKMSG        ;CHECK MSGS A0,B0,A1,B1
8222  037650  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
8223  037652  104273                          ERROR   273              ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8224  037654  104265                          ERROR   265              ;MSG B0 ERROR
8225  037656  104274                          ERROR   274              ;MSG A1 ERROR
8226  037660  104266                          ERROR   266              ;MSG B1 ERROR
8227
8228  037662  023737  001364  001350          CMP     CYLADD,FRCYL
8229  037670  001401                          BEQ     5$
8230  037672  104243                          ERROR   243              ;CYL ADDR IN RKMR3 NOT=RKDC
8231
8232  037674                          5$:
8233  037674  104415                          SCOP1
8234  037676  012706  001100                  MOV     #STACK,SP        ;RESTORE STK PTR
8235
8236  037702  004737  045522                  JSR     PC,SUBCLR
8237  037706  104024                          ERROR   24               ;CERR AFTER SCLR
8238
8239  037710  005037  001176                  CLR     $ESCAPE
8240  037714  013765  001350  000020          MOV     FRCYL,RKDC(R5)   ;CYL #
8241
8242
8243  037722  012700  001674                  MOV     #RHTAB,R0
8244  037726  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)      ;READ HEADER CMD
8245  037734  013737  001426  003372          MOV     T50000,TEMP1     ;SETUP  TIMEOUT
8246  037742  004737  043612                  JSR     PC,FRDY          ;FIND RDY
8247  037746  104171                          ERROR   171              ;NO RDY AFTER READ HEADER CMD
8248  037750  032737  100000  003334          BIT     #CERR,HCS1
8249  037756  001405                          BEQ     74$
8250  037760  104174                          ERROR   174              ;CERR AFTER READ HEADER CMD
8251  037762  104401  056333                  TYPE    ,MSG18           ;ABORT BALANCE OF TESTS
8252  037766  000137  043076                  JMP     $EOP             ;ABORT DRIVE
8253
8254  037772  016520  000024          74$:    MOV     RKDB(R5),(R0)+   ;1'ST WORD FROM SILO TO RHTAB
8255  037776  016520  000024                  MOV     RKDB(R5),(R0)+   ;2'ND WORD
8256  040002  016520  000024                  MOV     RKDB(R5),(R0)+   ;3'RD WORD
8257
8258
8259  040006  032765  100000  000010          BIT     #DLT,RKCS2(R5)
8260  040014  001407                          BEQ     75$
8261  040016  004737  045150                  JSR     PC,GSTAT
8262  040022  104173                          ERROR   173              ;DLT AFTER READ HEADER CMD
8263  040024  104401  056333                  TYPE    ,MSG18           ;ABORTING BALANCE OF TESTS
8264  040030  000137  043076                  JMP     $EOP             ;ABORT DRIVE
8265  040034                          75$:
8266
8267  040034  023737  001674  001350          CMP     RHTAB,FRCYL      ;CHECK WORD 0 (CYL#) ONLY
8268  040042  001401                          BEQ     73$              ;BR IF SAME
8269  040044  104311                          ERROR   311              ;READ CYL WORD HEADER ERROR
8270  040046                          73$:
8271
8272
8273  040046  023727  001352  000632          CMP     TOCYL,#410.      ;ALL CYL DONE?
```

```
8274  040054  001404                        BEQ     6$                ;BR IF YES
8275  040056  005237  001352                INC     TOCYL             ;ELSE DO ANOTHER
8276  040062  000137  036440                JMP     1$
8277  040066                          6$:
8278  040066  004737  047526                JSR     PC,SWTST          ;SEE IF SW 14 OR 8 IS SET
8279  040072  000520                        BR      TST40             ;;GO TO NEXT TEST
8280                                                                  ;RETURN HERE IF SW 14 IS SET OR
8281                                                                  ;SW 8 WITH SWR <7:0> APPLY
8282  040074                          8$:
8283
8284  040074  004737  045522                JSR     PC,SUBCLR
8285  040100  104024                        ERROR   24                ;CERR AFTER SCRL
8286
8287  040102  013765  001352  000020  76$:  MOV     TOCYL,RKDC(R5)    ;CYL#
8288
8289  040110  012765  000017  000000        MOV     #SEEK,RKCS1(R5)   ;SEEK CMD TO RECONDITION DRIVE.
8290  040116  013737  001414  003372        MOV     T10,TEMP1         ;SETUP TIMEOUT
8291  040124  004737  043612                JSR     PC,FRDY           ;FIND RDY
8292  040130  104131                        ERROR   131               ;NO RDY AFTER SEEK CMD.
8293
8294  040132  013737  001426  003372        MOV     T50000,TEMP1
8295  040140  004737  044222                JSR     PC,FATT2          ;FIND ATTN
8296  040144  104132                        ERROR   132               ;NO ATTN AFTER SEEK CMD
8297  040146  032737  100000  003334        BIT     #CERR,HCS1
8298  040154  001401                        BEQ     78$
8299  040156  104210                        ERROR   210               ;CERR AFTER SEEK CMD.
8300
8301  040160  004737  045522          78$:  JSR     PC,SUBCLR
8302  040164  104024                        ERROR   24                ;CERR AFTER SCLR
8303
8304  040166  023727  001352  000000        CMP     TOCYL,#0          ;ALL CYL DONE?
8305  040174  001403                        BEQ     77$               ;BR IF YES
8306  040176  005337  001352                DEC     TOCYL             ;ELSE DO ANOTHER
8307  040202  000737                        BR      76$
8308
8309  040204  004737  045522          77$:  JSR     PC,SUBCLR
8310  040210  104024                        ERROR   24                ;CERR AFTER SCLR
8311
8312  040212  005037  001176                CLR     $ESCAPE
8313  040216  005737  001410                TST     LPFLG
8314  040222  001402                        BEQ     79$
8315  040224  000177  140660                JMP     @$LPERR           ;SW 9 WAS SET.
8316  040230  000177  140652          79$:  JMP     @$LPADR           ;SW 14 OR 8 WAS SET
8317
8318  040234                          10$:
8319  040234  005237  001410                INC     LPFLG
8320  040240  032777  001000  140672        BIT     #SW9,@SWR         ;LOOP ON ERROR?
8321  040246  001312                        BNE     8$                ;YES, RECONDITION DRIVE
8322  040250  000137  036606                JMP     2$                ;RETURN TO MAINLINE
8323  040254                          12$:
8324  040254  005237  001410                INC     LPFLG
8325  040260  032777  001000  140652        BIT     #SW9,@SWR         ;LOOP ON ERROR?
8326  040266  001302                        BNE     8$                ;YES, RECONDITION DRIVE
8327  040270  000137  037102                JMP     3$                ;RETURN TO MAINLINE
8328  040274                          14$:
8329  040274  005237  001410                INC     LPFLG
```

```
8330  040300  032777  001000  140632          BIT     #SW9,@SWR       ;LOOP ON ERROR?
8331  040306  001272                          BNE     8$              ;YES, RECONDITION DRIVE
8332  040310  000137  037400                  JMP     4$              ;RETURN TO MAINLINE
8333  040314                          16$:
8334  040314  005237  001410                  INC     LPFLG
8335  040320  032777  001000  140612          BIT     #SW9,@SWR       ;LOOP ON ERROR?
8336  040326  001262                          BNE     8$              ;YES, RECONDITION DRIVE
8337  040330  000137  037674                  JMP     5$              ;RETURN TO MAINLINE
8338
8339                                  ;;*****************************************************************
8340                                  ;*TEST 40       SEEK TO ALL CYLS FROM CYL 410 & READ HEADERS
8341                                  ;;****************************************************************
8342  040334  000004                  TST40:  SCOPE
8343  040336  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
8344  040344  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
8345
8346
8347  040350  004737  045522                  JSR     PC,SUBCLR
8348  040354  104024                          ERROR   24              ;CERR AFTER SCLR
8349
8350  040356  012765  000632  000020          MOV     #410.,RKDC(R5)  ;QUICK SEEK TO CYL 410
8351
8352  040364  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
8353  040372  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
8354  040400  004737  043612                  JSR     PC,FRDY         ;FIND RDY
8355  040404  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
8356
8357  040406  013737  001426  003372          MOV     T50000,TEMP1
8358  040414  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
8359  040420  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
8360  040422  032737  100000  003334          BIT     #CERR,HCS1
8361  040430  001401                          BEQ     64$
8362  040432  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
8363
8364  040434  004737  045522          64$:    JSR     PC,SUBCLR
8365  040440  104024                          ERROR   24              ;CERR AFTER SCLR
8366
8367
8368  040442  012737  000632  001350          MOV     #410.,FRCYL     ;SETUP FROM CYL
8369  040450  012737  000631  001352          MOV     #409.,TOCYL     ;SETUP TO CYL
8370
8371  040456                          1$:
8372  040456  104415                          SCOP1
8373  040460  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
8374
8375  040464  004737  045522                  JSR     PC,SUBCLR
8376  040470  104024                          ERROR   24              ;CERR AFTER SCLR
8377
8378  040472  012737  042252  001176          MOV     #10$,$ESCAPE
8379  040500  013737  001350  003376          MOV     FRCYL,TEMP3     ;SETUP
8380  040506  013737  001352  003400          MOV     TOCYL,TEMP4     ;CYL DIFF
8381  040514  163737  003400  003376          SUB     TEMP4,TEMP3     ;FOR
8382  040522  013737  003376  001360          MOV     TEMP3,CALDIF    ;ERROR PRINTOUT
8383
8384  040530  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
8385
```

```
8386  040536  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
8387  040544  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
8388  040552  004737  043612                  JSR     PC,FRDY         ;FIND RDY
8389  040556  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
8390  040560  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
8391  040566  005037  003426                  CLR     E.B0
8392  040572  012737  005720  003430          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8393  040600  012737  000001  003432          MOV     #1,E.B1
8394
8395  040606  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8396  040612  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8397  040614  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
8398  040616  104204                          ERROR   204             ;MSG B0 ERROR
8399  040620  104205                          ERROR   205             ;MSG A1 ERROR
8400  040622  104206                          ERROR   206             ;MSG B1 ERROR
8401
8402  040624  012737  042272  001176  2$:     MOV     #12$,$ESCAPE
8403  040632  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
8404
8405  040640  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
8406  040644  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
8407  040646  032737  100000  003334          BIT     #CERR,HCS1
8408  040654  001401                          BEQ     65$
8409  040656  104210                          ERROR   210             ;CERR AFTER SEEK CMD
8410  040660                          65$:
8411
8412  040660  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8413  040666  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
8414  040672  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8415  040700  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8416  040706  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
8417  040712  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8418  040720  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8419
8420  040726  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8421  040732  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8422  040734  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
8423  040736  104134                          ERROR   134             ;MSG B0 ERROR
8424  040740  104135                          ERROR   135             ;MSG A1 ERROR
8425  040742  104136                          ERROR   136             ;MSG B1 ERROR
8426  040744  005737  001362                  TST     CYLDIF
8427  040750  001401                          BEQ     66$
8428  040752  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8429
8430  040754                          66$:
8431
8432  040754  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
8433  040762  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
8434  040770  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8435  040776  013737  001414  003372          MOV     T10,TEMP1
8436  041004  004737  043612                  JSR     PC,FRDY         ;FIND RDY
8437  041010  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8438  041012  004737  044074                  JSR     PC,TSTATN       ;TEST FOR ATTN
8439  041016  000401                          BR      67$
8440  041020  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8441  041022                          67$:
```

```
8442
8443  041022  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
8444  041030  005037  003426                  CLR     E.B0                    ;EXPECTED MSG B0
8445  041034  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8446  041042  012737  000001  003432          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
8447  041050  005037  003434                  CLR     E.A2                    ;EXPECTED MSG A2
8448  041054  012737  000002  003436          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
8449  041062  012737  000003  003442          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
8450
8451  041070  004737  044334                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
8452  041074  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
8453  041076  104273                          ERROR   273                     ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8454  041100  104265                          ERROR   265                     ;MSG B0 ERROR
8455  041102  104274                          ERROR   274                     ;MSG A1 ERROR
8456  041104  104266                          ERROR   266                     ;MSG B1 ERROR
8457
8458  041106  023737  001364  001352          CMP     CYLADD,TOCYL
8459  041114  001401                          BEQ     3$
8460  041116  104207                          ERROR   207                     ;CYL ADDR IN RKMR3 NOT=RKDC
8461
8462  041120                          3$:
8463  041120  104415                          SCOP1
8464  041122  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR
8465
8466  041126  004737  045522                  JSR     PC,SUBCLR
8467  041132  104024                          ERROR   24                      ;CERR AFTER SCLR
8468
8469  041134  005037  001176                  CLR     $ESCAPE
8470  041140  013765  001352  000020          MOV     TOCYL,RKDC(R5)          ;CYL #
8471
8472
8473  041146  012700  001674                  MOV     #RHTAB,R0
8474  041152  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
8475  041160  013737  001426  003372          MOV     T50000,TEMP1            ;SETUP  TIMEOUT
8476  041166  004737  043612                  JSR     PC,FRDY                 ;FIND RDY
8477  041172  104171                          ERROR   171                     ;NO RDY AFTER READ HEADER CMD
8478  041174  032737  100000  003334          BIT     #CERR,HCS1
8479  041202  001405                          BEQ     69$
8480  041204  104174                          ERROR   174                     ;CERR AFTER READ HEADER CMD
8481  041206  104401  056333                  TYPE    ,MSG18                  ;ABORT BALANCE OF TESTS
8482  041212  000137  043076                  JMP     $EOP                    ;ABORT DRIVE
8483
8484  041216  016520  000024          69$:    MOV     RKDB(R5),(R0)+          ;1'ST WORD FROM SILO TO RHTAB
8485  041222  016520  000024                  MOV     RKDB(R5),(R0)+          ;2'ND WORD
8486  041226  016520  000024                  MOV     RKDB(R5),(R0)+          ;3'RD WORD
8487
8488
8489  041232  032765  100000  000010          BIT     #DLT,RKCS2(R5)
8490  041240  001407                          BEQ     70$
8491  041242  004737  045150                  JSR     PC,GSTAT
8492  041246  104173                          ERROR   173                     ;DLT AFTER READ HEADER CMD
8493  041250  104401  056333                  TYPE    ,MSG18                  ;ABORTING BALANCE OF TESTS
8494  041254  000137  043076                  JMP     $EOP                    ;ABORT DRIVE
8495  041260                          70$:
8496
8497  041260  023737  001674  001352          CMP     RHTAB,TOCYL             ;CHECK WORD 0 (CYL#) ONLY
```

```
8498  041266  001401                        BEQ     68$             ;BR IF SAME
8499  041270  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
8500  041272                          68$:
8501
8502
8503  041272  104415                        SCOP1
8504  041274  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
8505
8506  041300  004737  045522                JSR     PC,SUBCLR
8507  041304  104024                        ERROR   24              ;CERR AFTER SCLR
8508
8509  041306  012737  042312  001176        MOV     #14$,$ESCAPE
8510  041314  013765  001350  000020        MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
8511  041322  013737  001350  001354        MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROUTINE
8512
8513  041330  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
8514  041336  013737  001414  003372        MOV     T10,TEMP1       ;SETUP  TIMEOUT
8515  041344  004737  043612                JSR     PC,FRDY         ;FIND RDY
8516  041350  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
8517  041352  012737  030140  003424        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
8518  041360  005037  003426                CLR     E.B0
8519  041364  012737  003720  003430        MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8520  041372  012737  000001  003432        MOV     #1,E.B1
8521
8522  041400  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8523  041404  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8524  041406  104203                        ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
8525  041410  104204                        ERROR   204             ;MSG B0 ERROR
8526  041412  104205                        ERROR   205             ;MSG A1 ERROR
8527  041414  104206                        ERROR   206             ;MSG B1 ERROR
8528
8529
8530  041416  012737  042332  001176  4$:   MOV     #16$,$ESCAPE
8531  041424  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
8532
8533  041432  004737  044222                JSR     PC,FATT2        ;FIND ATTN
8534  041436  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
8535  041440  032737  100000  003334        BIT     #CERR,HCS1
8536  041446  001401                        BEQ     71$
8537  041450  104210                        ERROR   210             ;CERR AFTER SEEK CMD
8538  041452                          71$:
8539
8540  041452  012737  050340  003424        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8541  041460  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
8542  041464  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8543  041472  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8544  041500  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
8545  041504  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8546  041512  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8547
8548  041520  004737  044334                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8549  041524  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8550  041526  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
8551  041530  104134                        ERROR   134             ;MSG B0 ERROR
8552  041532  104135                        ERROR   135             ;MSG A1 ERROR
8553  041534  104136                        ERROR   136             ;MSG B1 ERROR
```

```
8554  041536  005737  001362          TST     CYLDIF
8555  041542  001401                  BEQ     72$
8556  041544  104137                  ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8557
8558  041546                  72$:
8559
8560  041546  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
8561  041554  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
8562  041562  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8563  041570  013737  001414  003372  MOV     T10,TEMP1
8564  041576  004737  043612          JSR     PC,FRDY         ;FIND RDY
8565  041602  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8566  041604  004737  044074          JSR     PC,TSTATN       ;TEST FOR ATTN
8567  041610  000401                  BR      73$
8568  041612  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8569  041614                  73$:
8570
8571  041614  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
8572  041622  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
8573  041626  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
8574  041634  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8575  041642  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
8576  041646  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8577  041654  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8578
8579  041662  004737  044334          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8580  041666  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8581  041670  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8582  041672  104265                  ERROR   265             ;MSG B0 ERROR
8583  041674  104274                  ERROR   274             ;MSG A1 ERROR
8584  041676  104266                  ERROR   266             ;MSG B1 ERROR
8585
8586  041700  023737  001364  001350  CMP     CYLADD,FRCYL
8587  041706  001401                  BEQ     5$
8588  041710  104243                  ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
8589
8590  041712                  5$:
8591  041712  104415                  SCOP1
8592  041714  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
8593
8594  041720  004737  045522          JSR     PC,SUBCLR
8595  041724  104024                  ERROR   24              ;CERR AFTER SCLR
8596
8597  041726  005037  001176          CLR     $ESCAPE
8598  041732  013765  001350  000020  MOV     FRCYL,RKDC(R5)  ;CYL #
8599
8600
8601  041740  012700  001674          MOV     #RHTAB,R0
8602  041744  012765  000025  000000  MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
8603  041752  013737  001426  003372  MOV     T50000,TEMP1    ;SETUP  TIMEOUT
8604  041760  004737  043612          JSR     PC,FRDY         ;FIND RDY
8605  041764  104171                  ERROR   171             ;NO RDY AFTER READ HEADER CMD
8606  041766  032737  100000  003334  BIT     #CERR,HCS1
8607  041774  001405                  BEQ     75$
8608  041776  104174                  ERROR   174             ;CERR AFTER READ HEADER CMD
8609  042000  104401  056333          TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
```

```
8610  042004  000137  043076                JMP     $EOP                ;ABORT DRIVE
8611
8612  042010  016520  000024        75$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
8613  042014  016520  000024                MOV     RKDB(R5),(R0)+      ;2'ND WORD
8614  042020  016520  000024                MOV     RKDB(R5),(R0)+      ;3'RD WORD
8615
8616
8617  042024  032765  100000  000010        BIT     #DLT,RKCS2(R5)
8618  042032  001407                        BEQ     76$
8619  042034  004737  045150                JSR     PC,GSTAT
8620  042040  104173                        ERROR   173                 ;DLT AFTER READ HEADER CMD
8621  042042  104401  056333                TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
8622  042046  000137  043076                JMP     $EOP                ;ABORT DRIVE
8623  042052                        76$:
8624
8625  042052  023737  001674  001350        CMP     RHTAB,FRCYL         ;CHECK WORD 0 (CYL#) ONLY
8626  042060  001401                        BEQ     74$                 ;BR IF SAME
8627  042062  104311                        ERROR   311                 ;READ CYL WORD HEADER ERROR
8628  042064                        74$:
8629
8630
8631  042064  023727  001352  000000        CMP     TOCYL,#0            ;ALL CYL DONE?
8632  042072  001404                        BEQ     6$                  ;BR IF YES
8633  042074  005337  001352                DEC     TOCYL               ;ELSE DO ANOTHER
8634  042100  000137  040456                JMP     1$
8635  042104                        6$:
8636  042104  004737  047526                JSR     PC,SWTST            ;SEE IF SW 14 OR 8 IS SET
8637  042110  000520                        BR      TST41               ;;GO TO NEXT TEST
8638                                                                    ;RETURN HERE IF SW 14 IS SET OR
8639                                                                    ;SW 8 WITH SWR <7:0> APPLY
8640  042112                        8$:
8641
8642  042112  004737  045522                JSR     PC,SUBCLR
8643  042116  104024                        ERROR   24                  ;CERR AFTER SCRL
8644
8645  042120  013765  001352  000020  77$:  MOV     TOCYL,RKDC(R5)      ;CYL#
8646
8647  042126  012765  000017  000000        MOV     #SEEK,RKCS1(R5)     ;SEEK CMD TO RECONDITION DRIVE.
8648  042134  013737  001414  003372        MOV     T10,TEMP1           ;SETUP TIMEOUT
8649  042142  004737  043612                JSR     PC,FRDY             ;FIND RDY
8650  042146  104131                        ERROR   131                 ;NO RDY AFTER SEEK CMD.
8651
8652  042150  013737  001426  003372        MOV     T50000,TEMP1
8653  042156  004737  044222                JSR     PC,FATT2            ;FIND ATTN
8654  042162  104132                        ERROR   132                 ;NO ATTN AFTER SEEK CMD
8655  042164  032737  100000  003334        BIT     #CERR,HCS1
8656  042172  001401                        BEQ     79$
8657  042174  104210                        ERROR   210                 ;CERR AFTER SEEK CMD.
8658
8659  042176  004737  045522        79$:    JSR     PC,SUBCLR
8660  042202  104024                        ERROR   24                  ;CERR AFTER SCLR
8661
8662  042204  023727  001352  000632        CMP     TOCYL,#410.         ;ALL CYL DONE?
8663  042212  001403                        BEQ     78$                 ;BR IF YES
8664  042214  005237  001352                INC     TOCYL               ;ELSE DO ANOTHER
8665  042220  000737                        BR      77$
```

```
8666
8667  042222  004737  045522          78$:    JSR     PC,SUBCLR
8668  042226  104024                  ERROR   24              ;CERR AFTER SCLR
8669
8670  042230  005037  001176          CLR     $ESCAPE
8671  042234  005737  001410          TST     LPFLG
8672  042240  001402                  BEQ     80$
8673  042242  000177  136642          JMP     @$LPERR         ;SW 9 WAS SET.
8674  042246  000177  136634  80$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
8675
8676  042252                  10$:
8677  042252  005237  001410          INC     LPFLG
8678  042256  032777  001000  136654  BIT     #SW9,@SWR       ;LOOP ON ERROR?
8679  042264  001312                  BNE     8$              ;YES, RECONDITION DRIVE
8680  042266  000137  040624          JMP     2$              ;RETURN TO MAINLINE
8681  042272                  12$:
8682  042272  005237  001410          INC     LPFLG
8683  042276  032777  001000  136634  BIT     #SW9,@SWR       ;LOOP ON ERROR?
8684  042304  001302                  BNE     8$              ;YES, RECONDITION DRIVE
8685  042306  000137  041120          JMP     3$              ;RETURN TO MAINLINE
8686  042312                  14$:
8687  042312  005237  001410          INC     LPFLG
8688  042316  032777  001000  136614  BIT     #SW9,@SWR       ;LOOP ON ERROR?
8689  042324  001272                  BNE     8$              ;YES, RECONDITION DRIVE
8690  042326  000137  041416          JMP     4$              ;RETURN TO MAINLINE
8691  042332                  16$:
8692  042332  005237  001410          INC     LPFLG
8693  042336  032777  001000  136574  BIT     #SW9,@SWR       ;LOOP ON ERROR?
8694  042344  001262                  BNE     8$              ;YES, RECONDITION DRIVE
8695  042346  000137  041712          JMP     5$              ;RETURN TO MAINLINE
8696
8697  042352                  CYLINV:
8698
8699
8700                                  ;;*****************************************************************
8701                                  ;*TEST 41       SEEK TO ALL KEY INVALID CYLS
8702                                  ;*
8703                                  ;*      THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
8704                                  ;*      PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
8705                                  ;*      CYL IS SPECIFIED IN A SEEK.
8706                                  ;*
8707                                  ;*      INVALID CYLS ARE 411 THRU 511 (10)  IE. 633 THRU 777 (8)
8708                                  ;*
8709                                  ;*      THIS TEST CHECKS KEY INVALID CYLS 411,412,416,448 & 480
8710                                  ;*      FOR A FULL LOGIC TEST
8711                                  ;*
8712                                  ;;*****************************************************************
8713  042352  000004          TST41:  SCOPE
8714  042354  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
8715  042362  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
8716
8717  042366  004737  045522          JSR     PC,SUBCLR
8718  042372  104024                  ERROR   24              ;CERR AFTER SCLR
8719
8720  042374  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
8721  042402  013737  001414  003372  MOV     T10,TEMP1       ;SETUP TIMEOUT
```

```
8722  042410  004737  043612                  JSR     PC,FRDY         ;FIND RDY
8723  042414  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
8724
8725  042416  013737  001426  003372          MOV     T50000,TEMP1
8726  042424  004737  044222                  JSR     PC,FATT2        ;FIND ATTN
8727  042430  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
8728  042432  032737  100000  003334          BIT     #CERR,HCS1
8729  042440  001401                          BEQ     64$
8730  042442  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
8731
8732  042444  004737  045522          64$:    JSR     PC,SUBCLR
8733  042450  104024                          ERROR   24              ;CERR AFTER SCLR
8734
8735  042452  005000                          CLR     R0
8736  042454  005037  001350                  CLR     FRCYL           ;FROM CYL 0
8737
8738  042460                          1$:
8739  042460  104415                          SCOP1
8740  042462  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
8741
8742  042466  004737  045522                  JSR     PC,SUBCLR
8743  042472  104024                          ERROR   24              ;CERR AFTER SCLR
8744
8745  042474  016037  003304  001352          MOV     INVCYL(R0),TOCYL        ;GET INVALID CYL ADDR
8746  042502  013737  001352  001360          MOV     TOCYL,CALDIF
8747  042510  013765  001352  000020          MOV     TOCYL,RKDC(R5)
8748  042516  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
8749  042524  012737  000005  003372          MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
8750  042532  004737  043612                  JSR     PC,FRDY         ;FIND RDY
8751  042536  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
8752  042540  004737  044074                  JSR     PC,TSTATN
8753  042544  104245                          ERROR   245             ;NO ATTN AFTER SEEK TO INV CYL
8754
8755  042546  032737  000040  003364          BIT     #D.IDAE,HMR3
8756  042554  001001                          BNE     2$
8757  042556  104246                          ERROR   246             ;IDAE NOT SET AFTER SEEK TO INVALID ADDR
8758  042560  032737  000200  003364  2$:     BIT     #D.FLT,HMR3
8759  042566  001001                          BNE     4$
8760  042570  104247                          ERROR   247             ;FLT NOT SET AFTER SEEK TO INV ADDR
8761  042572  032737  020000  003362  4$:     BIT     #D.PIP,HMR2
8762  042600  001401                          BEQ     5$
8763  042602  104250                          ERROR   250             ;PIP SET AFTER SEEK TO INV ADDR
8764  042604  032737  040000  003362  5$:     BIT     #D.DSC,HMR2
8765  042612  001001                          BNE     6$
8766  042614  104251                          ERROR   251             ;DSC NOT SET AFTER SEEK TO INV ADDR
8767
8768  042616  005237  001462          6$:     INC     BYPCERR         ;BYPASS CHECKING FOR CERR IN GSTAT1
8769  042622  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
8770  042630  012737  002240  003426          MOV     #<D.SKI!D.FLT!D.IDAE>,E.B0
8771  042636  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8772  042644  012737  000001  003432          MOV     #1,E.B1
8773
8774  042652  004737  044334                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8775  042656  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8776  042660  104252                          ERROR   252             ;MSG A0 ERROR AFTER SEEK TO INV CYL
8777  042662  104253                          ERROR   253             ;MSG B0 ERROR
```

```
                                                ERROR   254             ;MSG A1 ERROR
                                                ERROR   255             ;MSG B1 ERROR
8780  042670  023737  001362  001352           CMP     CYLDIF,TOCYL
8781  042676  001401                           BEQ     7$
                                                ERROR   256             ;CYL DIFF IN RKMR2 NOT=CYL DIF
8783  042702  023737  001364  001352  7$:      CMP     CYLADD,TOCYL
8784  042710  001401                           BEQ     8$
                                                ERROR   257             ;CYL ADDR IN RKMR3 NOT=RKDC
8786
8787  042714                                8$:
8788
8789  042714  012765  100000  000000           MOV     #CCLR,RKCS1(R5)
8790  042722  013765  001222  000010           MOV     $UNIT,RKCS2(R5)
8791  042730  012765  000013  000000           MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
8792  042736  013737  001414  003372           MOV     T10,TEMP1
8793  042744  004737  043612                   JSR     PC,FRDY         ;FIND RDY
8794  042750  104124                           ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
8795
8796  042752  012765  100000  000000           MOV     #CCLR,RKCS1(R5)
8797  042760  013765  001222  000010           MOV     $UNIT,RKCS2(R5) ;DRIVE#
8798  042766  012765  000005  000000           MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8799  042774  013737  001414  003372           MOV     T10,TEMP1
8800  043002  004737  043612                   JSR     PC,FRDY         ;FIND RDY
8801  043006  104151                           ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8802  043010  004737  044074                   JSR     PC,TSTATN       ;TEST FOR ATTN
8803  043014  000401                           BR      66$
8804  043016  104154                           ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8805  043020                                66$:
8806
8807
8808  043020  004737  045150                   JSR     PC,GSTAT
8809  043024  032737  000040  003364           BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
8810  043032  001401                           BEQ     65$             ;BR IF YES
8811  043034  104155                           ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
8812
8813  043036  012765  100000  000000  65$:     MOV     #CCLR,RKCS1(R5)
8814  043044  013737  001412  003374           MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
8815  043052  004737  044126                   JSR     PC,FATT1
8816  043056  104055                           ERROR   55              ;NO ATTN AFTER RECAL CMD
8817
8818
8819  043060  062700  000002                   ADD     #2,R0
8820  043064  020027  000012                   CMP     R0,#10.
8821  043070  001402                           BEQ     $EOP
8822  043072  000137  042460                   JMP     1$
8823
8824
```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO ST5

043076                          $EOP:

                                        SCOPE
043076  000004
043100  012737  000001  001174          MOV     #1,$TIMES
043106  012706  001100                  MOV     #STACK,SP
043112  005237  001220                  INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
043116  023737  003454  001220          CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
043124  001403                          BEQ     $EOP1+2         ;BR IF YES
043126  000137  011362                  JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
043132  000004                  $EOP1:  SCOPE
043134  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
043140  005037  001174                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
043144  005237  001216                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
043150  042737  100000  001216          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
043156  005327                          DEC     (PC)+           ;;LOOP?
043160  000001                  $EOPCT: .WORD   1
043162  003022                          BGT     $DOAGN          ;;YES
043164  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
043166  000001                  $ENDCT: .WORD   1
043170  043160                          $EOPCT
043172  104401  043237                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
043176  013746  001216                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
043202  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
043204  104401  043234                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
043210  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
043214  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
043216  000005                          RESET                   ;;CLEAR THE WORLD
043220  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
043222  000240                          NOP                     ;;SAVE ROOM
043224  000240                          NOP                     ;;FOR
043226  000240                          NOP                     ;;ACT11
043230                          $DOAGN:
043230  000137                          JMP     @(PC)+          ;;RETURN
043232  007740                  $RTNAD: .WORD   ST5
043234    377     377     000   $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
043237    015   042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
043244  050040  051501  020123
043252  000043
```

```
                                .SBTTL  SUBROUTINES

                                ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                ;
043254  012700  003444          CLRFLG: MOV     #DDUMP,R0
043260  012701  177757                  MOV     #-17.,R1
043264  005020                  1$:     CLR     (R0)+
043266  005201                          INC     R1
043270  001375                          BNE     1$
043272  000207                          RTS     PC

                                ;
                                ;TYPE PROGRAM ID IF FTITLE=0
                                ;
043274  005737  001344          TITLE:  TST     FTITLE
043300  001024                          BNE     1$
043302  005237  001344                  INC     FTITLE
043306  104401  054636                  TYPE    ,MSG1                   ;PROGRAM ID
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
043312  005737  000042                  TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
043316  001012                          BNE     64$                     ;;BRANCH IF YES
043320  123727  001230  000001          CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
043326  001406                          BEQ     64$                     ;;BRANCH IF YES
043330  023727  001140  000176          CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
043336  001005                          BNE     65$                     ;;BRANCH IF NO
043340  104406                          GTSWR                           ;;GET SOFT-SWR SETTINGS
043342  000403                          BR      65$
043344  112737  000001  001134  64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
043352                          65$:
043352  000207                  1$:     RTS     PC

                                ;
                                ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
                                ;DRIVS, DRIV0-DRIV7 REGISTERS APPROPRIATELY
                                ;

043354  104411                  GDRVS:  RDLIN
043356  012600                          MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
043360  012701  177770                  MOV     #-8.,R1                 ;SET UP COUNT
043364  112002                  1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
043366  042702  177400                  BIC     #177400,R2              ;MASK HI BYTE
043372  012703  003456                  MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
043376  012704  000060                  MOV     #60,R4

043402  020402                  2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
043404  001415                          BEQ     3$                      ;BRANCH IF YES
043406  005723                          TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
043410  005204                          INC     R4
043412  020427  000070                  CMP     R4,#70
043416  001371                          BNE     2$                      ;S/B 0-7 OR TERMINATOR
043420  005702                          TST     R2
043422  001022                          BNE     4$
043424  020127  177770                  CMP     R1,#-8.
043430  001426                          BEQ     6$                      ;DEFAULT ALL DRIVES
```

```
 8927  043432  005037  003504          7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
 8928  043436  000207                          RTS     PC              ;FOUND TERMINATOR, EXIT
 8929
 8930  043440  005213                  3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
 8931  043442  005237  003454                  INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
 8932  043446  112002                          MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
 8933  043450  042702  177400                  BIC     #177400,R2      ;MASK
 8934  043454  022702  000054                  CMP     #54,R2          ;IS IT A COMMA?
 8935  043460  001407                          BEQ     5$              ;YES, GO TO NEXT WORD.
 8936  043462  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
 8937  043464  001001                          BNE     4$              ;IF NOT, SOMETHING WRONG.
 8938  043466  000761                          BR      7$              ;FOUND TERMINATOR, EXIT
 8939
 8940  043470  104401  057002          4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
 8941  043474  000137  007146                  JMP     PRGSRT          ;START ALL OVER
 8942
 8943  043500  005201                  5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
 8944  043502  001330                          BNE     1$              ;DRIVES TYPED IN.
 8945  043504  000771                          BR      4$              ;IF NORE, HAVE ERROR.
 8946
 8947  043506  005237  003504          6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
 8948  043512  000207                          RTS     PC              ;EXIT.
 8949
 8950                                  ;
 8951                                  ;ROUTINE TO INPUT RKBAS OR DEFAULT.
 8952                                  ;
 8953
 8954  043514  104412                  GBA:    RDOCT
 8955  043516  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 8956  043520  005700                          TST     R0
 8957  043522  001403                          BEQ     1$              ;BRANCH IF DEFAULT.
 8958  043524  010037  001264                  MOV     R0,$BASE
 8959  043530  000207                          RTS     PC
 8960  043532  012737  177440  001264  1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
 8961  043540  000207                          RTS     PC
 8962
 8963                                  ;
 8964                                  ;ROUTINE TO INPUT RKVEC OR DEFAULT
 8965                                  ;
 8966
 8967  043542  104412                  GINT:   RDOCT
 8968  043544  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 8969  043546  005700                          TST     R0
 8970  043550  001405                          BEQ     1$              ;BRANCH IF DEFAULT
 8971  043552  010037  001314                  MOV     R0,RKVEC
 8972  043556  004737  043574          2$:     JSR     PC,SETINT
 8973  043562  000207                          RTS     PC
 8974  043564  012737  000210  001314  1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
 8975  043572  000771                          BR      2$
 8976
 8977                                  ;
 8978                                  ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
 8979                                  ;
 8980
 8981  043574  013700  001314          SETINT: MOV     RKVEC,R0
 8982  043600  012720  050276                  MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
8983  043604  013710  001316          MOV     RKPRI,(R0)      ;PR5 TO RKVEC+2
8984  043610  000207                  RTS     PC
8985
8986
8987                                  ;
8988                                  ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
8989                                  ;ENTER WITH A COUNT IN TEMP1
8990                                  ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
8991                                  ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
8992                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8993                                  ;
8994  043612  032765  000200  000000  FRDY:   BIT     #RDY,RKCS1(R5)
8995  043620  001010                  BNE     1$
8996  043622  005337  003372          DEC     TEMP1
8997  043626  001371                  BNE     FRDY
8998  043630  004737  043746          JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
8999  043634  004737  045066          JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
9000  043640  000207                  RTS     PC              ;NO RDY, EXIT
9001  043642  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
9002  043646  004737  043746          JSR     PC,HOLD
9003  043652  004737  045066          JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
9004  043656  000207                  RTS     PC
9005
9006                                  ;
9007                                  ; ROUTINE TO FIND CONTROLLER READY & STORE DRIVE REGS ONLY
9008                                  ;
9009  043660  032765  000200  000000  FRDY1:  BIT     #RDY,RKCS1(R5)
9010  043666  001014                  BNE     1$
9011  043670  005337  003372          DEC     TEMP1
9012  043674  001371                  BNE     FRDY1
9013  043676  016537  000034  003362          MOV     RKMR2(R5),HMR2
9014  043704  016537  000036  003364          MOV     RKMR3(R5),HMR3
9015  043712  004737  045066          JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
9016  043716  000207                  RTS     PC              ;NO RDY, EXIT
9017  043720  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
9018  043724  016537  000034  003362          MOV     RKMR2(R5),HMR2
9019  043732  016537  000036  003364          MOV     RKMR3(R5),HMR3
9020  043740  004737  045066          JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
9021  043744  000207                  RTS     PC
9022                                  ;
9023                                  ;STORE ALL RK611 REGISTERS IN HOLDING REGS
9024                                  ;
9025
9026  043746  016537  000000  003334  HOLD:   MOV     RKCS1(R5),HCS1
9027  043754  016537  000010  003336          MOV     RKCS2(R5),HCS2
9028  043762  016537  000002  003340          MOV     RKWC(R5),HWC
9029  043770  016537  000004  003342          MOV     RKBA(R5),HBA
9030  043776  016537  000006  003344          MOV     RKDA(R5),HDA
9031  044004  016537  000012  003346          MOV     RKDS(R5),HDS
9032  044012  016537  000014  003350          MOV     RKER(R5),HER
9033  044020  016537  000016  003352          MOV     RKASOF(R5),HASOF
9034  044026  016537  000020  003354          MOV     RKDC(R5),HDC
9035  044034  016537  000026  003360          MOV     RKMR1(R5),HMR1
9036  044042  016537  000034  003362          MOV     RKMR2(R5),HMR2
9037  044050  016537  000036  003364          MOV     RKMR3(R5),HMR3
9038  044056  016537  000030  003366          MOV     RKECPS(R5),HPOS
```

```
 9039  044064  016537  000032  003370          MOV     RKECPT(R5),HPAT
 9040  044072  000207                          RTS     PC
 9041
 9042                                  ;
 9043                                  ;ROUTINE TO CHECK FOR CORRECT ATTN
 9044                                  ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
 9045                                  ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
 9046                                  ;
 9047  044074  010446                  ?STATN: MOV     R4,-(SP)        ;SAV R4
 9048  044076  013704  001222                  MOV     $UNIT,R4
 9049  044102  136437  003324  003353          BITB    ATTN(R4),HASOF+1
 9050  044110  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
 9051  044112  012604                          MOV     (SP)+,R4        ;RESTOR R4
 9052  044114  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
 9053  044120  000207                          RTS     PC
 9054  044122  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
 9055  044124  000207                          RTS     PC
 9056
 9057                                  ;
 9058                                  ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
 9059                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
 9060                                  ;RETURN IF NO ATTN (ERROR CONDITION)
 9061                                  ;RETURN +2 IF ATTN FOUND
 9062                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 9063                                  ;
 9064
 9065  044126  010446                  FATT1:  MOV     R4,-(SP)        ;SAV R4
 9066  044130  012737  177777  003372  3$:     MOV     #-1,TEMP1
 9067  044136  013704  001222                  MOV     $UNIT,R4
 9068  044142  136465  003324  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
 9069  044150  001014                          BNE     2$
 9070  044152  005337  003372                  DEC     TEMP1
 9071  044156  001371                          BNE     1$
 9072  044160  005337  003374                  DEC     TEMP2
 9073  044164  001361                          BNE     3$
 9074  044166  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
 9075  044172  004737  045150                  JSR     PC,GSTAT        ;GET LATEST STATUS
 9076  044176  012604                          MOV     (SP)+,R4        ;RESTOR R4
 9077  044200  000207                          RTS     PC
 9078  044202  005065  000026          2$:     CLR     RKMR1(R5)
 9079  044206  004737  045150                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
 9080  044212  012604                          MOV     (SP)+,R4        ;RESTOR R4
 9081  044214  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
 9082  044220  000207                          RTS     PC
 9083
 9084                                  ;
 9085                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
 9086                                  ;ENTER WITH COUNT IN TEMP1
 9087                                  ;RETURN IF NO ATTN (ERROR)
 9088                                  ;RETURN +2 IF ATTN FOUND
 9089                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 9090                                  ;
 9091
 9092  044222  010446                  FATT2:  MOV     R4,-(SP)        ;SAV R4
 9093  044224  013704  001222          2$:     MOV     $UNIT,R4
 9094  044230  136465  003324  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
```

```
 9095  044236  001011                        BNE     1$
 9096  044240  005337  003372                DEC     TEMP1
 9097  044244  001367                        BNE     2$
 9098  044246  005065  000026                CLR     RKMR1(R5)       ;SELECT WORD 0
 9099  044252  004737  045150                JSR     PC,GSTAT        ;GET LATEST STATUS.
 9100  044256  012604                        MOV     (SP)+,R4        ;RESTOR R4
 9101  044260  000207                        RTS     PC
 9102  044262  005065  000026          1$:   CLR     RKMR1(R5)
 9103  044266  004737  045150                JSR     PC,GSTAT
 9104  044272  012604                        MOV     (SP)+,R4        ;RESTOR R4
 9105  044274  062716  000002                ADD     #2,(SP)         ;SKIP OVER ERROR
 9106  044300  000207                        RTS     PC
 9107                                  ;
 9108                                  ;ENTER WITH A COUNT IN TEMP1
 9109                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
 9110                                  ;WHEN COUNT IS 0...BASED ON AN 11/05.
 9111                                  ;
 9112  044302  005737  003372          DLY:  TST     TEMP1           ;5.6 US
 9113  044306  001403                        BEQ     1$              ;2.5 US
 9114  044310  005337  003372                DEC     TEMP1           ;6.8 US
 9115  044314  000772                        BR      DLY             ;2.5 US
 9116  044316  000207                  1$:   RTS     PC              ;3.8 US
 9117                                  ;
 9118                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
 9119                                  ;
 9120
 9121  044320  104401  056226          BYP:  TYPE    ,MSG14          ;BYPASS DRIVE
 9122  044324  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 9123                                                                ;;TYPE DR#
 9124  044326  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
 9125  044330     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
 9126  044331     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 9127  044332  000207                        RTS     PC
 9128                                  ;
 9129                                  ; THIS ROUTINE READS ALL MSG A&B WORDS & CHECKS THEM AS REQ'D
 9130                                  ;
 9131  044334  017637  000000  001466  CHKMSG: MOV    @(SP),CHKFLG    ;PASS MSGS TO BE TESTED
 9132  044342  062716  000002                ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
 9133  044346  004737  045212                JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS
 9134
 9135  044352  053737  001222  003424        BIS     $UNIT,E.A0      ;SET UNIT #
 9136  044360  053737  001222  003430        BIS     $UNIT,E.A1
 9137  044366  053737  001222  003434        BIS     $UNIT,E.A2
 9138  044374  053737  001222  003440        BIS     $UNIT,E.A3
 9139
 9140  044402  013746  003372                MOV     TEMP1,-(SP)     ;SAVE TEMP 1
 9141
 9142  044406  013737  003424  003372        MOV     E.A0,TEMP1
 9143  044414  004737  047422                JSR     PC,SBPAR        ;GET PARITY FOR MSG A0
 9144  044420  013737  003372  003424        MOV     TEMP1,E.A0
 9145
 9146  044426  013737  003430  003372        MOV     E.A1,TEMP1
 9147  044434  004737  047422                JSR     PC,SBPAR        ;GET PARITY FOR MSG A1
 9148  044440  013737  003372  003430        MOV     TEMP1,E.A1
 9149
 9150  044446  013737  003434  003372        MOV     E.A2,TEMP1
```

```
9151  044454  004737  047422              JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
9152  044460  013737  003372  003434      MOV     TEMP1,E.A2
9153
9154  044466  013737  003426  003372      MOV     E.B0,TEMP1
9155  044474  004737  047422              JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
9156  044500  013737  003372  003426      MOV     TEMP1,E.B0
9157
9158  044506  013737  003432  003372      MOV     E.B1,TEMP1
9159  044514  004737  047422              JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
9160  044520  013737  003372  003432      MOV     TEMP1,E.B1
9161
9162  044526  013737  003436  003372      MOV     E.B2,TEMP1
9163  044534  004737  047422              JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
9164  044540  013737  003372  003436      MOV     TEMP1,E.B2
9165
9166  044546  013737  003442  003372      MOV     E.B3,TEMP1
9167  044554  004737  047422              JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
9168  044560  013737  003372  003442      MOV     TEMP1,E.B3
9169
9170  044566  012637  003372              MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
9171  044572  013737  001176  001172      MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE
9172
9173  044600  023737  003404  003424      CMP     H.A0,E.A0       ;TEST MSG A0
9174  044606  001411                      BEQ     2$              ;BR IF OK
9175  044610  012737  044622  001176      MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
9176  044616  011646                      MOV     (SP),-(SP)      ;COPY RET ADDR
9177  044620  000207                      RTS     PC              ;& RETURN TO MAINLINE ERROR
9178
9179  044622  032777  001000  134310  1$: BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
9180  044630  001107                      BNE     20$             ;& BR IF LOOP ON ERROR
9181  044632  062716  000002          2$: ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
9182
9183  044636  023737  003406  003426      CMP     H.B0,E.B0       ;TEST MSG B0
9184  044644  001411                      BEQ     5$              ;BR IF OK
9185  044646  012737  044660  001176      MOV     #4$,$ESCAPE     ;ELSE SETUP ESCAPE
9186  044654  011646                      MOV     (SP),-(SP)      ;COPY RET ADDR
9187  044656  000207                      RTS     PC              ;& RETURN TO MAINLINE ERROR
9188
9189  044660  032777  001000  134252  4$: BIT     #SW9,@SWR       ;RETURN HERE FROM MAINLINE ERROR
9190  044666  001070                      BNE     20$             ;& BR IF LOOP ON ERROR
9191  044670  062716  000002          5$: ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
9192
9193  044674  023737  003410  003430      CMP     H.A1,E.A1       ;TEST MSG A1
9194  044702  001411                      BEQ     8$              ;BR IF OK
9195  044704  012737  044716  001176      MOV     #7$,$ESCAPE
9196  044712  011646                      MOV     (SP),-(SP)
9197  044714  000207                      RTS     PC
9198
9199  044716  032777  001000  134214  7$: BIT     #SW9,@SWR
9200  044724  001051                      BNE     20$
9201  044726  062716  000002          8$: ADD     #2,(SP)
9202
9203  044732  023737  003412  003432      CMP     H.B1,E.B1       ;TEST MSG B1
9204  044740  001411                      BEQ     11$             ;BR IF OK
9205  044742  012737  044754  001176      MOV     #10$,$ESCAPE
9206  044750  011646                      MOV     (SP),-(SP)
```

```
 9207  044752  000207                          RTS     PC
 9208
 9209  044754  032777  001000  134156  10$:    BIT     #SW9,@SWR
 9210  044762  001032                          BNE     20$
 9211  044764  062716  000002          11$:    ADD     #2,(SP)
 9212
 9213  044770  032737  000001  001466  12$:    BIT     #T.A2,CHKFLG    ;TEST MSG A2?
 9214  044776  001402                          BEQ     13$             ;BR IF NO
 9215  045000  004737  046162                  JSR     PC,RCYLD        ;PUT INFO IN CYLDIF, DO NOT CHECK
 9216
 9217  045004  032737  000002  001466  13$:    BIT     #T.B2,CHKFLG    ;TEST MSG B2?
 9218  045012  001402                          BEQ     14$             ;BR IF NO
 9219  045014  004737  046234                  JSR     PC,RCYLA        ;PUT INFO IN CYLADD, DO NOT CHECK
 9220
 9221  045020  032737  000004  001466  14$:    BIT     #T.B3,CHKFLG    ;TEST MSG B3?
 9222  045026  001404                          BEQ     15$
 9223  045030  004737  046272                  JSR     PC,RSEC         ;PUT INFO IN SECTOR, DO NOT CHECK
 9224  045034  004737  046330                  JSR     PC,RHEAD        ;PUT INFO IN HEADA, DO NOT CHECK
 9225
 9226  045040  013737  001172  001176  15$:    MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
 9227  045046  000207                          RTS     PC
 9228
 9229  045050  012706  001100          20$:    MOV     #STACK,SP       ;RESET STACK PTR
 9230  045054  013737  001172  001176          MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
 9231  045062  000177  134022                  JMP     @$LPERR
 9232
 9233                                  ;
 9234                                  ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
 9235                                  ; IE: IF NED, CTO OR MDS SET, MSG A&B ARE INVALID
 9236                                  ;
 9237  045066  005737  001462          CKCERR: TST     BYPCERR
 9238  045072  001025                          BNE     4$
 9239  045074  032737  100000  003334          BIT     #CERR,HCS1
 9240  045102  001001                          BNE     1$              ;BR IF CERR
 9241  045104  000207                          RTS     PC
 9242
 9243  045106  032737  004000  003334  1$:     BIT     #CTO,HCS1
 9244  045114  001402                          BEQ     2$              ;BR IF NOT CTO
 9245  045116  104326                          ERROR   326             ;CTO ERROR, MSG A&B INVALID
 9246  045120  000207                          RTS     PC
 9247
 9248  045122  032737  010000  003336  2$:     BIT     #NED,HCS2
 9249  045130  001401                          BEQ     3$              ;BR IF NOT NED
 9250  045132  104327                          ERROR   327             ;NED ERROR, MSG A&B INVALID
 9251
 9252  045134  032737  001000  003336  3$:     BIT     #MDS,HCS2
 9253  045142  001401                          BEQ     4$
 9254  045144  104330                          ERROR   330             ;MDS ERROR, MSG A&B INVALID
 9255
 9256  045146  000207                  4$:     RTS     PC
 9257
 9258                                  ;
 9259                                  ;THIS ROUTINE DOES THE SELECT DRIVE CMD TO GET STATUS
 9260                                  ;IT THEN WAITS FOR CONTROLLER  READY.
 9261                                  ;IF RDY NOT RECEIVED BY A TIMEOUT, AN ERROR IS FLAGGED
 9262                                  ;
```

```
 9263  045150  013746  003372          GSTAT:  MOV     TEMP1,-(SP)      ;SAVE TEMP1
 9264  045154  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 9265  045162  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;GET STATUS WITH SELECT DRIVE CMD
 9266  045170  013737  001414  003372          MOV     T10,TEMP1
 9267  045176  004737  043612                  JSR     PC,FRDY          ;FIND RDY
 9268  045202  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRIVE CMD
 9269  045204  012637  003372                  MOV     (SP)+,TEMP1      ;RESTOR TEMP1
 9270  045210  000207                          RTS     PC
 9271
 9272                                  ;
 9273                                  ; THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
 9274                                  ; & ALL CONTROLLER REGISTERS
 9275                                  ;
 9276  045212  013746  003372          GSTAT1: MOV     TEMP1,-(SP)      ;SAVE TEMP 1
 9277  045216  004737  043746                  JSR     PC,HOLD          ;GET ALL CONTR REGS
 9278  045222  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
 9279  045230  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 9280  045236  012765  000003  000026          MOV     #3,RKMR1(R5)     ;SELECT WORD 3
 9281  045244  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)       ;GET STATUS
 9282  045252  013737  001414  003372          MOV     T10,TEMP1
 9283  045260  004737  043660                  JSR     PC,FRDY1         ;FIND RDY & STORE DRIVE REGS ONLY
 9284  045264  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRV CMD
 9285  045266  013737  003362  003420          MOV     HMR2,H.A3        ;STORE MSG A3
 9286  045274  013737  003364  003422          MOV     HMR3,H.B3        ;STORE MSG B3
 9287
 9288  045302  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 9289  045310  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 9290  045316  012765  000002  000026          MOV     #2,RKMR1(R5)     ;SELECT WORD 2
 9291  045324  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
 9292  045332  013737  001414  003372          MOV     T10,TEMP1
 9293  045340  004737  043660                  JSR     PC,FRDY1         ;FIND RDY & STORE DRIVE REGS ONLY
 9294  045344  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRV CMD
 9295  045346  013737  003362  003414          MOV     HMR2,H.A2        ;STORE MSG A2
 9296  045354  013737  003364  003416          MOV     HMR3,H.B2        ;STORE MSG B2
 9297
 9298  045362  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 9299  045370  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 9300  045376  012765  000001  000026          MOV     #1,RKMR1(R5)     ;SELECT WORD 1
 9301  045404  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
 9302  045412  013737  001414  003372          MOV     T10,TEMP1
 9303  045420  004737  043660                  JSR     PC,FRDY1         ;FIND RDY & STORE DRIVE REGS ONLY
 9304  045424  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRV CMD
 9305  045426  013737  003362  003410          MOV     HMR2,H.A1        ;STORE MSG A1
 9306  045434  013737  003364  003412          MOV     HMR3,H.B1        ;STORE MSG B1
 9307
 9308  045442  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 9309  045450  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 9310  045456  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)       ;SELECT WORD 0
 9311  045464  013737  001414  003372          MOV     T10,TEMP1
 9312  045472  004737  043660                  JSR     PC,FRDY1         ;FIND RDY & STORE DRIVE REGS ONLY
 9313  045476  104117                          ERROR   117              ;RDY NOT SET BY END OF SEL DRV CMD
 9314  045500  013737  003362  003404          MOV     HMR2,H.A0        ;STORE MSG A0
 9315  045506  013737  003364  003406          MOV     HMR3,H.B0        ;STORE MSG B0
 9316
 9317  045514  012637  003372                  MOV     (SP)+,TEMP1      ;RESTORE TEMP1
 9318  045520  000207                          RTS     PC
```

```
                                        ;
                                        ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
                                        ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
                                        ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
                                        ;RETURN IF CERR SET
                                        ;RETURN +2 IF CERR CLEAR
                                        ;
045522  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
045530  013737  001414  003372          MOV     T10,TEMP1
045536  004737  043612                  JSR     PC,FRDY         ;FIND RDY
045542  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
045544  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
045552  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
045556  004737  045150                  JSR     PC,GSTAT        ;GET STATUS
045562  032737  100000  003334          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
045570  001401                          BEQ     1$
045572  000207                          RTS     PC
045574  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
045600  000207                          RTS     PC
                                        ;
                                        ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                        ;
045602  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
045610  004737  045150                  JSR     PC,GSTAT
045614  013737  003364  001406          MOV     HMR3,SECTOR
045622  042737  177017  001406          BIC     #^C<M.SECT>,SECTOR
045630  006237  001406                  ASR     SECTOR          ;RIGHT JUSTIFY
045634  006237  001406                  ASR     SECTOR          ;SECTOR
045640  006237  001406                  ASR     SECTOR          ;INFO
045644  006237  001406                  ASR     SECTOR
045650  000207                          RTS     PC
                                        ;
                                        ;FIND SECTOR 0 IN 22 SECTOR FORMAT.
                                        ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
                                        ;
045652  013746  003372  FSO22:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
045656  013737  001424  003372          MOV     T5000,TEMP1     ;SETUP TIMEOUT
045664  004737  045602          1$:     JSR     PC,RDSEC        ;READ SECTOR
045670  005737  001406                  TST     SECTOR          ;LOOK FOR SECTOR 0
045674  001005                          BNE     2$
045676  004737  045602                  JSR     PC,RDSEC
045702  005737  001406                  TST     SECTOR
045706  001406                          BEQ     3$              ;BR IF SAME TWICE
045710  005337  003372          2$:     DEC     TEMP1
045714  001363                          BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
045716  012637  003372                  MOV     (SP)+,TEMP1     ;ELSE RESTORE TEMP1
045722  000207                          RTS     PC              ;EXIT
045724  012637  003372          3$:     MOV     (SP)+,TEMP1
045730  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
045734  000207                          RTS     PC
                                        ;
```

```
                                        ;FIND NEXT SECTOR IN 22 SECTOR FORMAT
                                        ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
                                        ;
045736 013746 003372            FNS22:  MOV     TEMP1,-(SP)     ;SAVE TEMP 1
045742 013737 001420 003372             MOV     T500,TEMP1      ;SETUP TIMEOUT
045750 004737 045602            1$:     JSR     PC,RDSEC                ;READ SECTOR
045754 023737 001402 001406             CMP     PSEC,SECTOR
045762 001406                           BEQ     3$              ;BR IF SAME
045764 004737 045602                    JSR     PC,RDSEC                ;ELSE TRY READ DIFFERENT TWICE
045770 023737 001402 001406             CMP     PSEC,SECTOR
045776 001006                           BNE     2$              ;BR IF DIFFERENT TWICE
046000 005337 003372            3$:     DEC     TEMP1           ;ELSE TRY AGAIN IF TIME LEFT
046004 001361                           BNE     1$
046006 012637 003372                    MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
046012 000207                           RTS     PC
046014 012637 003372            2$:     MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
046020 062716 000002                    ADD     #2,(SP)         ;SKIP OVER ERROR
046024 000207                           RTS     PC
                                        ;
                                        ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                        ;
046026 012765 000002 000026     RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
046034 004737 045150                    JSR     PC,GSTAT
046040 013737 003362 001362             MOV     HMR2,CYLDIF
046046 042737 160017 001362             BIC     #↑C<M.CDIF>,CYLDIF
046054 006237 001362                    ASR     CYLDIF          ;RIGHT JUSTIFY
046060 006237 001362                    ASR     CYLDIF          ;CYL DIFF/OFFSET
046064 006237 001362                    ASR     CYLDIF          ;INFO
046070 006237 001362                    ASR     CYLDIF
046074 023727 001362 000777             CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
046102 001002                           BNE     1$              ;BR IF NOT
046104 005037 001362                    CLR     CYLDIF          ;CLR IF YES
046110 000207                   1$:     RTS     PC
                                        ;
                                        ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
                                        ;
046112 012765 000002 000026     RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
046120 004737 045150                    JSR     PC,GSTAT
046124 013737 003364 001364             MOV     HMR3,CYLADD
046132 042737 160017 001364             BIC     #↑C<M.CADD>,CYLADD
046140 006237 001364                    ASR     CYLADD          ;RIGHT JUSTIFY
046144 006237 001364                    ASR     CYLADD          ;CYL ADDR
046150 006237 001364                    ASR     CYLADD          ;INFO
046154 006237 001364                    ASR     CYLADD
046160 000207                           RTS     PC
                                        ;
                                        ; READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                        ;
046162 013737 003414 001362     RCYLD:  MOV     H.A2,CYLDIF
046170 042737 160017 001362             BIC     #↑C<M.CDIF>,CYLDIF      ;CLEAR UNWANTED INFO
046176 006237 001362                    ASR     CYLDIF          ;RIGHT JUSTIFY
046202 006237 001362                    ASR     CYLDIF
046206 006237 001362                    ASR     CYLDIF
046212 006237 001362                    ASR     CYLDIF
046216 023727 001362 000777             CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
046224 001002                           BNE     1$              ;BR IF NO
```

```
 9431  046226  005037  001362                  CLR     CYLDIF          ;ELSE CLEAR
 9432  046232  000207                  1$:     RTS     PC
 9433                                  ;
 9434                                  ; READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
 9435                                  ;
 9436  046234  013737  003416  001364  RCYLA:  MOV     H.B2,CYLADD
 9437  046242  042737  160017  001364          BIC     #↑C<M.CADD>,CYLADD      ;CLEAR UNWANTED INFO
 9438  046250  006237  001364                  ASR     CYLADD          ;RIGHT JUSTIFY
 9439  046254  006237  001364                  ASR     CYLADD
 9440  046260  006237  001364                  ASR     CYLADD
 9441  046264  006237  001364                  ASR     CYLADD
 9442  046270  000207                          RTS     PC
 9443                                  ;
 9444                                  ; READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
 9445                                  ;
 9446  046272  013737  003422  001406  RSEC:   MOV     H.B3,SECTOR
 9447  046300  042737  177017  001406          BIC     #↑C<M.SECT>,SECTOR      ;CLEAR UNWANTED INFO
 9448  046306  006237  001406                  ASR     SECTOR          ;RIGHT JUSTIFY
 9449  046312  006237  001406                  ASR     SECTOR
 9450  046316  006237  001406                  ASR     SECTOR
 9451  046322  006237  001406                  ASR     SECTOR
 9452  046326  000207                          RTS     PC
 9453                                  ;
 9454                                  ; READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
 9455                                  ;
 9456  046330  013737  003422  001432  RHEAD:  MOV     H.B3,HEADA
 9457  046336  042737  170777  001432          BIC     #↑C<M.HEAD>,HEADA       ;CLEAR UNWANTED INFO
 9458  046344  006237  001432                  ASR     HEADA           ;RIGHT JUSTIFY IT
 9459  046350  000337  001432                  SWAB    HEADA
 9460  046354  000207                          RTS     PC
 9461                                  ;
 9462                                  ;FIND LIMIT DETECT ON SEEK IN RKMR3 BEFORE TIMEOUT
 9463                                  ;RETURN IF NOT FOUND: ERROR
 9464                                  ;RETURN+2 IF FOUND: SKIP OVER ERROR
 9465                                  ;
 9466  046356  005037  001460          FLIM:   CLR     LIMERR          ;LIMIT DETECT ERROR FLAG
 9467  046362  012737  177777  003372          MOV     #-1,TEMP1       ;SETUP TIMEOUT
 9468  046370  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
 9469  046376  004737  045150          1$:     JSR     PC,GSTAT
 9470  046402  032737  020000  003364          BIT     #D.LIMD,HMR3
 9471  046410  001006                          BNE     2$              ;EXIT IF SET
 9472  046412  005337  003372                  DEC     TEMP1
 9473  046416  001367                          BNE     1$
 9474  046420  005237  001460                  INC     LIMERR          ;SET LIMIT DETECT FLAG
 9475  046424  000207                          RTS     PC
 9476  046426  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 9477  046432  000207                          RTS     PC
 9478                                  ;
 9479                                  ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE TIMEOUT
 9480                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
 9481                                  ;RETURN IF NOT FOUND
 9482                                  ;RETURN+2 IF FOUND - SKIP OVER ERROR
 9483                                  ;
 9484  046434  012737  177777  003372  FHDHM:  MOV     #-1,TEMP1       ;ALL 1'S
 9485  046442  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
 9486  046450  004737  045150          1$:     JSR     PC,GSTAT
```

```
 9487  046454  032737  000040  003362          BIT     #D.HDHM,HMR2
 9488  046462  001007                          BNE     2$
 9489  046464  005337  003372                  DEC     TEMP1
 9490  046470  001367                          BNE     1$
 9491  046472  005337  003374                  DEC     TEMP2
 9492  046476  001356                          BNE     FHDHM
 9493  046500  000207                          RTS     PC
 9494  046502  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 9495  046506  000207                          RTS     PC
 9496                                  ;
 9497                                  ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE TIMEOUT
 9498                                  ;RETURN IF NOT FOUND
 9499                                  ;RETURN+2 IF FOUND: SKIP OVER ERROR
 9500                                  ;
 9501  046510  012737  177777  003372  FLOAD:  MOV     #-1,TEMP1       ;SETUP TIMEOUT
 9502  046516  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
 9503  046524  004737  045150          1$:     JSR     PC,GSTAT
 9504  046530  032737  010000  003362          BIT     #D.LOAD,HMR2
 9505  046536  001004                          BNE     2$
 9506  046540  005337  003372                  DEC     TEMP1
 9507  046544  001367                          BNE     1$
 9508  046546  000207                          RTS     PC
 9509  046550  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 9510  046554  000207                          RTS     PC
 9511                                  ;
 9512                                  ;ROUTINE TO FIND SPOK BEFORE TIMEOUT
 9513                                  ;ENTER WITH APPROX TIME IN TEMP2
 9514                                  ;RETURN IF NOT CLEARED
 9515                                  ;RETURN +2 IF CLEARED TO SKIP OVER ERROR
 9516                                  ;
 9517  046556  012737  177777  003372  FSPOK:  MOV     #-1,TEMP1       ;ALL 1'S
 9518  046564  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
 9519  046572  004737  045150          1$:     JSR     PC,GSTAT
 9520  046576  032737  001000  003362          BIT     #D.SPOK,HMR2    ;SEE IF SPOK CLEARED
 9521  046604  001407                          BEQ     2$
 9522  046606  005337  003372                  DEC     TEMP1
 9523  046612  001367                          BNE     1$
 9524  046614  005337  003374                  DEC     TEMP2
 9525  046620  001356                          BNE     FSPOK
 9526  046622  000207                          RTS     PC
 9527  046624  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 9528  046630  000207                          RTS     PC
 9529                                  ;
 9530                                  ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
 9531                                  ;ENTER WITH CYL # IN 'CALADD'
 9532                                  ;ENTER WITH HEAD # IN 'HEAD'
 9533                                  ;ENTER WITH FORMAT IN 'FORMAT'
 9534                                  ;
 9535  046632  010046                  FHDTAB: MOV     R0,-(SP)        ;SAV R0
 9536  046634  010146                          MOV     R1,-(SP)        ;SAV R1
 9537  046636  012700  001470                  MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
 9538  046642  005001                          CLR     R1              ;SECTOR COUNTER
 9539  046644  013737  001430  001434          MOV     HEAD,HD1
 9540  046652  006337  001434                  ASL     HD1
 9541  046656  006337  001434                  ASL     HD1
 9542  046662  006337  001434                  ASL     HD1
```

```
9543  046666  006337  001434                ASL     HD1
9544  046672  006337  001434                ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
9545  046676  013737  001436  001440        MOV     FORMAT,FMT1
9546  046704  000337  001440                SWAB    FMT1
9547  046710  006337  001440                ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
9548
9549  046714  013720  001366        1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
9550  046720  010110                        MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
9551  046722  053710  001434                BIS     HD1,(R0)        ;               -HEAD NO
9552  046726  053710  001440                BIS     FMT1,(R0)       ;               -FORMAT
9553  046732  005737  001464                TST     BYPFMT
9554  046736  001403                        BEQ     2$              ;BR IF TRUE FORMAT
9555  046740  052710  140000                BIS     #<BIT14!BIT15>,(R0)     ;SET GOOD SECTOR FLAGS
9556  046744  000402                        BR      3$
9557  046746  004737  047026        2$:     JSR     PC,SECFLG       ;GET  SECTOR FLAGS
9558
9559  046752  013737  001366  003372  3$:     MOV     CALADD,TEMP1
9560  046760  011037  003374                MOV     (R0),TEMP2
9561  046764  043737  001366  003374        BIC     CALADD,TEMP2
9562  046772  042037  003372                BIC     (R0)+,TEMP1
9563  046776  053737  003372  003374        BIS     TEMP1,TEMP2
9564  047004  013720  003374                MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
9565
9566  047010  005201                        INC     R1              ;SECTOR CTR
9567  047012  020127  000026                CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
9568  047016  001336                        BNE     1$              ;BR IF NO
9569
9570  047020  012601                        MOV     (SP)+,R1        ;RESTOR R1
9571  047022  012600                        MOV     (SP)+,R0        ;RESTOR R0
9572  047024  000207                        RTS     PC
9573
9574                                ;
9575                                ; THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS TEST
9576                                ; & SETS BITS 14 & 15 APPROPRIATELY.
9577                                ;
9578  047026  010246        SECFLG: MOV     R2,-(SP)        ;SAVE R2
9579  047030  005737  001436                TST     FORMAT
9580  047034  001016                        BNE     1$              ;BR IF 20 SECTOR FORMAT
9581  047036  012702  002314                MOV     #BSE22H+8.,R2
9582  047042  004737  047076                JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
9583  047046  052710  100000                BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
9584
9585  047052  012702  054502                MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
9586  047056  004737  047076                JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
9587  047062  052710  040000                BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
9588
9589  047066  012602                        MOV     (SP)+,R2        ;ELSE RETURN HERE
9590  047070  000207                        RTS     PC
9591
9592
9593  047072  012602        1$:     MOV     (SP)+,R2        ;RESTORE R2
9594  047074  000207                RTS     PC
9595
9596                                ;
9597                                ; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
9598                                ; ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC) IN TEMP1
```

```
                                        ; RETURN IF NO COMPARE
                                        ; RETURN +4 IF COMPARE
                                        ;
047076  010346                  FLGTST: MOV     R3,-(SP)        ;SAVE R3
                                1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
047100  021227  177777                  BNE     2$              ;BR IF NO
047104  001002                          MOV     (SP)+,R3        ;RESTORE R3
047106  012603                          RTS     PC
047110  000207
                                2$:     CMP     (R2)+,CALADD    ;SEE IF = CYL #, & ADV PTR TO TRACK/SECTOR WORD
047112  022237  001366                  BEQ     3$
047116  001403                          ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
047120  062702  000002                  BR      1$
047124  000765
                                3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
047126  013703  001430                  SWAB    R3
047132  000303                          BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
047134  050103                          CMP     (R2)+,R3        ;SECTOR/HEAD COMPARE? & INCR TO NEXT CYL WORD
047136  022203                          BEQ     4$              ;BR IF YES
047140  001401                          BR      1$              ;TRY NEXT CYL
047142  000756
                                4$:     MOV     (SP)+,R3        ;RESTORE R3
047144  012603                          ADD     #4,(SP)         ;INCREMENT RET ADDR
047146  062716  000004                  RTS     PC
047152  000207
                                ;
                                ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
                                ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
                                ;
047154  010046                  SORT:   MOV     R0,-(SP)        ;SAVE R0
047156  010146                          MOV     R1,-(SP)        ;SAVE R1
047160  004737  045602                  JSR     PC,RDSEC
047164  062737  000001  001406          ADD     #1,SECTOR
047172  004737  047262                  JSR     PC,MULT6        ;MULT SECTOR BY 6
047176  012700  000204                  MOV     #132.,R0
047202  163700  001406                  SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
047206  010037  001406                  MOV     R0,SECTOR
047212  062737  001674  001406          ADD     #RHTAB,SECTOR   ;SAVE INDEX
047220  062700  001674                  ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
047224  012701  002100                  MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
                                1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
047230  012021
047232  020027  002100                  CMP     R0,#RHTAB+132.
047236  001374                          BNE     1$
047240  012700  001674                  MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
                                2$:     MOV     (R0)+,(R1)+
047244  012021
047246  020037  001406                  CMP     R0,SECTOR
047252  001374                          BNE     2$
047254  012601                          MOV     (SP)+,R1        ;RESTOR R1
047256  012600                          MOV     (SP)+,R0        ;RESTOR R0
047260  000207                          RTS     PC
```

```
                                        ;
                                        ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
                                        ;
047262  006337  001406          MULT6:  ASL     SECTOR          ;2 X SECTOR
047266  013746  001406                  MOV     SECTOR,-(SP)
047272  006337  001406                  ASL     SECTOR          ;4 X SECTOR
047276  062637  001406                  ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
047302  000207                          RTS     PC
                                        ;
                                        ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
                                        ;
047304  005037  001376          CLKON:  CLR     TIMUP
047310  005737  003500                  TST     PCLKF
047314  001004                          BNE     1$              ;BRANCH IF P-CLOCK PRESENT
047316  012777  000100  132002          MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
047324  000207                          RTS     PC
047326  012777  177777  131766  1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
047334  012777  000135  131756          MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
047342  000207                          RTS     PC              ;LINE FREQ & RUN
                                        ;
                                        ;KW11-L & KW11-P INTERRUPT HANDLER
                                        ;
047344  005037  001376          CLOCK:  CLR     TIMUP
047350  005337  001372                  DEC     COUNT
047354  001010                          BNE     1$
047356  013737  001370  001372          MOV     HZ,COUNT
047364  005337  001374                  DEC     SEC
047370  001002                          BNE     1$
047372  005237  001376                  INC     TIMUP           ;SORRY, TIME IS UP
047376  000002                  1$:     RTI
                                        ;
                                        ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
                                        ;
047400  005737  003500          CLKOF:  TST     PCLKF
047404  001003                          BNE     1$              ;BRACH IF P-CLOCK PRESENT
047406  005077  131714                  CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
047412  000207                          RTS     PC
047414  005077  131700          1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
047420  000207                          RTS     PC
                                        ;
                                        ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MSGS
                                        ;ENTER WITH THE EXPECTED WORD IN TEMP1
                                        ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
                                        ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
                                        ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
                                        ; THE PARITY BIT IS NOT SET IN B .
                                        ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
                                        ;SET IN TEMP1
                                        ;
047422  010046                  SBPAR:  MOV     R0,-(SP)        ;SAVE R0
047424  010146                          MOV     R1,-(SP)        ;SAVE R1
047426  012700  000021                  MOV     #17.,R0         ;SHIFT COUNTER
047432  005001                          CLR     R1              ;COUNT # OF 1'S IN TEMP1
047434  000241                          CLC                     ;CLEAR CARRY
```

```
                                1$:     ROL     TEMP1
047436  006137  003372                  BCC     2$              ;BR IF CARRY CLEAR
047442  103001                          INC     R1              ;COUNT # OF 1'S
047444  005201                  2$:     DEC     R0              ;SHIFT COUNTER
047446  005300                          BNE     1$
047450  001372
                                        BIT     #BIT0,R1
047452  032701  000001                  BNE     3$              ;BR IF ODD # IN R0
047456  001003                          BIS     #M.PAR,TEMP1    ;SET PARITY BIT
047460  052737  100000  003372
047466  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
047470  012600                          MOV     (SP)+,R0        ;RESTORE R0
047472  000207                          RTS     PC

                                ;
                                ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                ; IE: MY LOOP MACRO
                                ;
047474  032777  001000  131436  $SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
047502  001406                          BEQ     1$              ;BR IF NO
047504  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
047510  001403                          BEQ     1$              ;BR IF NO
047512  013716  001110                  MOV     $LPERR,(SP)
047516  000002                          RTI
047520  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
047524  000002                          RTI

                                ;
                                ;CHECK FOR SW14 (LOOP ON TEST) OR SW8 (LOOP ON SPECIFIC TEST)
                                ;
                                ;RETURN IF NEITHER SET
                                ;RETURN +2 IF EITHER SET
                                ;
                                ;THIS SUBROUTINE IS USED AT THE END OF ANY TEST THAT REQUIRES
                                ;RECONDITIONING OF THE DRIVE BEFORE LOOPING ON AN ERROR OR TEST
                                ;
047526  005037  001176          $SWTST: CLR     $ESCAPE
047532  005037  001410                  CLR     LPFLG
047536  032777  040000  131374          BIT     #SW14,@SWR      ;LOOP ON TEST?
047544  001403                          BEQ     3$              ;BR IF NO
047546  062716  000002          1$:     ADD     #2,(SP)
047552  000207                  2$:     RTS     PC

047554  032777  000400  131356  3$:     BIT     #SW8,@SWR       ;LOOP ON SPECIFIC TEST?
047562  001773                          BEQ     2$              ;BR IF NO
047564  127737  131350  001102          CMPB    @SWR,$TSTNM     ;RIGHT TEST? SWR <7:0>
047572  001765                          BEQ     1$              ;BR IF YES
047574  000207                          RTS     PC

                                ;
                                ;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
                                ;IT IS USED TO ALLOW THE OPERATOR TO HALT THE CPU WHILE INSURING
                                ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
                                ;THE CPU.
```

```
                                        ;
047576  022626                  STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT

047600  004737  045522                  JSR     PC,SUBCLR
047604  104024                  ERROR   24              ;CERR AFTER

047606  005737  003316                  TST     UNLD            ;SEE IF HEADS UNLOADED
047612  001434                  BEQ     3$              ;BR IF NO
047614  005737  000042                  TST     42              ;SEE IF MANUAL OR AUTO MODE
047620  001403                  BEQ     1$              ;BR IF MANUAL MODE
047622  104401  056676                  TYPE    ,MSG74          ;PGM ABORT PENDING
047626  000402                  BR      2$
047630  104401  056725          1$:     TYPE    ,MSG75          ;HALT PENDING
047634                          2$:

047634  004737  045522                  JSR     PC,SUBCLR
047640  104024                  ERROR   24              ;CERR AFTER SCLR

047642  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
047650  013737  001414  003372          MOV     T10,TEMP1       ;SET TIMEOUT
047656  004737  043612                  JSR     PC,FRDY         ;FIND RDY
047662  104121                  ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.

047664  013737  001420  003374          MOV     T500,TEMP2      ;SETUP  TIMEOUT
047672  004737  044126                  JSR     PC,FATT1        ;FIND ATTN
047676  104067                  ERROR   67              ;NO ATTN AFTER ST SPIN CMD.

047700  005037  003316                  CLR     UNLD

047704  005737  003320          3$:     TST     BADHDR          ;SEE IF HEADERS VALID
047710  001466                  BEQ     4$              ;BR IF YES
047712  005237  003322                  INC     HPEND

047716  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
047724  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
047732  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                        ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                        ;IN RKMR2 & RKMR3 RESP.
047740  013737  001414  003372          MOV     T10,TEMP1
047746  004737  043612                  JSR     PC,FRDY         ;FIND RDY
047752  104124                  ERROR   124             ;RDY NOT SET AFTER RECAL CMD

047754  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
047762  004737  045150                  JSR     PC,GSTAT
047766  032737  020000  003362          BIT     #D.RTZ,HMR2
047774  001001                  BNE     64$
047776  104244                  ERROR   244             ;RTZ NOT SET DURING RECAL CMD
050000  013737  001414  003374  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
050006  004737  044126                  JSR     PC,FATT1        ;FIND ATTN
050012  104055                  ERROR   55              ;NO ATTN AFTER RECAL CMD

050014  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
050022  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
050030  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
050036  013737  001414  003372          MOV     T10,TEMP1
050044  004737  043612                  JSR     PC,FRDY         ;FIND RDY
```

```
 9823  050050  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 9824  050052  004737  044074                JSR     PC,TSTATN       ;TEST FOR ATTN
 9825  050056  000401                        BR      65$
 9826  050060  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 9827  050062                        65$:
 9828
 9829
 9830  050062  000137  031234                JMP     FORM            ;WRITE VALID FORMATS
 9831
 9832  050066  005737  000042        4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
 9833  050072  001410                        BEQ     5$              ;BR IF MANUAL MODE
 9834  050074  104401  056747                TYPE    ,MSG76          ;PGM ABORTED
 9835  050100  005037  043160                CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
 9836  050104  005037  001176                CLR     $ESCAPE
 9837  050110  000137  043132                JMP     $EOP1           ;ABORT PROGRAM
 9838
 9839  050114  104401  056765        5$:     TYPE    ,MSG77          ;CPU HALTED
 9840  050120  000000                        HALT
 9841  050122  000137  007740                JMP     ST5             ;START OVER IF CONTINUE PRESSED
 9842
 9843                                ;
 9844                                ;CHECK IF HALT PENDING
 9845                                ;RET IF YES
 9846                                ;RET+4 IF NO
 9847                                ;
 9848  050126  005037  003320        HPEN:   CLR     BADHDR          ;CLR VALID HALT FLAG
 9849  050132  005737  003322                TST     HPEND           ;SEE IF HALT PENDING
 9850  050136  001002                        BNE     1$              ;BR IF YES
 9851  050140  062716  000004                ADD     #4,(SP)         ;ELSE BUMP RET ADDR
 9852  050144  000207                1$:     RTS     PC              ;& RET
 9853
 9854                                .SBTTL  UNEXPECTED TIMEOUT HANDLER
 9855
 9856                                ;
 9857                                ;THIS ROUTINE IS ENTERED IF THERE IS
 9858                                ;       A.  NON EXISTANT MEMORY (NO SSYN)
 9859                                ;       B.  BOUNDRY ERROR
 9860                                ;       C.  STACK OVERFLOW
 9861                                ;
 9862
 9863  050146  011600                BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
 9864  050150  005740                        TST     -(R0)           ;GET PC BEFORE UPDATE
 9865  050152  032777  020000  130760        BIT     #SW13,@SWR      ;INHIBIT ERROR TYPOUT?
 9866  050160  001005                        BNE     1$              ;YES, DON'T TYPE
 9867  050162  104401  057143                TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
 9868  050166  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 9869                                                                ;;TYPE PC
 9870  050170  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
 9871  050172     006                        .BYTE   6               ;;TYPE 6 DIGIT(S)
 9872  050173     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 9873  050174  032777  001000  130736  1$:   BIT     #SW9,@SWR       ;LOOP ON ERROR?
 9874  050202  001403                        BEQ     2$              ;NO, BRANCH
 9875  050204  022626                        CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
 9876  050206  000177  130674                JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
 9877                                                                ;THAT GAVE BAD TIMEOUT
 9878  050212  032777  040000  130720  2$:   BIT     #SW14,@SWR      ;LOOP ON TEST?
```

```
 9879  050220  001401                        BEQ     3$              ;NO BRANCH
 9880  050222  000002                        RTI                     ;YES
 9881
 9882  050224  000000                3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
 9883                                                                ;AS INDICATED. YOU CAN LOOP ON
 9884                                                                ;ERROR, LOOP ON TEST OR INHIBIT
 9885                                                                ;ERROR TYPEOUT BY SETTING THOSE
 9886                                                                ;SWITCHES.
 9887                                                                ;
 9888  050226  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK
 9889  050230  000137  043132                JMP     $EOP1           ;ABORT TESTS
 9890
 9891                          .SBTTL  MEMORY CHECK ENABLE TRAP
 9892
 9893  050234  012737  050250  001176  MEMERR: MOV     #1$,$ESCAPE     ;LOAD ESCAPE
 9894  050242  011637  001334                MOV     (SP),TRAPPC     ;STORE PC
 9895  050246  104236                        ERROR   236             ;UNEXP MEM PARITY TRAP
 9896
 9897  050250  005037  001176          1$:   CLR     $ESCAPE
 9898  050254  032777  001000  130656        BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
 9899  050262  001001                        BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
 9900  050264  000002                        RTI                     ;ELSE RETURN
 9901
 9902  050266  012706  001100          2$:   MOV     #STACK,SP       ;INIT STACK
 9903  050272  000177  130612                JMP     @$LPERR         ;LOOP ON ERROR
 9904
 9905                          .SBTTL  RK06 INTERRUPT HANDLER
 9906
 9907  050276  011600                INTER:  MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
 9908  050300  005740                        TST     -(R0)           ;GET PC BEFORE UPDATE.
 9909  050302  104401  055671                TYPE    ,MSG6           ;INT AT PC=
 9910  050306  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 9911                                                                ;;TYPE PC
 9912  050310  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
 9913  050312     006                        .BYTE   6               ;;TYPE 6 DIGIT(S)
 9914  050313     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 9915  050314  000002                        RTI
 9916
 9917                          .SBTTL  POWER DOWN AND UP ROUTINES
 9918
 9919                          ;POWER DOWN ROUTINE
 9920
 9921  050316  012737  050330  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
 9922  050324  000000                        HALT
 9923  050326  000776                        BR      .-2             ;HANG UP.
 9924
 9925                          ;POWER UP ROUTINE
 9926
 9927  050330  005037  050402  $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
 9928  050334  005237  050402  1$:     INC     $PWRCT          ;WAIT FOR THE INCR
 9929  050340  001375                        BNE     1$              ;OF WORD
 9930  050342  012737  050316  000024        MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
 9931  050350  012737  000340  000026        MOV     #PR7,PWRVEC+2   ;PRIORITY 7
 9932  050356  012737  000340  000036        MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
 9933  050364  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK
 9934  050370  104401  056057                TYPE    ,MSG11          ;REPORT POWER FAIL
```

N14


```
 9935  050374  000005                  RESET
 9936  050376  000137  011620          JMP     PFSRT
 9937
 9938  050402  000000                  $PWRCT: 0                       ;WAIT COUNT FOR TTY
 9939
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE

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

050404                          $SCOPE:
050404  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
050406  032777  040000  130524  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
050414  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
050416  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
050420  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
050424  012737  050444  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
050432  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
050436  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050442  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
050444  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
050446  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050452  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
050454                          6$:;#####END OF CODE FOR THE XOR TESTER#####
050454  032777  000400  130456          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
050462  001404                          BEQ     2$              ;;BR IF NO
050464  127737  130450  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
050472  001465                          BEQ     $OVER           ;;BR IF YES
050474  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
050500  001421                          BEQ     3$              ;;BR IF NO
050502  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
050510  101015                          BHI     3$              ;;BR IF NO
050512  032777  001000  130420          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
050520  001404                          BEQ     4$              ;;BR IF NO
050522  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
050530  000446                          BR      $OVER
050532  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
050536  005037  001174                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
050542  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
050544  032777  004000  130366  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
050552  001011                          BNE     1$              ;;BR IF YES
050554  005737  001216                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
050560  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
050562  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
050566  023737  001174  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
050574  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
050576  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
050604  013737  050662  001174          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
050612  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
050616  113737  001102  001214          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
  9996  050624  011637  001106                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
  9997  050630  011637  001110                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
  9998  050634  005037  001176                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
  9999  050640  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 10000  050646  013777  001102  130266 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 10001  050654  013716  001106                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 10002  050660  000002                        RTI                     ;;FIXES PS
 10003  050662  003720                $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 10004                                  .SBTTL  ERROR HANDLER ROUTINE
 10005
 10006                                  ;;**************************************************************
 10007                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 10008                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 10009                                  ;*AND GO TO TYPERR ON ERROR
 10010                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 10011                                  ;*SW15=1        HALT ON ERROR
 10012                                  ;*SW13=1        INHIBIT ERROR TYPEOUTS
 10013                                  ;*SW10=1        BELL ON ERROR
 10014                                  ;*SW09=1        LOOP ON ERROR
 10015                                  ;*CALL
 10016                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 10017
 10018  050664                         $ERROR:
 10019  050664  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 10020  050666  105237  001103        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 10021  050672  001775                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 10022  050674  013777  001102  130240        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 10023  050702  032777  002000  130230        BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 10024  050710  001402                        BEQ     1$              ;;NO - SKIP
 10025  050712  104401  001200                TYPE    ,$BELL          ;;RING BELL
 10026  050716  005237  001112        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 10027  050722  011637  001116                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 10028  050726  162737  000002  001116        SUB     #2,$ERRPC
 10029  050734  117737  130156  001114        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 10030  050742  032777  020000  130170        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 10031  050750  001004                        BNE     20$             ;;SKIP TYPEOUTS
 10032  050752  004737  067352                JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 10033  050756  104401  001205                TYPE    ,$CRLF
 10034  050762                         20$:
 10035  050762  122737  000001  001230        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 10036  050770  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
 10037  050772  113737  001114  051004        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 10038  051000  004737  051610                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 10039  051004     000                 21$:    .BYTE   0
 10040  051005     000                         .BYTE   0
 10041  051006  000777                22$:    BR      22$             ;;APT ERROR LOOP
 10042  051010  005777  130124        2$:     TST     @SWR            ;;HALT ON ERROR
 10043  051014  100002                        BPL     3$              ;;SKIP IF CONTINUE
 10044  051016  000000                        HALT                    ;;HALT ON ERROR!
 10045  051020  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 10046  051022  032777  001000  130110 3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 10047  051030  001402                        BEQ     4$              ;;BR IF NO
 10048  051032  013716  001110                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 10049  051036  005737  001176        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 10050  051042  001402                        BEQ     5$              ;;BR IF NONE
 10051  051044  013716  001176                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
10052  051050
10053  051050  022737  043220  000042  5$:     CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
10054  051056  001001                          BNE     6$              ;;BRANCH IF NO
10055  051060  000000                          HALT                    ;;YES
10056  051062                          6$:
10057  051062  000002                          RTI                     ;;RETURN
10058                                  .SBTTL  TYPE ROUTINE
10059
10060                                  ;;*****************************************************************
10061                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10062                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10063                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10064                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10065                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10066                                  ;*
10067                                  ;*CALL:
10068                                  ;*1) USING A TRAP INSTRUCTION
10069                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10070                                  ;*OR
10071                                  ;*      TYPE
10072                                  ;*      MESADR
10073                                  ;*
10074
10075  051064  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
10076  051070  100002                  BPL     1$              ;;BR IF YES
10077  051072  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
10078  051074  000430                  BR      3$              ;;LEAVE
10079  051076  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
10080  051100  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
10081  051104  122737  000001  001230  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
10082  051112  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
10083  051114  132737  000100  001231  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
10084  051122  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
10085  051124  010037  051134          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
10086  051130  004737  051600          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
10087  051134  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
10088  051136  132737  000040  001231  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
10089  051144  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
10090  051146  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
10091  051150  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
10092  051152  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
10093  051154  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
10094  051156  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
10095  051162  000002                  RTI                     ;;RETURN
10096  051164  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
10097  051170  001430                  BEQ     8$
10098  051172  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
10099  051176  001006                  BNE     5$
10100  051200  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
10101  051202  104401                  TYPE                    ;;TYPE A CR AND LF
10102  051204  001205                  $CRLF
10103  051206  105037  051342          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
10104  051212  000755                  BR      2$              ;;GET NEXT CHARACTER
10105  051214  004737  051276  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
10106  051220  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
10107  051224  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
10108  051226  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
10109                                                                   ;;AND THE NULL CHAR.
10110  051232  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
10111  051236  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
10112  051240  004737  051276                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
10113  051244  105337  051342                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
10114  051250  000770                          BR      7$              ;;LOOP
10115
10116                                  ;HORIZONTAL TAB PROCESSOR
10117
10118  051252  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
10119  051256  004737  051276          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
10120  051262  132737  000007  051342          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
10121  051270  001372                          BNE     9$              ;;TAB STOP
10122  051272  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
10123  051274  000724                          BR      2$              ;;GET NEXT CHARACTER
10124  051276  105777  127646          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
10125  051302  100375                          BPL     $TYPEC
10126  051304  116677  000002  127640          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
10127  051312  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
10128  051320  001003                          BNE     1$              ;;BRANCH IF NO
10129  051322  105037  051342                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
10130  051326  000406                          BR      $TYPEX          ;;EXIT
10131  051330  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
10132  051336  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
10133  051340  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
10134  051342  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
10135  051344  000207          $TYPEX: RTS     PC
10136
10137                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10138
10139                                  ;;**************************************************************
10140                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
10141                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
10142                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
10143                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
10144                                  ;*REPLACED WITH SPACES.
10145                                  ;*CALL:
10146                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
10147                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
10148
10149  051346                          $TYPDS:
10150  051346  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10151  051350  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10152  051352  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10153  051354  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10154  051356  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10155  051360  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
10156  051364  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
10157  051370  100004                          BPL     1$              ;;BR IF INPUT IS POS.
10158  051372  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
10159  051374  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
10160  051402  005000          1$:     CLR     R0                      ;;ZERO THE CONSTANTS INDEX
10161  051404  012703  051562                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
10162  051410  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
10163  051414  005002          2$:     CLR     R2                      ;;CLEAR THE BCD NUMBER
```

```
10164  051416  016001  051552          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
10165  051422  160105          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
10166  051424  002402                  BLT     4$              ;;BR IF DONE
10167  051426  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
10168  051430  000774                  BR      3$
10169  051432  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
10170  051434  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
10171  051436  001002                  BNE     5$              ;;FALL THROUGH IF 0
10172  051440  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
10173  051442  100407                  BMI     7$              ;;BR IF YES
10174  051444  106316          5$:     ASLB    (SP)            ;;MSD?
10175  051446  103003                  BCC     6$              ;;BR IF NO
10176  051450  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
10177  051456  052702  000060  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
10178  051462  052702  000040  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
10179  051466  110223                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
10180  051470  005720                  TST     (R0)+           ;;JUST INCREMENTING
10181  051472  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
10182  051476  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
10183  051500  003002                  BGT     8$              ;;GO TO EXIT
10184  051502  010502                  MOV     R5,R2           ;;GET THE LSD
10185  051504  000764                  BR      6$              ;;GO CHANGE TO ASCII
10186  051506  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
10187  051510  100003                  BPL     9$              ;;BR IF NO
10188  051512  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
10189  051520  105013          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
10190  051522  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
10191  051524  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
10192  051526  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10193  051530  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10194  051532  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10195  051534  104401  051562          TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
10196  051540  016666  000002  000004  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
10197  051546  012616                  MOV     (SP)+,(SP)
10198  051550  000002                  RTI                     ;;RETURN TO USER
10199  051552  023420          $DTBL:  10000.
10200  051554  001750                  1000.
10201  051556  000144                  100.
10202  051560  000012                  10.
10203  051562  000004          $DBLK:  .BLKW   4
10204                           .SBTTL  APT COMMUNICATIONS ROUTINE
10205
10206                           ;;**********************************************************************
10207  051572  112737  000001  052036  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
10208  051600  112737  000001  052034  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
10209  051606  000403                  BR      $ATYC
10210  051610  112737  000001  052036  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
10211  051616                          $ATYC:
10212  051616  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10213  051620  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10214  051622  105737  052034          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
10215  051626  001450                  BEQ     5$              ;;IF NOT:  BR
10216  051630  122737  000001  001230  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
10217  051636  001031                  BNE     3$              ;;IF NOT:  BR
10218  051640  132737  000100  001231  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
10219  051646  001425                  BEQ     3$              ;;IF NOT:  BR
```

```
10220  051650  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
10221  051654  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10222  051662  005737  001210  1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
10223  051666  001375                  BNE     1$              ;;IF NOT:  WAIT
10224  051670  010037  001224          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
10225  051674  105720          2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
10226  051676  001376                  BNE     2$
10227  051700  163700  001224          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
10228  051704  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
10229  051706  010037  001226          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
10230  051712  012737  000004  001210  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
10231  051720  000413                  BR      5$
10232  051722  017637  000004  051746  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
10233  051730  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
10234  051736  013746  177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
10235  051742  004737  051064          JSR     PC,$TYPE        ;;CALL TYPE MACRO
10236  051746  000000          4$:     .WORD   0
10237  051750                  5$:
10238  051750  105737  052036  10$:     TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
10239  051754  001416                  BEQ     12$             ;;IF NOT:  BR
10240  051756  005737  001230          TST     $ENV            ;;RUNNING UNDER APT?
10241  051762  001413                  BEQ     12$             ;;IF NOT:  BR
10242  051764  005737  001210  11$:     TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
10243  051770  001375                  BNE     11$             ;;IF NOT:  WAIT
10244  051772  017637  000004  001212  MOV     @4(SP),$FATAL   ;;GET ERROR #
10245  052000  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10246  052006  005237  001210          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
10247  052012  105037  052036  12$:     CLRB    $FFLG           ;;CLEAR FATAL FLAG
10248  052016  105037  052035          CLRB    $LFLG           ;;CLEAR LOG FLAG
10249  052022  105037  052034          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
10250  052026  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10251  052030  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10252  052032  000207                  RTS     PC              ;;RETURN
10253  052034     000  $MFLG:   .BYTE   0               ;;MESSG. FLAG
10254  052035     000  $LFLG:   .BYTE   0               ;;LOG FLAG
10255  052036     000  $FFLG:   .BYTE   0               ;;FATAL FLAG
10256          052040          .EVEN
10257          000200  APTSIZE=200
10258          000001  APTENV=001
10259          000100  APTSPOOL=100
10260          000040  APTCSUP=040
10261                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
10262
10263                  ;;*****************************************************************
10264                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
10265                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
10266                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
10267                  ;*CALL:
10268                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
10269                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
10270                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
10271                  ;*      .BYTE   M               ;;M=1 OR 0
10272                  ;*                                      ;;1=TYPE LEADING ZEROS
10273                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
10274                  ;*
10275                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
10276                                   ;*$TYPOS OR $TYPOC
10277                                   ;*CALL:
10278                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
10279                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
10280                                   ;*
10281                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
10282                                   ;*CALL:
10283                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
10284                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
10285
10286  052040  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
10287  052044  116637  000001  052263           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
10288  052052  112637  052265                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
10289  052056  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
10290  052062  000406                           BR      $TYPON
10291  052064  112737  000001  052263  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
10292  052072  112737  000006  052265           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
10293  052100  112737  000005  052262  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
10294  052106  010346                           MOV     R3,-(SP)        ;;SAVE R3
10295  052110  010446                           MOV     R4,-(SP)        ;;SAVE R4
10296  052112  010546                           MOV     R5,-(SP)        ;;SAVE R5
10297  052114  113704  052265                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
10298  052120  005404                           NEG     R4
10299  052122  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
10300  052126  110437  052264                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
10301  052132  113704  052263                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
10302  052136  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
10303  052142  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
10304  052144  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
10305  052146  000404                           BR      3$              ;;GO DO MSB
10306  052150  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
10307  052152  006105                           ROL     R5
10308  052154  006105                           ROL     R5
10309  052156  010503                           MOV     R5,R3
10310  052160  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
10311  052162  105337  052264                   DECB    $OMODE          ;;TYPE THIS DIGIT?
10312  052166  100016                           BPL     7$              ;;BR IF NO
10313  052170  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
10314  052174  001002                           BNE     4$              ;;TEST FOR 0
10315  052176  005704                           TST     R4              ;;SUPPRESS THIS 0?
10316  052200  001403                           BEQ     5$              ;;BR IF YES
10317  052202  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
10318  052204  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
10319  052210  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
10320  052214  110337  052260                   MOVB    R3,8$           ;;SAVE FOR TYPING
10321  052220  104401  052260                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
10322  052224  105337  052262           7$:     DECB    $OCNT           ;;COUNT BY 1
10323  052230  003347                           BGT     2$              ;;BR IF MORE TO DO
10324  052232  002402                           BLT     6$              ;;BR IF DONE
10325  052234  005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
10326  052236  000744                           BR      2$              ;;GO DO THE LAST DIGIT
10327  052240  012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
10328  052242  012604                           MOV     (SP)+,R4        ;;RESTORE R4
10329  052244  012603                           MOV     (SP)+,R3        ;;RESTORE R3
10330  052246  016666  000002  000004           MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
10331  052254  012616                           MOV     (SP)+,(SP)
```

```
10332  052256  000002                  RTI                             ;;RETURN
10333  052260     000          8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
10334  052261     000                  .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
10335  052262     000          $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
10336  052263     000          $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
10337  052264  000000          $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
10338                          .SBTTL  TTY INPUT ROUTINE
10339
10340                          ;;**************************************************************************
10341                          .ENABL  LSB
10342  052266  000000          $TKCNT: .WORD   0                       ;;NUMBER OF ITEMS IN QUEUE
10343  052270  000000          $TKQIN: .WORD   0                       ;;INPUT POINTER
10344  052272  000000          $TKQOUT: .WORD  0                       ;;OUTPUT POINTER
10345  052274  000001          $TKQSRT: .BLKB  1                       ;;TTY KEYBOARD QUEUE
10346           052275          $TKQEND=.
10347           052276          .EVEN
10348
10349                          ;*TK INITIALIZE ROUTINE
10350                          ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
10351                          ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
10352                          ;
10353                          ;*CALL:
10354                          ;*      JSR     PC,$TKINT
10355                          ;*      RETURN
10356                          ;
10357  052276  005037  052266  $TKINT: CLR     $TKCNT                  ;;CLEAR COUNT OF ITEMS IN QUEUE
10358  052302  012737  052274  052270          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
10359  052310  013737  052270  052272          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
10360  052316  012737  052346  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
10361  052324  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
10362  052332  005777  126610                  TST     @$TKB           ;;CLEAR DONE FLAG
10363  052336  012777  000100  126600          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
10364  052344  000207                  RTS     PC                      ;;RETURN TO CALLER
10365
10366                          ;*TK SERVICE ROUTINE
10367                          ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
10368                          ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
10369                          ;*IT IN THE QUEUE.
10370                          ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
10371                          ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
10372                          ;
10373  052346  117746  126574  $TKSRV: MOVB    @$TKB,-(SP)             ;;PICKUP THE CHARACTER
10374  052352  042716  177600          BIC     #↑C177,(SP)             ;;STRIP THE JUNK
10375  052356  021627  000003          CMP     (SP),#3                 ;;IS IT A CONTROL C?
10376  052362  001007                  BNE     1$                      ;;BRANCH IF NO
10377  052364  104401  053474          TYPE    ,$CNTLC                 ;;TYPE A CONTROL-C (↑C)
10378  052370  004737  052276          JSR     PC,$TKINT               ;;INIT THE KEYBOARD
10379  052374  005726                  TST     (SP)+                   ;;CLEAN UP STACK
10380  052376  000137  047576          JMP     STOP                    ;;CONTROL C RESTART
10381  052402  021627  000007  1$:     CMP     (SP),#7                 ;;IS IT A CONTROL G?
10382  052406  001004                  BNE     2$                      ;;BRANCH IF NO
10383  052410  022737  000176  001140  CMP     #SWREG,SWR              ;;IS SOFT-SWR SELECTED?
10384  052416  001500                  BEQ     6$                      ;;GO TO SWR CHANGE
10385
10386  052420                  2$:
10387  052420  022737  000001  052266  CMP     #1,$TKCNT               ;;IS THE QUEUE FULL?
```

```
10388  052426  001004                        BNE     3$              ;;BRANCH IF NO
10389  052430  104401  001200                TYPE    ,$BELL          ;;RING THE TTY BELL
10390  052434  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
10391  052436  000451                        BR      5$              ;;EXIT
10392  052440  021627  000023        3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
10393  052444  001021                        BNE     32$             ;;BRANCH IF NO
10394  052446  005077  126472                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
10395  052452  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
10396  052454  105777  126464        31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
10397  052460  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
10398  052462  117746  126460                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
10399  052466  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10400  052472  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
10401  052476  001366                        BNE     31$             ;;BRANCH IF NO
10402  052500  012777  000100  126436        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
10403  052506  000002                        RTI                     ;;RETURN
10404  052510  005237  052266        32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
10405  052514  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
10406  052520  002405                        BLT     4$              ;;BRANCH IF YES
10407  052522  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
10408  052526  003002                        BGT     4$              ;;BRANCH IF YES
10409  052530  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
10410  052534  112677  177530        4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
10411  052540  005237  052270                INC     $TKQIN          ;;UPDATE THE POINTER
10412  052544  023727  052270  052275        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
10413  052552  001003                        BNE     5$              ;;BRANCH IF NO
10414  052554  012737  052274  052270        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
10415  052562  000002                5$:     RTI                     ;;RETURN
10416
10417                                  ;;*****************************************************************
10418                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
10419                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
10420                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
10421                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
10422  052564  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
10423  052572  001124                        BNE     15$             ;;EXIT IF NOT
10424  052574  105777  126344                TSTB    @$TKS           ;;IS A CHAR WAITING?
10425  052600  100121                        BPL     15$             ;;IF NOT, EXIT
10426  052602  117746  126340                MOVB    @$TKB,-(SP)     ;;YES
10427  052606  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
10428  052612  021627  000007                CMP     (SP),#7         ;;IS IT A CONTROL-G?
10429  052616  001300                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
10430                                                                 ;;AND EXIT
10431
10432                                  ;;*****************************************************************
10433                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
10434                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
10435                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
10436  052620  123727  001134  000001 6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
10437  052626  001674                        BEQ     2$              ;;BRANCH IF YES
10438  052630  005726                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
10439  052632  004737  052276                JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
10440  052636  005077  126302                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
10441  052642  112737  000001  001135        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
10442
10443  052650  104401  053506                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
```

```
10444  052654  104401  053513          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
10445  052660  013746  000176                  MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
10446  052664  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
10447  052666  104401  053524                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
10448  052672  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
10449  052674  005046                          CLR     -(SP)           ;;THE NEW SWR
10450  052676  105777  126242          7$:     TSTB    @$TKS           ;;CHAR THERE?
10451  052702  100375                          BPL     7$              ;;IF NOT TRY AGAIN
10452
10453  052704  117746  126236                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
10454  052710  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10455
10456  052714  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
10457  052720  001015                          BNE     9$              ;;BRANCH IF NOT
10458  052722  104401  053474                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
10459  052726  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
10460  052732  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
10461  052740  001003                          BNE     8$              ;;BRANCH IF NO
10462  052742  012777  000100  126174          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
10463  052750  000137  047576          8$:     JMP     STOP            ;;CONTROL-C RESTART
10464
10465
10466  052754  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
10467  052760  001005                          BNE     10$             ;;BRANCH IF NOT
10468  052762  104401  053501                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
10469  052766  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
10470  052772  000737                          BR      19$             ;;LET'S TRY IT AGAIN
10471
10472
10473  052774  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
10474  053000  001022                          BNE     16$             ;;BRANCH IF NO
10475  053002  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
10476  053006  001403                          BEQ     11$             ;;BRANCH IF YES
10477  053010  016677  000002  126122          MOV     2(SP),@SWR      ;;SAVE NEW SWR
10478  053016  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
10479  053022  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
10480  053026  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
10481  053034  001003                          BNE     15$             ;;BRANCH IF NOT
10482  053036  012777  000100  126100          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
10483  053044  000002          15$:    RTI                             ;;RETURN
10484  053046  004737  051276          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
10485  053052  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
10486  053056  002420                          BLT     18$             ;;BRANCH IF YES
10487  053060  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
10488  053064  003015                          BGT     18$             ;;BRANCH IF YES
10489  053066  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
10490  053072  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
10491  053076  001403                          BEQ     17$             ;;BRANCH IF YES
10492  053100  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
10493  053102  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
10494  053104  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
10495  053106  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
10496  053112  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
10497  053116  000667                          BR      7$              ;;GET THE NEXT ONE
10498  053120  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
10499  053124  000720                          BR      20$             ;;SIMULATE CONTROL-U
```

```
10500                                   .DSABL  LSB
10501
10502
10503                                   ;;*****************************************************************
10504                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10505                                   ;*CALL:
10506                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
10507                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10508                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
10509                                   ;
10510
10511  053126  011646                   $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
10512  053130  016666  000004  000002           MOV     4(SP),2(SP)     ;;THE PS
10513  053136  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
10514  053142  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
10515  053144  012746  053152                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
10516  053150  000002                          RTI                     ;;POP NEW PC AND PS
10517  053152                           64$:
10518  053152  005737  052266           1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
10519  053156  001775                          BEQ     1$
10520  053160  005337  052266                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
10521  053164  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
10522  053172  005237  052272                  INC     $TKQOUT         ;;UPDATE THE POINTER
10523  053176  023727  052272  052275          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
10524  053204  001003                          BNE     2$              ;;BRANCH IF NO
10525  053206  012737  052274  052272          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
10526  053214  000002                   2$:     RTI                     ;;RETURN
10527                                   ;;*****************************************************************
10528                                   ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10529                                   ;*CALL:
10530                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10531                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10532                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10533
10534  053216  010346                   $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
10535  053220  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
10536  053222  012703  053452           1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
10537  053226  022703  053474           2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
10538  053232  101456                          BLOS    4$              ;;BR IF YES
10539  053234  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10540  053236  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
10541  053240  122713  000177           10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
10542  053244  001022                          BNE     5$              ;;BR IF NO
10543  053246  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
10544  053250  001007                          BNE     6$              ;;BR IF NO
10545  053252  112737  000134  053450          MOVB    #'\,9$          ;;TYPE A BACK SLASH
10546  053260  104401  053450                  TYPE    ,9$
10547  053264  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
10548  053270  005303                   6$:     DEC     R3              ;;BACKUP BY ONE
10549  053272  020327  053452                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
10550  053276  103434                          BLO     4$              ;;BR IF YES
10551  053300  111337  053450                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
10552  053304  104401  053450                  TYPE    ,9$             ;;GO TYPE
10553  053310  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
10554  053312  005716                   5$:     TST     (SP)            ;;RUBOUT KEY SET?
10555  053314  001406                          BEQ     7$              ;;BR IF NO
```

```
10556  053316  112737  000134  053450          MOVB    #'\,9$          ;;TYPE A BACK SLASH
10557  053324  104401  053450                  TYPE    9$
10558  053330  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
10559  053332  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
10560  053336  001003                          BNE     8$              ;;BR IF NO
10561  053340  104401  053501                  TYPE    $CNTLU          ;;TYPE A CONTROL "U"
10562  053344  000726                          BR      1$              ;;GO START OVER
10563  053346  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
10564  053352  001011                          BNE     3$              ;;BRANCH IF NO
10565  053354  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
10566  053356  104401  001205                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
10567  053362  104401  053452                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
10568  053366  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
10569  053370  104401  001204          4$:     TYPE    $QUES           ;;TYPE A '?'
10570  053374  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
10571  053376  111337  053450          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
10572  053402  104401  053450                  TYPE    9$
10573  053406  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10574  053412  001305                          BNE     2$              ;;LOOP IF NOT RETURN
10575  053414  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10576  053420  104401  001206                  TYPE    $LF             ;;TYPE A LINE FEED
10577  053424  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
10578  053426  012603                          MOV     (SP)+,R3        ;;RESTORE R3
10579  053430  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10580  053432  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10581  053440  012766  053452  000004          MOV     #$TTYIN,4(SP)
10582  053446  000002                          RTI                     ;;RETURN
10583  053450     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
10584  053451     000                          .BYTE   0               ;;TERMINATOR
10585  053452  000022          $TTYIN: .BLKB   22              ;;RESERVE 22 BYTES FOR TTY INPUT
10586  053474  041536  005015     000  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
10587  053501     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
10588  053506  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
10589  053513     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
10590  053520  036440  000040
10591  053524  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
10592  053532  020075     000
10593          053536                  .EVEN
10594                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
10595
10596                                  ;;*****************************************************************
10597                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
10598                                  ;*CHANGE IT TO BINARY.
10599                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
10600                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
10601                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
10602                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
10603                                  ;*CALL:
10604                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
10605                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
10606                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
10607
10608  053536  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
10609  053540  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
10610  053546  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10611  053550  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
10612  053552  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10613  053554  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
10614  053556  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
10615  053560  010037  053664          MOV     R0,5$           ;;AND SAVE IT
10616  053564  005001                  CLR     R1              ;;CLEAR DATA WORD
10617  053566  005002                  CLR     R2
10618  053570  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
10619  053572  001420                  BEQ     3$              ;;IF ZERO GET OUT
10620  053574  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
10621  053600  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
10622  053602  122716  000067          CMPB    #'7,(SP)
10623  053606  002423                  BLT     4$
10624  053610  006301                  ASL     R1              ;;*2
10625  053612  006102                  ROL     R2
10626  053614  006301                  ASL     R1              ;;*4
10627  053616  006102                  ROL     R2
10628  053620  006301                  ASL     R1              ;;*8
10629  053622  006102                  ROL     R2
10630  053624  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
10631  053630  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
10632  053632  000756                  BR      2$              ;;LOOP
10633  053634  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
10634  053636  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
10635  053642  010237  053674          MOV     R2,$HIOCT
10636  053646  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10637  053650  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10638  053652  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10639  053654  000002                  RTI                     ;;RETURN
10640  053656  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
10641  053660  105010                  CLRB    (R0)            ;;SET A TERMINATOR
10642  053662  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
10643  053664  000000          5$:     .WORD   0
10644  053666  104401  001204          TYPE    ,$QUES          ;;"?" "CR" & "LF"
10645  053672  000730                  BR      1$              ;;TRY AGAIN
10646  053674  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
10647                          .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
10648
10649                          ;;*****************************************************************
10650                          ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
10651                          ;*UNSIGNED OCTAL ASCIZ NUMBER.
10652                          ;*CALL
10653                          ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
10654                          ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
10655                          ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
10656
10657
10658  053676  104413          $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
10659  053700  016601  000002          MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
10660  053704  012705  054015          MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
10661  053710  012704  000014          MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
10662  053714  012703  177770          MOV     #^C7,R3         ;;MASK
10663  053720  012100                  MOV     (R1)+,R0        ;;LOWER WORD
10664  053722  012101                  MOV     (R1)+,R1        ;;HIGH WORD
10665  053724  005002                  CLR     R2              ;;TERMINATOR
10666  053726  110245          1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
10667  053730  010002                  MOV     R0,R2           ;;GET THIS DIGIT
```

```
10668  053732  005304                        DEC     R4              ;;COUNT THIS CHARACTER
10669  053734  003007                        BGT     3$              ;;BR IF NOT THE LAST DIGIT
10670  053736  001405                        BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
10671  053740  005205                        INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
10672  053742  010566  000002                MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
10673  053746  104414                        RESREG                  ;;RESTORE ALL REGISTERS
10674  053750  000207                        RTS     PC              ;;RETURN TO USER
10675  053752  006203                2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
10676  053754  006001                3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
10677  053756  006000                        ROR     R0              ;;      THE NEXT OCTAL DIGIT
10678  053760  006001                        ROR     R1
10679  053762  006000                        ROR     R0
10680  053764  006001                        ROR     R1
10681  053766  006000                        ROR     R0
10682  053770  040302                        BIC     R3,R2           ;;MASK OUT ALL JUNK
10683  053772  062702  000060                ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
10684  053776  000753                        BR      1$              ;;GO PUT IT IN THE DATA TABLE
10685  054000  000016                $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
10686                                  .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
10687
10688                                  ;;*****************************************************************
10689                                  ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
10690                                  ;*UNSIGNED DECIMAL ASCIZ NUMBER.
10691                                  ;*CALL
10692                                  ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
10693                                  ;*      JSR     PC,@#$SB2D      ;;CALL
10694                                  ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
10695
10696
10697  054016  016637  000002  054046 $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
10698  054024  012746  054046                MOV     #1$,-(SP)       ;;SET POINTER
10699  054030  004737  054052                JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
10700  054034  062716  000005                ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
10701  054040  012666  000002                MOV     (SP)+,2(SP)     ;;PICKUP POINTER
10702  054044  000207                        RTS     PC              ;;RETURN
10703  054046  000000  000000        1$:     .WORD   0,0
10704                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
10705
10706                                  ;;*****************************************************************
10707                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
10708                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
10709                                  ;*POSITIVE.
10710                                  ;*CALL
10711                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
10712                                  ;*      JSR     PC,@#$DB2D
10713                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
10714                                                                  ;;IS ON THE STACK
10715
10716
10717  054052  104413                $DB2D:  SAVREG                  ;;SAVE REGISTERS
10718  054054  016602  000002                MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
10719  054060  012700  054232                MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
10720  054064  010066  000002                MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
10721  054070  012201                        MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
10722  054072  012202                        MOV     (R2)+,R2
10723  054074  012737  000012  054150        MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
```

```
10724  054102  012704  054162          MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
10725  054106  012705  054164          MOV     #$TNPWR+2,R5
10726  054112  005003              1$: CLR     R3              ;;CLEAR PARTIAL
10727  054114  161401              2$: SUB     (R4),R1         ;;SUBTRACT TEN POWER
10728  054116  005602                  SBC     R2
10729  054120  161502                  SUB     (R5),R2
10730  054122  002402                  BLT     3$              ;;BR IF TEN POWER TO LARGE
10731  054124  005203                  INC     R3              ;;ADD 1 TO PARTIAL
10732  054126  000772                  BR      2$              ;;LOOP
10733  054130  062401              3$: ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
10734  054132  005502                  ADC     R2
10735  054134  062402                  ADD     (R4)+,R2
10736  054136  022525                  CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
10737  054140  052703  000060          BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
10738  054144  110320                  MOVB    R3,(R0)+        ;;SAVE IT
10739  054146  005327                  DEC     (PC)+           ;;DONE?
10740  054150  000000              4$: .WORD   0
10741  054152  001357                  BNE     1$              ;;BR IF NO
10742  054154  105020                  CLRB    (R0)+           ;;TERMINATOR
10743  054156  104414                  RESREG                  ;;RESTORE REGISTERS
10744  054160  000207                  RTS     PC              ;;RETURN
10745  054162  145000          $TNPWR: 145000                  ;;1.0E09
10746  054164  035632                  35632
10747  054166  160400                  160400                  ;;1.0E08
10748  054170  002765                  2765
10749  054172  113200                  113200                  ;;1.0E07
10750  054174  000230                  230
10751  054176  041100                  041100                  ;;1.0E06
10752  054200  000017                  17
10753  054202  103240                  103240                  ;;1.0E05
10754  054204  000001                  1
10755  054206  023420                  23420                   ;;1.0E04
10756  054210  000000                  0
10757  054212  001750                  1750                    ;;1.0E03
10758  054214  000000                  0
10759  054216  000144                  144                     ;;1.0E02
10760  054220  000000                  0
10761  054222  000012                  12                      ;;1.0E0i
10762  054224  000000                  0
10763  054226  000001                  1                       ;;1.0E00
10764  054230  000000                  0
10765  054232  000014          $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
10766                          .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
10767
10768                          ;;*****************************************************************
10769                          ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
10770                          ;*LEADING NUMBERS.
10771                          ;*CALL
10772                          ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
10773                          ;*      JSR     PC,@#$SUPRS
10774
10775
10776  054246  010046          $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
10777  054250  016600  000004          MOV     4(SP),R0        ;;PICKUP THE POINTER
10778  054254  105710              1$: TSTB    (R0)            ;;TERMINATEOR?
10779  054256  001403                  BEQ     2$              ;;BR IF YES
```

```
10780  054260  122720  000060          CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
10781  054264  001773                  BEQ     1$              ;;BR IF YES
10782  054266  005300          2$:     DEC     R0              ;;BACKUP BY "1"
10783  054270  010037  054276          MOV     R0,3$           ;;SAVE FOR TYPING
10784  054274  104401                  TYPE                    ;;GO TYPE
10785  054276  000000          3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
10786  054300  012600                  MOV     (SP)+,R0        ;;RESTORE R0
10787  054302  012616                  MOV     (SP)+,(SP)      ;;RESTORE THE STACK
10788  054304  000207                  RTS     PC              ;;RETURN
10789                          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
10790
10791                          ;;*****************************************************************
10792                          ;*SAVE R0-R5
10793                          ;*CALL:
10794                          ;*      SAVREG
10795                          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
10796                          ;*
10797                          ;*TOP---(+16)
10798                          ;* +2---(+18)
10799                          ;* +4---R5
10800                          ;* +6---R4
10801                          ;* +8---R3
10802                          ;*+10---R2
10803                          ;*+12---R1
10804                          ;*+14---R0
10805
10806  054306                  $SAVREG:
10807  054306  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10808  054310  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10809  054312  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10810  054314  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10811  054316  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10812  054320  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10813  054322  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
10814  054326  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
10815  054332  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
10816  054336  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
10817  054342  000002                  RTI
10818
10819                          ;*RESTORE R0-R5
10820                          ;*CALL:
10821                          ;*      RESREG
10822  054344                  $RESREG:
10823  054344  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
10824  054350  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
10825  054354  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
10826  054360  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
10827  054364  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
10828  054366  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
10829  054370  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
10830  054372  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10831  054374  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10832  054376  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10833  054400  000002                  RTI
10834                          .SBTTL  TRAP DECODER
10835
```

```
10836                                   ;;*****************************************************************
10837                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10838                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10839                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10840                                   ;*GO TO THAT ROUTINE.
10841
10842  054402  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
10843  054404  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
10844  054410  005740                          TST     -(R0)           ;;BACKUP BY 2
10845  054412  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
10846  054414  006300                          ASL     R0              ;;POSITION FOR INDEXING
10847  054416  016000  054436                 MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
10848  054422  000200                          RTS     R0              ;;GO TO ROUTINE
10849
10850
10851                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
10852
10853  054424  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
10854  054426  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
10855  054434  000002                          RTI                     ;;RESTORE THE PSW
10856
10857                                   .SBTTL  TRAP TABLE
10858
10859                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10860                                   ;*BY THE "TRAP" INSTRUCTION.
10861
10862                                   ;       ROUTINE
10863                                   ;       -------
10864  054436  054424                   $TRPAD: .WORD   $TRAP2
10865  054440  051064                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
10866  054442  052064                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10867  054444  052040                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
10868  054446  052100                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
10869  054450  051346                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
10870
10871  054452  052654                           $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
10872
10873  054454  052564                           $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
10874  054456  053126                           $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
10875  054460  053216                           $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
10876  054462  053536                           $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
10877  054464  054306                           $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
10878  054466  054344                           $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
10879  054470  047474                           SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
10880
```

```
10881
10882                                   .SBTTL  SERVICE MSGS
10883                                   .EVEN
10884  054472  000062                   BSE22S: .BLKW   50.              ;22 SECTOR SOFTWARE INFO
10885
10886
10887  054636  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
10888  054644  051525  051040  030113
10889  054652  020066  051104  053111
10890  054660  020105  044504  043501
10891  054666  047516  052123  041511
10892  054674  005015  040515  047111          .ASCII  <CR><LF>/MAINDEC-11-DZR6H-D-PB/<CR><LF>
10893  054702  042504  026503  030461
10894  054710  042055  051132  044066
10895  054716  042055  050055  006502
10896  054724     012
10897  054725     015  004412  025052          .ASCII  <CR><LF>/       *** CAUTION ***/<CR><LF>
10898  054732  020052  040503  052125
10899  054740  047511  020116  025052
10900  054746  006452     012
10901  054751     015  052012  044510          .ASCII  <CR><LF>/THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING CONTROL-C/
10902  054756  020123  051120  043517
10903  054764  040522  020115  044123
10904  054772  052517  042114  041040
10905  055000  020105  040510  052114
10906  055006  042105  047440  046116
10907  055014  020131  054502  052040
10908  055022  050131  047111  020107
10909  055030  047503  052116  047522
10910  055036  026514     103
10911  055041     015  047412  044124          .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND,OR THE DRIVE/
10912  055046  051105  044527  042523
10913  055054  020054  040503  052122
10914  055062  044522  043504  020105
10915  055070  047506  046522  052101
10916  055076  044524  043516  040440
10917  055104  042116  047454  020122
10918  055112  044124  020105  051104
10919  055120  053111     105
10920  055123     015  046412  054501          .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/<CR><LF>
10921  055130  041040  020105  042514
10922  055136  052106  044440  020116
10923  055144  047101  052440  042116
10924  055152  052105  051105  044515
10925  055160  042516  020104  052123
10926  055166  052101  006505     012
10927  055173     015  044412  044516          .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
10928  055200  044524  046101  054514
10929  055206  020054  051104  053111
10930  055214  051505  052040  020117
10931  055222  042502  052040  051505
10932  055230  042524  020104  044123
10933  055236  052517  042114  044040
10934  055244  053101  035105  005015
10935  055252  005015  027101  020040          .ASCII  <CR><LF>/A.  HEADS MANUALLY LOADED/
10936  055260  042510  042101  020123
```

```
10937  055266  040515  052516  046101
10938  055274  054514  046040  040517
10939  055302  042504     104
10940  055305     015  041012  020056          .ASCII  <CR><LF>/B.  CORRECT PORT SELECTED/
10941  055312  041440  051117  042522
10942  055320  052103  050040  051117
10943  055326  020124  042523  042514
10944  055334  052103  042105
10945  055340  005015  027103  020040          .ASCII  <CR><LF>/C.  WRITE LOCK DISABLED/
10946  055346  051127  052111  020105
10947  055354  047514  045503  042040
10948  055362  051511  041101  042514
10949  055370     104
10950  055371     015  042012  020056          .ASCII  <CR><LF>/D.  DRIVE READY INDICATOR ON/<CR><LF>
10951  055376  042040  044522  042526
10952  055404  051040  040505  054504
10953  055412  044440  042116  041511
10954  055420  052101  051117  047440
10955  055426  006516     012
10956  055431     015  042012  044522          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE BOTH/
10957  055436  042526  020123  047516
10958  055444  020124  047524  041040
10959  055452  020105  042524  052123
10960  055460  042105  046440  051525
10961  055466  020124  040510  042526
10962  055474  041040  052117     110
10963  055501     015  050012  051117          .ASCIZ  <CR><LF>/PORTS DESELECTED/<CR><LF>
10964  055506  051524  042040  051505
10965  055514  046105  041505  042524
10966  055522  006504  000012
10967  055526  005015  052520  020124  MSG2:   .ASCIZ  <CR><LF>/PUT SCRATCH PACK IN DRIVE 0/
10968  055534  041523  040522  041524
10969  055542  020110  040520  045503
10970  055550  044440  020116  051104
10971  055556  053111  020105  000060
10972  055564  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
10973  055572  024105  024523  052040
10974  055600  020117  042502  052040
10975  055606  051505  042524  035104
10976  055614  000040
10977  055616  005015  052502  020123  MSG4:   .ASCIZ  <CR><LF>/BUS ADDR (177440): /
10978  055624  042101  051104  024040
10979  055632  033461  032067  030064
10980  055640  035051  000040
10981  055644  005015  047503  052116  MSG5:   .ASCIZ  <CR><LF>/CONTR ADDR (210): /
10982  055652  020122  042101  051104
10983  055660  024040  030462  024460
10984  055666  020072     000
10985  055671     015  044412  052116  MSG6:   .ASCIZ  <CR><LF>/INTR AT PC=/
10986  055676  020122  052101  050040
10987  055704  036503     000
10988  055707     015  042012  044522  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
10989  055714  042526  030040  053440
10990  055722  046111  020114  047516
10991  055730  020124  042502  052040
10992  055736  051505  042524  000104
```

```
 10993  055744  005015  042524  052123  MSG8:   .ASCIZ  <CR><LF>/TEST 16 TAKES 2 TO 4 MIN./<CR><LF>
 10994  055752  030440  020066  040524
 10995  055760  042513  020123  020062
 10996  055766  047524  032040  046440
 10997  055774  047111  006456  000012
 10998  056002  005015  054502  040520  MSG9:   .ASCIZ  <CR><LF>/BYPASSING TEST 16/<CR><LF>
 10999  056010  051523  047111  020107
 11000  056016  042524  052123  030440
 11001  056024  006466  000012
 11002  056030  005015  053412  046111  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVE(S):/
 11003  056036  020114  042524  052123
 11004  056044  042040  044522  042526
 11005  056052  051450  035051     000
 11006  056057     015  005012  047520  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
 11007  056064  042527  020122  050125
 11008  056072  051040  051505  040524
 11009  056100  052122  052040  020117
 11010  056106  042524  052123  030440
 11011  056114  005015     000
 11012  056117     015  050012  041501  MSG12:  .ASCIZ  <CR><LF>/PACK BEING FORMATTED/<CR><LF>
 11013  056124  020113  042502  047111
 11014  056132  020107  047506  046522
 11015  056140  052101  042524  006504
 11016  056146  000012
 11017  056150  005015  047516  046040  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS/
 11018  056156  047440  020122  020120
 11019  056164  046103  041517  051513
 11020  056172  005015  046101  020114          .ASCIZ  <CR><LF>/ALL TIMING TESTS BYPASSED/
 11021  056200  044524  044515  043516
 11022  056206  052040  051505  051524
 11023  056214  041040  050131  051501
 11024  056222  042523  000104
 11025  056226  005015  054502  040520  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
 11026  056234  051523  047111  020107
 11027  056242  051104  053111  020105
 11028  056250     000
 11029  056251     015  005012  051104  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
 11030  056256  053111  020105     000
 11031  056263     015  042012  044522  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL #/
 11032  056270  042526  051440  051105
 11033  056276  040511  020114  000043
 11034  056304  005015  040503  052122  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO./
 11035  056312  044522  043504  020105
 11036  056320  042523  044522  046101
 11037  056326  047040  027117     000
 11038  056333     015  005012  041101  MSG18:  .ASCIZ  <CR><LF><LF>/ABORTING BAL OF TESTS/<CR><LF><LF>
 11039  056340  051117  044524  043516
 11040  056346  041040  046101  047440
 11041  056354  020106  042524  052123
 11042  056362  006523  005012     000
 11043  056367     015  005012  046101  MSG19:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
 11044  056374  020114  051104  053111
 11045  056402  051505  052040  051505
 11046  056410  042524  006504  005012
 11047  056416     000
 11048  056417     015  046412  042117  MSG20:  .ASCII  <CR><LF>/MODIFIED VERSION OF FORMAT PACK TEST FOR MODULE TESTING/
```

```
11049  056424  043111  042511  020104
11050  056432  042526  051522  047511
11051  056440  020116  043117  043040
11052  056446  051117  040515  020124
11053  056454  040520  045503  052040
11054  056462  051505  020124  047506
11055  056470  020122  047515  052504
11056  056476  042514  052040  051505
11057  056504  044524  043516
11058  056510  005015  047524  051040          .ASCIZ  <CR><LF>/TO RESTORE HEADERS ON CYL 0 & 1, ALL TRACKS/
11059  056516  051505  047524  042522
11060  056524  044040  040505  042504
11061  056532  051522  047440  020116
11062  056540  054503  020114  020060
11063  056546  020046  026061  040440
11064  056554  046114  052040  040522
11065  056562  045503  000123
11066  056566  005015  054502  040520  MSG21:  .ASCIZ  <CR><LF>/BYPASSING TESTS 36,40,41 FOR MODULE TESTING/<CR><LF>
11067  056574  051523  047111  020107
11068  056602  042524  052123  020123
11069  056610  033063  032054  026060
11070  056616  030464  043040  051117
11071  056624  046440  042117  046125
11072  056632  020105  042524  052123
11073  056640  047111  006507  000012
11074  056646  005015  047506  046522  MSG22:  .ASCIZ  <CR><LF>/FORMATTING FINISHED/<CR><LF>
11075  056654  052101  044524  043516
11076  056662  043040  047111  051511
11077  056670  042510  006504  000012
11078
11079  056676  005015  043520  020115  MSG74:  .ASCIZ  <CR><LF>/PGM ABORT PENDING.../
11080  056704  041101  051117  020124
11081  056712  042520  042116  047111
11082  056720  027107  027056     000
11083  056725     015  044012  046101  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING.../
11084  056732  020124  042520  042116
11085  056740  047111  027107  027056
11086  056746     000
11087  056747     015  050012  046507  MSG76:  .ASCIZ  <CR><LF>/PGM ABORTED/
11088  056754  040440  047502  052122
11089  056762  042105     000
11090  056765     015  041412  052520  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
11091  056772  044040  046101  042524
11092  057000  000104
11093
11094
11095                                  .SBTTL  ERR MSGS
11096
11097  057002  005015  051105  026122  EM1:    .ASCIZ  <CR><LF>/ERR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
11098  057010  047440  046116  020131
11099  057016  020060  044124  052522
11100  057024  033440  040440  046114
11101  057032  053517  042105  020054
11102  057040  051124  020131  043501
11103  057046  044501  006516  000012
11104  057054  042523  042514  052103  EM2:    .ASCIZ  /SELECTED DRIVE # IN RKCS2 CANNOT BE READ BACK IN RKMR2/
```

```
11105  057062  042105  042040  044522
11106  057070  042526  021440  044440
11107  057076  020116  045522  051503
11108  057104  020062  040503  047116
11109  057112  052117  041040  020105
11110  057120  042522  042101  041040
11111  057126  041501  020113  047111
11112  057134  051040  046513  031122
11113  057142     000
11114  057143     015  040412  047502  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXP TIME OUT AT PC=/
11115  057150  052122  052040  051505
11116  057156  051524  027056  052456
11117  057164  042516  050130  052040
11118  057172  046511  020105  052517
11119  057200  020124  052101  050040
11120  057206  036503     000
11121  057211     106  052101  046101  EM4:    .ASCII  /FATAL ERROR/<CR><LF>
11122  057216  042440  051122  051117
11123  057224  005015
11124  057226  041101  051117  044524          .ASCII  /ABORTING BALANCE OF TESTS/<CR><LF>
11125  057234  043516  041040  046101
11126  057242  047101  042503  047440
11127  057250  020106  042524  052123
11128  057256  006523     012
11129  057261     110  040505  020104          .ASCIZ  /HEAD POSITION CANNOT BE DETERMINED/
11130  057266  047520  044523  044524
11131  057274  047117  041440  047101
11132  057302  047516  020124  042502
11133  057310  042040  052105  051105
11134  057316  044515  042516  000104
11135  057324  042115  020123  042523  EM5:    .ASCIZ  /MDS SET IN RKCS2/
11136  057332  020124  047111  051040
11137  057340  041513  031123     000
11138  057345     125  042506  051440  EM6:    .ASCIZ  /UFE SET IN RKCS2/
11139  057352  052105  044440  020116
11140  057360  045522  051503  000062
11141  057366  051104  020101  047111  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
11142  057374  051040  042113  020123
11143  057402  020046  042516  020104
11144  057410  047111  051040  041513
11145  057416  031123  051040  051505
11146  057424  052105  020073  051127
11147  057432  047117  020107  047520
11148  057440  052122  051440  046105
11149  057446  041505  042524  037504
11150  057454     000
11151  057455     104  044522  042526  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT TYPED BY OPERATOR/
11152  057462  050040  042522  042523
11153  057470  052116  041040  052125
11154  057476  047040  052117  052040
11155  057504  050131  042105  041040
11156  057512  020131  050117  051105
11157  057520  052101  051117     000
11158  057525     104  044522  042526  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT TYPED BY OPERATOR/
11159  057532  047040  052117  050040
11160  057540  042522  042523  052116
```

```
11161  057546  041040  052125  052040
11162  057554  050131  042105  041040
11163  057562  020131  050117  051105
11164  057570  052101  051117     000
11165  057575     101  047502  052122  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REF CONTR REG/
11166  057602  052040  051505  051524
11167  057610  027056  041456  047101
11168  057616  047516  020124  042522
11169  057624  020106  047503  052116
11170  057632  020122  042522  000107
11171  057640  051104  020101  047111  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
11172  057646  051040  042113  020123
11173  057654  020046  042516  020104
11174  057662  047111  051040  041513
11175  057670  031123  041040  052117
11176  057676  020110  042523  000124
11177  057704  047503  052116  020122  EM12:   .ASCIZ  /CONTR NOT READY IN RKCS1/
11178  057712  047516  020124  042522
11179  057720  042101  020131  047111
11180  057726  051040  041513  030523
11181  057734     000
11182  057735     116  020117  052101  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
11183  057742  047124  044440  020116
11184  057750  045522  051501  043117
11185  057756     000
11186  057757     127  047522  043516  EM14:   .ASCIZ  /WRONG ATTN IN RKASOF/
11187  057764  040440  052124  020116
11188  057772  047111  051040  040513
11189  060000  047523  000106
11190  060004  051104  054504  047040  EM15:   .ASCIZ  /DRDY NOT CLEARED IN RKMR2/
11191  060012  052117  041440  042514
11192  060020  051101  042105  044440
11193  060026  020116  045522  051115
11194  060034  000062
11195  060036  051504  020103  047516  EM16:   .ASCIZ  /DSC NOT SET IN RKMR2/
11196  060044  020124  042523  020124
11197  060052  047111  051040  046513
11198  060060  031122     000
11199  060063     115  043523  040440  EM17:   .ASCIZ  /MSG A0 ERROR/
11200  060070  020060  051105  047522
11201  060076  000122
11202  060100  051515  020107  030102  EM18:   .ASCIZ  /MSG B0 ERROR/
11203  060106  042440  051122  051117
11204  060114     000
11205  060115     115  043523  040440  EM19:   .ASCIZ  /MSG A1 ERROR/
11206  060122  020061  051105  047522
11207  060130  000122
11208  060132  051515  020107  030502  EM20:   .ASCIZ  /MSG B1 ERROR/
11209  060140  042440  051122  051117
11210  060146     000
11211  060147     103  051105  020122  EM21:   .ASCIZ  /CERR SET IN RKCS1/
11212  060154  042523  020124  047111
11213  060162  051040  041513  030523
11214  060170     000
11215  060171     122  051514  044440  EM22:   .ASCIZ  /RLS IN RKCS2 SET CERR IN RKCS1/
11216  060176  020116  045522  051503
```

```
11217  060204  020062  042523  020124
11218  060212  042503  051122  044440
11219  060220  020116  045522  051503
11220  060226  000061
11221  060230  043125  020105  047111  EM23:   .ASCIZ  /UFE IN RKCS2 SET (SACK) AFTER RLS IN RKCS2 SENT/
11222  060236  051040  041513  031123
11223  060244  051440  052105  024040
11224  060252  040523  045503  020051
11225  060260  043101  042524  020122
11226  060266  046122  020123  047111
11227  060274  051040  041513  031123
11228  060302  051440  047105  000124
11229  060310  053126  047040  052117  EM24:   .ASCIZ  /VV NOT SET IN RKMR2/
11230  060316  051440  052105  044440
11231  060324  020116  045522  051115
11232  060332  000062
11233  060334  051104  020126  054524  EM25:   .ASCIZ  /DRV TYPE SET IN RKMR2/
11234  060342  042520  051440  052105
11235  060350  044440  020116  045522
11236  060356  051115  000062
11237  060362  042104  020124  042523  EM26:   .ASCIZ  /DDT SET IN RKDS/
11238  060370  020124  047111  051040
11239  060376  042113  000123
11240  060402  052104  042531  051440  EM27:   .ASCIZ  /DTYE SET IN RKER/
11241  060410  052105  044440  020116
11242  060416  045522  051105     000
11243  060423     104  054524  020105  EM28:   .ASCIZ  /DTYE NOT SET IN RKER/
11244  060430  047516  020124  042523
11245  060436  020124  047111  051040
11246  060444  042513  000122
11247  060450  052104  042531  044440  EM29:   .ASCIZ  /DTYE IN RKER DID NOT SET CERR IN RKCS1/
11248  060456  020116  045522  051105
11249  060464  042040  042111  047040
11250  060472  052117  051440  052105
11251  060500  041440  051105  020122
11252  060506  047111  051040  041513
11253  060514  030523     000
11254  060517     103  042055  050040  EM30:   .ASCIZ  /C-D PAR ERR SET IN RKMR3/
11255  060524  051101  042440  051122
11256  060532  051440  052105  044440
11257  060540  020116  045522  051115
11258  060546  000063
11259  060550  026504  020103  040520  EM31:   .ASCIZ  /D-C PAR SET IN RKCS1/
11260  060556  020122  042523  020124
11261  060564  047111  051040  041513
11262  060572  030523     000
11263  060575     106  052114  047040  EM32:   .ASCIZ  /FLT NOT SET IN RKMR3/
11264  060602  052117  051440  052105
11265  060610  044440  020116  045522
11266  060616  051115  000063
11267  060622  026503  020104  040520  EM33:   .ASCIZ  /C-D PAR ERR NOT SET IN RKMR3/
11268  060630  020122  051105  020122
11269  060636  047516  020124  042523
11270  060644  020124  047111  051040
11271  060652  046513  031522     000
11272  060657     104  041455  050040  EM34:   .ASCIZ  /D-C PAR NOT SET IN RKCS1/
```

```
11273  060664  051101  047040  052117
11274  060672  051440  052105  044440
11275  060700  020116  045522  051503
11276  060706  000061
11277  060710  026504  020103  040520  EM35:   .ASCIZ  /D-C PAR IN RKCS1 DID NOT SET CERR IN RKCS1/
11278  060716  020122  047111  051040
11279  060724  041513  030523  042040
11280  060732  042111  047040  052117
11281  060740  051440  052105  041440
11282  060746  051105  020122  047111
11283  060754  051040  041513  030523
11284  060762     000
11285  060763     103  046131  040440  EM36:   .ASCIZ  /CYL ADDR IN B2 NOT SAME AS RKDC/
11286  060770  042104  020122  047111
11287  060776  041040  020062  047516
11288  061004  020124  040523  042515
11289  061012  040440  020123  045522
11290  061020  041504     000
11291  061023     103  046131  042040  EM37:   .ASCIZ  /CYL DIFF IN A2 NOT SAME AS RKDC/
11292  061030  043111  020106  047111
11293  061036  040440  020062  047516
11294  061044  020124  040523  042515
11295  061052  040440  020123  045522
11296  061060  041504     000
11297  061063     103  046131  042040  EM38:   .ASCIZ  /CYL DIFF IN RKMR2 NOT SAME AS 'CYL DIFF'/
11298  061070  043111  020106  047111
11299  061076  051040  046513  031122
11300  061104  047040  052117  051440
11301  061112  046501  020105  051501
11302  061120  023440  054503  020114
11303  061126  044504  043106  000047
11304  061134  054503  020114  044504  EM39:   .ASCIZ  /CYL DIFF & OFST IN A2 NOT =0/
11305  061142  043106  023040  047440
11306  061150  051506  020124  047111
11307  061156  040440  020062  047516
11308  061164  020124  030075     000
11309  061171     103  046131  040440  EM40:   .ASCIZ  /CYL ADDR IN B2 NOT =0/
11310  061176  042104  020122  047111
11311  061204  041040  020062  047516
11312  061212  020124  030075     000
11313  061217     103  046131  040440  EM41:   .ASCIZ  /CYL ADDR IN B2 DID NOT REMAIN =0/
11314  061224  042104  020122  047111
11315  061232  041040  020062  044504
11316  061240  020104  047516  020124
11317  061246  042522  040515  047111
11318  061254  036440  000060
11319  061260  042510  042101  040440  EM43:   .ASCIZ  /HEAD ADDR IN B3 NOT =0/
11320  061266  042104  020122  047111
11321  061274  041040  020063  047516
11322  061302  020124  030075     000
11323  061307     110  040505  020104  EM44:   .ASCIZ  /HEAD DECODE IN B3 INCORRECT/
11324  061314  042504  047503  042504
11325  061322  044440  020116  031502
11326  061330  044440  041516  051117
11327  061336  042522  052103     000
11328  061343     104  044522  042526  EM45:   .ASCII  /DRIVE READY IN RKMR2 NOT SET BY 1 SEC FROM FWD/
```

```
11329  061350  051040  040505  054504
11330  061356  044440  020116  045522
11331  061364  051115  020062  047516
11332  061372  020124  042523  020124
11333  061400  054502  030440  051440
11334  061406  041505  043040  047522
11335  061414  020115  053506     104
11336  061421     015  044412  020116          .ASCIZ  <CR><LF>/IN RTZ PORTION OF START SPIN CMD/
11337  061426  052122  020132  047520
11338  061434  052122  047511  020116
11339  061442  043117  051440  040524
11340  061450  052122  051440  044520
11341  061456  020116  046503  000104
11342  061464  051515  020107  031101  EM46:   .ASCIZ  /MSG A2 ERR/
11343  061472  042440  051122     000
11344  061477     115  043523  041040  EM47:   .ASCIZ  /MSG B2 ERR/
11345  061504  020062  051105  000122
11346  061512  051515  020107  031502  EM48:   .ASCIZ  /MSG B3 ERR/
11347  061520  042440  051122     000
11348  061525     106  042127  047040  EM49:   .ASCIZ  /FWD NOT SET IN RKMR2 IN RTZ PORTION OF START SPIN CMD/
11349  061532  052117  051440  052105
11350  061540  044440  020116  045522
11351  061546  051115  020062  047111
11352  061554  051040  055124  050040
11353  061562  051117  044524  047117
11354  061570  047440  020106  052123
11355  061576  051101  020124  050123
11356  061604  047111  041440  042115
11357  061612     000
11358  061613     106  042127  047040  EM50:   .ASCIZ  /FWD NOT SET IN RKMR2 FROM START SPIN CMD/
11359  061620  052117  051440  052105
11360  061626  044440  020116  045522
11361  061634  051115  020062  051106
11362  061642  046517  051440  040524
11363  061650  052122  051440  044520
11364  061656  020116  046503  000104
11365  061664  053506  020104  047516  EM51:   .ASCIZ  /FWD NOT CLEARED IN RKMR2 BY 5 SEC OF MOTION FROM START SPIN CMD/
11366  061672  020124  046103  040505
11367  061700  042522  020104  047111
11368  061706  051040  046513  031122
11369  061714  041040  020131  020065
11370  061722  042523  020103  043117
11371  061730  046440  052117  047511
11372  061736  020116  051106  046517
11373  061744  051440  040524  052122
11374  061752  051440  044520  020116
11375  061760  046503  000104
11376  061764  030062  051440  041505  EM52:   .ASCIZ  /20 SEC FORMAT NOT SET IN RKMR2/
11377  061772  043040  051117  040515
11378  062000  020124  047516  020124
11379  062006  042523  020124  047111
11380  062014  051040  046513  031122
11381  062022     000
11382  062023     123  041505  030040  EM53:   .ASCIZ  /SEC 0 NOT FOUND BY 50 MS/
11383  062030  047040  052117  043040
11384  062036  052517  042116  041040
```

```
11385  062044  020131  030065  046440
11386  062052  000123
11387  062054  044504  043106  051440  EM54:   .ASCIZ  /DIFF SEC NOT FOUND BY 3 MS/
11388  062062  041505  047040  052117
11389  062070  043040  052517  042116
11390  062076  041040  020131  020063
11391  062104  051515     000
11392  062107     101  052124  020116  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
11393  062114  047516  020124  046103
11394  062122  040505  042522  020104
11395  062130  047111  051040  040513
11396  062136  047523  000106
11397  062142  047125  054105  020120  EM56:   .ASCIZ  /UNEXP MEM PAR TRAP/
11398  062150  042515  020115  040520
11399  062156  020122  051124  050101
11400  062164     000
11401  062165     127  042503  040040  EM57:   .ASCIZ  /WCE @ CYL 411, TRK 2, SEC 2/
11402  062172  041440  046131  032040
11403  062200  030461  020054  051124
11404  062206  020113  026062  051440
11405  062214  041505  031040     000
11406  062221     015  051412  042520  EM58:   .ASCIZ  <CR><LF>/SPEED OK IN RKMR2 NOT =0 BY TIMEOUT/
11407  062226  042105  047440  020113
11408  062234  047111  051040  046513
11409  062242  031122  047040  052117
11410  062250  036440  020060  054502
11411  062256  052040  046511  047505
11412  062264  052125     000
11413  062267     114  046511  042040  EM59:   .ASCIZ  /LIM DET NOT SET IN RKMR3/
11414  062274  052105  047040  052117
11415  062302  051440  052105  044440
11416  062310  020116  045522  051115
11417  062316  000063
11418  062320  042510  042101  020123  EM60:   .ASCIZ  /HEADS HOME NOT SET IN RKMR2/
11419  062326  047510  042515  047040
11420  062334  052117  051440  052105
11421  062342  044440  020116  045522
11422  062350  051115  000062
11423  062354  047514  042101  044040  EM61:   .ASCIZ  /LOAD HEADS NOT SET IN RKMR2/
11424  062362  040505  051504  047040
11425  062370  052117  051440  052105
11426  062376  044440  020116  045522
11427  062404  051115  000062
11428  062410  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
11429  062416  020124  047111  051040
11430  062424  041513  031123     000
11431  062431     115  043523  041040  EM64:   .ASCIZ  /MSG B3 HEAD REG NOT =0/
11432  062436  020063  042510  042101
11433  062444  051040  043505  047040
11434  062452  052117  036440  000060
11435  062460  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERR/
11436  062466  040505  042504  020122
11437  062474  051105  000122
11438  062500  054503  020114  042101  EM66:   .ASCIZ  /CYL ADDR IN RKMR3 INCORRECT/
11439  062506  051104  044440  020116
11440  062514  045522  051115  020063
```

```
11441  062522  047111  047503  051122
11442  062530  041505  000124
11443  062534  042522  042101  047111  EM67:   .ASCIZ  /READING CYL 0 HEADERS ON CYL 1/
11444  062542  020107  054503  020114
11445  062550  020060  042510  042101
11446  062556  051105  020123  047117
11447  062564  041440  046131  030440
11448  062572     000
11449  062573     122  040505  044504  EM68:   .ASCIZ  /READING CYL 1 HEADERS ON CYL 0/
11450  062600  043516  041440  046131
11451  062606  030440  044040  040505
11452  062614  042504  051522  047440
11453  062622  020116  054503  020114
11454  062630  000060
11455  062632  046101  043511  020116  EM69:   .ASCIZ  /ALIGN CART USED/
11456  062640  040503  052122  052440
11457  062646  042523  000104
11458  062652  047125  054105  020120  EM70:   .ASCIZ  /UNEXP ATTN/
11459  062660  052101  047124     000
11460  062665     104  041523  051440  EM71:   .ASCIZ  /DSC SET IN RKMR2/
11461  062672  052105  044440  020116
11462  062700  045522  051115  000062
11463  062706  047506  046522  052101  EM72:   .ASCIZ  /FORMAT TEST BYPASSED/
11464  062714  052040  051505  020124
11465  062722  054502  040520  051523
11466  062730  042105     000
11467  062733     103  047524  051440  EM73:   .ASCIZ  /CTO SET IN RKCS1/
11468  062740  052105  044440  020116
11469  062746  045522  051503  000061
11470  062754  052122  020132  047516  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
11471  062762  020124  042523  020124
11472  062770  047111  051040  046513
11473  062776  031122     000
11474  063001     111  040504  020105  EM75:   .ASCIZ  /IDAE NOT SET IN RKMR3/
11475  063006  047516  020124  042523
11476  063014  020124  047111  051040
11477  063022  046513  031522     000
11478  063027     120  050111  051440  EM76:   .ASCIZ  /PIP SET IN RKMR2/
11479  063034  052105  044440  020116
11480  063042  045522  051115  000062
11481  063050  040506  046125  020124  EM77:   .ASCIZ  /FAULT NOT =0 IN RKMR3/
11482  063056  047516  020124  030075
11483  063064  044440  020116  045522
11484  063072  051115  000063
11485  063076  054503  020114  044504  EM78:   .ASCIZ  /CYL DIFF IN RKMR2 DID NOT REMAIN = 1 IN SEEK TO SELF/
11486  063104  043106  044440  020116
11487  063112  045522  051115  020062
11488  063120  044504  020104  047516
11489  063126  020124  042522  040515
11490  063134  047111  036440  030440
11491  063142  044440  020116  042523
11492  063150  045505  052040  020117
11493  063156  042523  043114     000
11494  063163     116  042105  051440  EM79:   .ASCIZ  /NED SET IN RKCS2/
11495  063170  052105  044440  020116
11496  063176  045522  051503  000062
```

```
11497  063204  047125  047514  042101  EM80:   .ASCIZ  /UNLOAD NOT SET IN RKMR2/
11498  063212  047040  052117  051440
11499  063220  052105  044440  020116
11500  063226  045522  051115  000062
11501  063234  050123  047111  047040  EM81:   .ASCIZ  /SPIN NOT SET IN RKMR2/
11502  063242  052117  051440  052105
11503  063250  044440  020116  045522
11504  063256  051115  000062
11505  063262  052122  020132  047516  EM82:   .ASCIZ  /RTZ NOT SET IN RKMR2/
11506  063270  020124  042523  020124
11507  063276  047111  051040  046513
11508  063304  031122     000
11509  063307     122  040505  020104  EM83:   .ASCIZ  /READ HEADER ERR   WORD 0 (CYL#)/
11510  063314  042510  042101  051105
11511  063322  042440  051122  020040
11512  063330  053440  051117  020104
11513  063336  020060  041450  046131
11514  063344  024443     000
11515  063347     106  051117  040515  EM84:   .ASCIZ  /FORMAT IN RKMR3 NOT SET/
11516  063354  020124  047111  051040
11517  063362  046513  031522  047040
11518  063370  052117  051440  052105
11519  063376     000
11520  063377     111  046114  040440  EM85:   .ASCIZ  /ILL ADDR IN RKMR3 NOT =0/
11521  063404  042104  020122  047111
11522  063412  051040  046513  031522
11523  063420  047040  052117  036440
11524  063426  000060
11525  063430  044127  046111  020105  EM86:   .ASCIZ  /WHILE WAITING FOR CONTR RDY OR AFTER CONTR RDY REC'D/
11526  063436  040527  052111  047111
11527  063444  020107  047506  020122
11528  063452  047503  052116  020122
11529  063460  042122  020131  051117
11530  063466  040440  052106  051105
11531  063474  041440  047117  051124
11532  063502  051040  054504  051040
11533  063510  041505  042047     000
11534  063515     103  047101  047516  EM87:   .ASCIZ  /CANNOT READ BSE INFO/
11535  063522  020124  042522  042101
11536  063530  041040  042523  044440
11537  063536  043116  000117
11538  063542  047516  042040  044522  EM88:   .ASCII  /NO DRIVES FOUND ON BUS/<CR><LF>
11539  063550  042526  020123  047506
11540  063556  047125  020104  047117
11541  063564  041040  051525  005015
11542  063572  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY & PRESS 'CONT'/<CR><LF>
11543  063600  047503  051122  041505
11544  063606  046124  020131  020046
11545  063614  051120  051505  020123
11546  063622  041447  047117  023524
11547  063630  005015     000
11548  063633     116  020117  051104  EM89:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/<CR><LF>
11549  063640  053111  051505  043040
11550  063646  052517  042116  044440
11551  063654  020116  042504  044526
11552  063662  042503  046440  050101
```

```
11553  063670  024040  042044  053105
11554  063676  024515  005015
11555  063702  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY & RESTART/<CR><LF>
11556  063710  047503  051122  041505
11557  063716  046124  020131  020046
11558  063724  042522  052123  051101
11559  063732  006524  000012
11560
11561                                  .SBTTL  DATA HEADERS
11562
11563  063736  042524  052123  047040  DH1:    .ASCIZ  /TEST NO.  PC/
11564  063744  027117  020040  041520
11565  063752     000
11566  063753     122  046513  030522  DH2:    .ASCIZ  /RKMR1  RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2/
11567  063760  051011  046513  031122
11568  063766  051011  046513  031522
11569  063774  051011  042513  004522
11570  064002  045522  051504  051011
11571  064010  041513  030523  051011
11572  064016  041513  031123     000
11573  064023     122  053513  004503  DH3:    .ASCIZ  /RKWC   RKBA    RKDA    RKASOF  RKDC    RKECPS  RKECPT/
11574  064030  045522  040502  051011
11575  064036  042113  004501  045522
11576  064044  051501  043117  051011
11577  064052  042113  004503  045522
11578  064060  041505  051520  051011
11579  064066  042513  050103  000124
11580  064074  051106  046517  041440  DH6:    .ASCIZ  /FROM CYL  TO CYL  CYL DIFF/
11581  064102  046131  020040  047524
11582  064110  041440  046131  020040
11583  064116  054503  020114  044504
11584  064124  043106     000
11585  064127     127  042510  020116  DH8:    .ASCIZ  /WHEN DRIVE UNLOADED/
11586  064134  051104  053111  020105
11587  064142  047125  047514  042101
11588  064150  042105     000
11589  064153     101  052106  051105  DH9:    .ASCIZ  /AFTER START SPIN CMD REC'D BY DRIVE/
11590  064160  051440  040524  052122
11591  064166  051440  044520  020116
11592  064174  046503  020104  042522
11593  064202  023503  020104  054502
11594  064210  042040  044522  042526
11595  064216     000
11596  064217     101  020124  047105  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
11597  064224  020104  043117  044040
11598  064232  040505  020104  047514
11599  064240  042101  047111  000107
11600  064246  043101  042524  020122  DH11:   .ASCIZ  /AFTER START SPIN CMD & FWD SET/
11601  064254  052123  051101  020124
11602  064262  050123  047111  041440
11603  064270  042115  023040  043040
11604  064276  042127  051440  052105
11605  064304     000
11606  064305     101  020124  047111  DH12:   .ASCIZ  /AT INNER LIM FROM START SPIN CMD/
11607  064312  042516  020122  044514
11608  064320  020115  051106  046517
```

```
11609  064326  051440  040524  052122
11610  064334  051440  044520  020116
11611  064342  046503  000104
11612  064346  051106  046517  047440  DH13:   .ASCIZ  /FROM OUTER LIM TO CYL 0 DURING LOADING/
11613  064354  052125  051105  046040
11614  064362  046511  052040  020117
11615  064370  054503  020114  020060
11616  064376  052504  044522  043516
11617  064404  046040  040517  044504
11618  064412  043516     000
11619  064415     101  052106  051105  DH14:   .ASCIZ  /AFTER SEEK WITH BAD PAR/
11620  064422  051440  042505  020113
11621  064430  044527  044124  041040
11622  064436  042101  050040  051101
11623  064444     000
11624  064445     101  052106  051105  DH16:   .ASCIZ  /AFTER LOADING HEAD REG & SEEK CMD/
11625  064452  046040  040517  044504
11626  064460  043516  044040  040505
11627  064466  020104  042522  020107
11628  064474  020046  042523  045505
11629  064502  041440  042115     000
11630  064507     101  052106  051105  DH17:   .ASCIZ  /AFTER RECAL CMD/
11631  064514  051040  041505  046101
11632  064522  041440  042115     000
11633  064527     101  052106  051105  DH18:   .ASCIZ  /AFTER UNLOAD CMD/
11634  064534  052440  046116  040517
11635  064542  020104  046503  000104
11636  064550  043101  042524  020122  DH19:   .ASCIZ  /AFTER PACK CMD/
11637  064556  040520  045503  041440
11638  064564  042115     000
11639  064567     101  052106  051105  DH20:   .ASCIZ  /AFTER SELECT DRIVE CMD/
11640  064574  051440  046105  041505
11641  064602  020124  051104  053111
11642  064610  020105  046503  000104
11643  064616  043101  042524  020122  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
11644  064624  052523  051502  051531
11645  064632  042524  020115  046103
11646  064640  040505  000122
11647  064644  043101  042524  020122  DH22:   .ASCIZ  /AFTER DRIVE CLEAR CMD/
11648  064652  051104  053111  020105
11649  064660  046103  040505  020122
11650  064666  046503  000104
11651  064672  042524  052123  047040  DH23:   .ASCIZ  /TEST NO.       TRAP PC/
11652  064700  027117  052011  040522
11653  064706  020120  041520     000
11654  064713     101  052106  051105  DH25:   .ASCIZ  /AFTER SEEK CMD/
11655  064720  051440  042505  020113
11656  064726  046503  000104
11657  064732  043101  042524  020122  DH26:   .ASCIZ  /AFTER READ DATA CMD/
11658  064740  042522  042101  042040
11659  064746  052101  020101  046503
11660  064754  000104
11661  064756  042411  050130  041505  DH28:   .ASCIZ  /       EXPECT/
11662  064764  000124
11663  064766  040411  052103  040525  DH29:   .ASCIZ  /       ACTUAL/
11664  064774  000114
```

```
11665  064776  043101  042524  020122  DH30:   .ASCIZ  /AFTER READ HEADER CMD/
11666  065004  042522  042101  044040
11667  065012  040505  042504  020122
11668  065020  046503  000104
11669  065024  030101  041011  004460  DH32:   .ASCIZ  /A0     B0      A1      B1      A2      B2      B3/
11670  065032  030501  041011  004461
11671  065040  031101  041011  004462
11672  065046  031502     000
11673  065051     104  051125  047111  DH33:   .ASCIZ  /DURING SEEK CMD/
11674  065056  020107  042523  045505
11675  065064  041440  042115     000
11676  065071     123  041505  047524  DH34:   .ASCIZ  /SECTOR REG UNSTABLE/
11677  065076  020122  042522  020107
11678  065104  047125  052123  041101
11679  065112  042514     000
11680  065115     102  052105  042527  DH35:   .ASCIZ  /BETWEEN SECTOR COUNTS/
11681  065122  047105  051440  041505
11682  065130  047524  020122  047503
11683  065136  047125  051524     000
11684  065143     106  047522  020115  DH36:   .ASCIZ  /FROM SECT  TO SECT/
11685  065150  042523  052103  020040
11686  065156  047524  051440  041505
11687  065164  000124
11688  065166  050123  042505  020104  DH37:   .ASCIZ  /SPEED OK NOT =0 10 SEC AFTER UNLOAD/
11689  065174  045517  047040  052117
11690  065202  036440  020060  030061
11691  065210  051440  041505  040440
11692  065216  052106  051105  052440
11693  065224  046116  040517  000104
11694  065232  043101  042524  020122  DH38:   .ASCIZ  /AFTER LIM DET/
11695  065240  044514  020115  042504
11696  065246  000124
11697  065250  043101  042524  020122  DH39:   .ASCIZ  /AFTER WRITE HEADER CMD/
11698  065256  051127  052111  020105
11699  065264  042510  042101  051105
11700  065272  041440  042115     000
11701  065277     127  051117  021504  DH40:   .ASCIZ  /WORD#  HEADER WAS  SHOULD BE/
11702  065304  044011  040505  042504
11703  065312  020122  040527  020123
11704  065320  051440  047510  046125
11705  065326  020104  042502     000
11706  065333     104  051125  047111  DH41:   .ASCIZ  /DURING RECAL CMD/
11707  065340  020107  042522  040503
11708  065346  020114  046503  000104
11709  065354  047117  051440  041505  DH42:   .ASCIZ  /ON SEC 0,2,4,6,8  CYL 410 TRK 2/
11710  065362  030040  031054  032054
11711  065370  033054  034054  020040
11712  065376  054503  020114  030464
11713  065404  020060  051124  020113
11714  065412  000062
11715  065414  047506  046522  052101  DH44:   .ASCIZ  /FORMAT & ALL R-W TESTS WILL BE BYPASSED/
11716  065422  023040  040440  046114
11717  065430  051040  053455  052040
11718  065436  051505  051524  053440
11719  065444  046111  020114  042502
11720  065452  041040  050131  051501
```

```
11721  065460  042523  000104
11722  065464  042502  040503  051525  DH45:   .ASCIZ  /BECAUSE OF LIMIT DETECT ERROR ON PREVIOUS TEST/
11723  065472  020105  043117  046040
11724  065500  046511  052111  042040
11725  065506  052105  041505  020124
11726  065514  051105  047522  020122
11727  065522  047117  050040  042522
11728  065530  044526  052517  020123
11729  065536  042524  052123     000
11730  065543     103  052517  042114  DH46:   .ASCIZ  /COULD NOT READ BSE INFO ON PREV TEST/
11731  065550  047040  052117  051040
11732  065556  040505  020104  051502
11733  065564  020105  047111  047506
11734  065572  047440  020116  051120
11735  065600  053105  052040  051505
11736  065606  000124
11737  065610  043101  042524  020122  DH48:   .ASCIZ  /AFTER SEEK CMD TO INV CYL/
11738  065616  042523  045505  041440
11739  065624  042115  052040  020117
11740  065632  047111  020126  054503
11741  065640  000114
11742  065642  051515  020107  023101  DH49:   .ASCIZ  /MSG A&B IN RKMR2 & RKMR3 RESP., ARE INVALID/
11743  065650  020102  047111  051040
11744  065656  046513  031122  023040
11745  065664  051040  046513  031522
11746  065672  051040  051505  027120
11747  065700  020054  051101  020105
11748  065706  047111  040526  044514
11749  065714  000104
11750  065716  043101  042524  020122  DH51:   .ASCIZ  /AFTER SEEK TO SELF CMD/
11751  065724  042523  045505  052040
11752  065732  020117  042523  043114
11753  065740  041440  042115     000
11754  065745     105  050130  041440  DH52:   .ASCIZ  /EXP CYL#       CYL HEADER WAS/
11755  065752  046131  004443  054503
11756  065760  020114  042510  042101
11757  065766  051105  053440  051501
11758  065774     000
11759  065775     117  020116  042523  DH53:   .ASCIZ  /ON SEC 10,12,14,16,18,20 CYL 410 TRK 2/
11760  066002  020103  030061  030454
11761  066010  026062  032061  030454
11762  066016  026066  034061  031054
11763  066024  020060  054503  020114
11764  066032  030464  020060  051124
11765  066040  020113  000062
11766                          .SBTTL  ERR OUTPUT DATA
11767
11768                                  .EVEN
11769  066044  001214  001116          DT1:    $TESTN,$ERRPC
11770  066050  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11771  066056  003350  003346  003334
11772  066064  003336
11773  066066  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11774  066074  003352  003354  003366
11775  066102  003370
11776  066104  001214  001116  001350  DT4:    $TESTN,$ERRPC,FRCYL,TOCYL,CALDIF
```

```
11777  066112  001352  001360
11778  066116  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11779  066124  003350  003346  003334
11780  066132  003336
11781  066134  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11782  066142  003352  003354  003366
11783  066150  003370
11784  066152  001214  001116  001402  DT6:    $TESTN,$ERRPC,PSEC,ESEC
11785  066160  001404
11786  066162  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11787  066170  003350  003346  003334
11788  066176  003336
11789  066200  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11790  066206  003352  003354  003366
11791  066214  003370
11792  066216  001214  001116  001442  DT7:    $TESTN,$ERRPC,WDCNT,HDWD,TEMP1
11793  066224  001454  003372
11794  066230  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11795  066236  003350  003346  003334
11796  066244  003336
11797  066246  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11798  066254  003352  003354  003366
11799  066262  003370
11800  066264  001214  001116  001352  DT8:    $TESTN,$ERRPC,TOCYL,FRCYL,CALDIF
11801  066272  001350  001360
11802  066276  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11803  066304  003350  003346  003334
11804  066312  003336
11805  066314  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11806  066322  003352  003354  003366
11807  066330  003370
11808  066332  001214  001116  001352  DT9:    $TESTN,$ERRPC,TOCYL,RHTAB
11809  066340  001674
11810  066342  003360  003362  003364          HMR1,HMR2,HMR3, HER,HDS,HCS1,HCS2
11811  066350  003350  003346  003334
11812  066356  003336
11813  066360  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11814  066366  003352  003354  003366
11815  066374  003370
11816  066376  001214  001116  001350  DT10:   $TESTN,$ERRPC,FRCYL,RHTAB
11817  066404  001674
11818  066406  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11819  066414  003350  003346  003334
11820  066422  003336
11821  066424  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11822  066432  003352  003354  003366
11823  066440  003370
11824  066442  001214  001334          DT11:   $TESTN,TRAPPC
11825  066446  001214  001116  003424  DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
11826  066454  003426  003430  003432
11827  066462  003404  003406  003410
11828  066470  003412
11829  066472  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11830  066500  003350  003346  003334
11831  066506  003336
11832  066510  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
```

```
11833  066516  003352  003354  003366
11834  066524  003370
11835
11836  066526  001214  001116  003424  DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
11837  066534  003426  003430  003432
11838  066542  003434  003436
11839  066546  003404  003406  003410          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
11840  066554  003412  003414  003416
11841  066562  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11842  066570  003350  003346  003334
11843  066576  003336
11844  066600  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11845  066606  003352  003354  003366
11846  066614  003370
11847
11848  066616  001214  001116  003424  DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
11849  066624  003426  003430  003432
11850  066632  003434  003436  003442
11851  066640  003404  003406  003410          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
11852  066646  003412  003414  003416
11853  066654  003422
11854  066656  003360  003362  003364          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11855  066664  003350  003346  003334
11856  066672  003336
11857  066674  003340  003342  003344          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11858  066702  003352  003354  003366
11859  066710  003370
11860
11861                                  .SBTTL  ERR DATA FORMATS
11862
11863  066712  000003                  DF1:    3
11864  066714     002     000                  .BYTE 2,0
11865  066716  063753                          DH2
11866  066720     007     000                  .BYTE 7,0
11867  066722  064023                          DH3
11868  066724     007     000                  .BYTE   7,0
11869
11870  066726  000005                  DF2:    5
11871  066730     000     000                  .BYTE 0,0
11872  066732  065642                          DH49
11873  066734     000     000                  .BYTE 0,0
11874  066736  063736                          DH1
11875  066740     002     000                  .BYTE 2,0
11876  066742  063753                          DH2
11877  066744     007     000                  .BYTE 7,0
11878  066746  064023                          DH3
11879  066750     007     000                  .BYTE   7,0
11880
11881  066752  000001                  DF3:    1
11882  066754     002     000                  .BYTE   2,0
11883  066756  000003                  DF4:    3
11884  066760     002     000                  .BYTE   2,0
11885  066762  063753                          DH2
11886  066764     007     000                  .BYTE   7,0
11887  066766  064023                          DH3
11888  066770     007     000                  .BYTE   7,0
```

```
11889
11890  066772  000004                DF5:    4
11891  066774     000      000               .BYTE 0,0
11892  066776  063736                        DH1
11893  067000     002      000               .BYTE 2,0
11894  067002  063753                        DH2
11895  067004     007      000               .BYTE  7,0
11896  067006  064023                        DH3
11897  067010     007      000               .BYTE  7,0
11898
11899  067012  000005                DF6:    5
11900  067014     000      000               .BYTE 0,0
11901  067016  063736                        DH1
11902  067020     002      000               .BYTE 2,0
11903  067022  064074                        DH6
11904  067024     003      000               .BYTE 3,0
11905  067026  063753                        DH2
11906  067030     007      000               .BYTE  7,0
11907  067032  064023                        DH3
11908  067034     007      000               .BYTE  7,0
11909
11910
11911
11912  067036  000004                DF10:   4
11913  067040     000      000               .BYTE 0,0
11914  067042  063736                        DH1
11915  067044     002      000               .BYTE 2,0
11916  067046  063753                        DH2
11917  067050     007      000               .BYTE 7,0
11918  067052  064023                        DH3
11919  067054     007      000               .BYTE  7,0
11920
11921  067056  000005                DF12:   5
11922  067060     000      000               .BYTE 0,0
11923  067062  063736                        DH1
11924  067064     002      000               .BYTE 2,0
11925  067066  065143                        DH36
11926  067070     002      000               .BYTE 2,0
11927  067072  063753                        DH2
11928  067074     007      000               .BYTE  7,0
11929  067076  064023                        DH3
11930  067100     007      000               .BYTE  7,0
11931
11932  067102  000004                DF14:   4
11933  067104     002      000               .BYTE  2,0
11934  067106  065277                        DH40
11935  067110     003      000               .BYTE  3,0
11936  067112  063753                        DH2
11937  067114     007      000               .BYTE  7,0
11938  067116  064023                        DH3
11939  067120     007      000               .BYTE  7,0
11940
11941
11942  067122  000004                DF15:   4
11943  067124     000      000               .BYTE  0,0
11944  067126  063736                        DH1
```

```
11945  067130  002  000                .BYTE   2,0
11946  067132  063753                  DH2
11947  067134  007  000                .BYTE   7,0
11948  067136  064023                  DH3
11949  067140  007  000                .BYTE   7,0
11950
11951  067142  000004            DF16: 4
11952  067144  000  000                .BYTE 0,0
11953  067146  063736                  DH1
11954  067150  002  000                .BYTE 2,0
11955  067152  063753                  DH2
11956  067154  007  000                .BYTE   7,0
11957  067156  064023                  DH3
11958  067160  007  000                .BYTE   7,0
11959
11960  067162  000005            DF17: 5
11961  067164  000  000                .BYTE   0,0
11962  067166  065414                  DH44
11963  067170  000  000                .BYTE   0,0
11964  067172  063736                  DH1
11965  067174  002  000                .BYTE   2,0
11966  067176  063753                  DH2
11967  067200  007  000                .BYTE   7,0
11968  067202  064023                  DH3
11969  067204  007  000                .BYTE   7,0
11970  067206  000007            DF20: 7
11971  067210  000  000                .BYTE   0,0
11972  067212  063736                  DH1
11973  067214  002  000                .BYTE   2,0
11974  067216  064756                  DH28
11975  067220  000  000                .BYTE   0,0
11976  067222  065024                  DH32
11977  067224  004  000                .BYTE   4,0
11978  067226  064766                  DH29
11979  067230  004  000                .BYTE   4,0
11980  067232  063753                  DH2
11981  067234  007  000                .BYTE   7,0
11982  067236  064023                  DH3
11983  067240  007  000                .BYTE   7,0
11984  067242  000004            DF21: 4
11985  067244  002  000                .BYTE   2,0
11986  067246  065745                  DH52
11987  067250  002  000                .BYTE   2,0
11988  067252  063753                  DH2
11989  067254  007  000                .BYTE   7,0
11990  067256  064023                  DH3
11991  067260  007  000                .BYTE   7,0
11992  067262  000007            DF22: 7
11993  067264  000  000                .BYTE   0,0
11994  067266  063736                  DH1
11995  067270  002  000                .BYTE   2,0
11996  067272  064756                  DH28
11997  067274  000  000                .BYTE   0,0
11998  067276  065024                  DH32
11999  067300  006  000                .BYTE   6,0
12000  067302  064766                  DH29
```

```
12001  067304     006      000                .BYTE   6,0
12002  067306  063753                         DH2
12003  067310     007      000                .BYTE   7,0
12004  067312  064023                         DH3
12005  067314     007      000                .BYTE   7,0
12006
12007  067316  000007                 DF23:   7
12008  067320     000      000                .BYTE   0,0
12009  067322  063736                         DH1
12010  067324     002      000                .BYTE   2,0
12011  067326  064756                         DH28
12012  067330     000      000                .BYTE   0,0
12013  067332  065024                         DH32
12014  067334     007      000                .BYTE   7,0
12015  067336  064766                         DH29
12016  067340     007      000                .BYTE   7,0
12017  067342  063753                         DH2
12018  067344     007      000                .BYTE   7,0
12019  067346  064023                         DH3
12020  067350     007      000                .BYTE   7,0
```

```
 12021                                  ;;*****************************************************************
 12022                                  .SBTTL  TYPE ERR ROUTINE
 12023                                  ;*ENTRY JSR PC,TYP ERR
 12024                                  ;*RETURN RTS PC
 12025                                  ;*
 12026                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 12027                                  ;*ERR IS TO BE REPORTED. IT THEN USES THE "ERR TABLE" ($ERRTB)
 12028                                  ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
 12029                                  ;*THE ERR.
 12030                                  ;;*****************************************************************
 12031  067352  104413                  TYPERR: SAVREG
 12032  067354  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERR NUMBER
 12033  067360  042700  177400                  BIC     #177400,R0      ;CLEAR SIGN EXTENSION
 12034  067364  005300                          DEC     R0              ;FORM INDEX FOR ERR TABLE
 12035  067366  006300                          ASL     R0
 12036  067370  006300                          ASL     R0
 12037  067372  006300                          ASL     R0
 12038  067374  062700  003506          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERR ENTRY
 12039  067400  012037  067414                  MOV     (R0)+,2$        ;GET EM POINTER
 12040  067404  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
 12041  067406  104401  001205                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
 12042  067412  104401                          TYPE                    ;TYPE ERR MSG (EM)
 12043  067414  000000          2$:     .WORD   0                       ;EM POINTER GOES HERE
 12044  067416  012037  067432          3$:     MOV     (R0)+,4$        ;GET DH POINTER
 12045  067422  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
 12046  067424  104401  001205                  TYPE    ,$CRLF          ;TYPE CR-LF
 12047  067430  104401                          TYPE                    ;TYPE DATA HEADER
 12048  067432  000000          4$:     .WORD   0                       ;DH POINTER GOES HERE
 12049  067434  012001          5$:     MOV     (R0)+,R1        ;GET DT POINTER
 12050  067436  001455                  BEQ     20$             ;BRANCH IF THERE ARE NONE
 12051  067440  005004                  CLR     R4              ;SET INDENT SWITCH
 12052  067442  012000                  MOV     (R0)+,R0        ;GET DF POINTER
 12053  067444  012002                  MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
 12054  067446  001446                  BEQ     17$             ;DH NUM IS 0-BRANCH
 12055  067450  005104                  COM     R4              ;NO INDENT
 12056  067452  104401  001205          TYPE    ,$CRLF
 12057  067456  112003          10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
 12058  067460  105720                  TSTB    (R0)+           ;BUMP PAST FORMAT WORD
 12059  067462  005703                  TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
 12060  067464  001407                  BEQ     14$             ;NO - SKIP DATA PRINT
 12061  067466  013146          11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
 12062  067470  104402                  TYPOC                   ;TYPE IT
 12063  067472  005303                  DEC     R3              ;MORE DATA WORDS
 12064  067474  001403                  BEQ     14$             ;NO-BRANCH
 12065  067476  104401  067626          TYPE    ,SPACE2         ;TYPE SEPARATORS
 12066  067502  000771                  BR      11$             ;LOOP
 12067  067504  005302          14$:    DEC     R2              ;MORE DH'S?
 12068  067506  003431                  BLE     20$             ;NO-BRANCH
 12069  067510  104401  001205          TYPE    ,$CRLF
 12070  067514  005760  000002          TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
 12071  067520  001404                  BEQ     15$             ;YES-BRANCH BYPASS INDENT
 12072  067522  005104                  COM     R4              ;INDENT?
 12073  067524  001002                  BNE     15$             ;NO-BRANCH
 12074  067526  104401  067626          TYPE    ,SPACE2         ;YES-TYPE SPACES
 12075  067532  012037  067540  15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
 12076  067536  104401                  TYPE                    ;TYPE DH
```

```
 12077   067540   000000                 16$:    .WORD   0               ;DH POINTER GOES HERE
 12078   067542   105710                         TSTB    (R0)            ;TYPE A DT?
 12079   067544   001003                         BNE     21$             ;YES-BRANCH
 12080   067546   062700   000002                ADD     #2,R0           ;INCREMENT DF POINTER
 12081   067552   000754                         BR      14$             ;SEE IF END OF DF BLOCK
 12082   067554   104401   001205        21$:    TYPE    ,$CRLF
 12083   067560   005704                         TST     R4              ;INDENT?
 12084   067562   001335                         BNE     10$             ;NO-BRANCH
 12085   067564   104401   067626        17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES
 12086   067570   000732                         BR      10$             ;LOOP
 12087   067572   104414                 20$:    RESREG
 12088
 12089   067574   032777   010000   111336       BIT     #SW12,@SWR      ;ABORT DRV AFTER 20 ERRS?
 12090   067602   001410                         BEQ     25$             ;BR IF NO
 12091   067604   023727   001103   000024       CMP     $ERFLG,#20.     ;ELSE SEE IF 20 ERRS
 12092   067612   001004                         BNE     25$             ;BR IF NO
 12093   067614   012706   001100               MOV     #STACK,SP       ;ELSE RESTORE STK
 12094   067620   000137   043076               JMP     $EOP            ;AND DROP DRIVE
 12095   067624   000207                 25$:    RTS     PC
 12096   067626   020040      000        SPACE2: .ASCIZ/  /              ;2 SPACES
```

```
12097                                   ;  ODT-11 -- V005A
12098
12099                                   ; DEC-11-UODPA-A-LA
12100
12101                                   ; COPYRIGHT 1969,1970,1972
12102                                   ; DIGITAL EQUIPMENT CORPORATION
12103                                   ; MAYNARD, MASSACHUSETTS 01754
12104                                           .ENABL  ABS,AMA
12105   067632                                  .EVEN
12106   067712                                  .=.+60
12107   000000                          R0      =       %0      ; REGISTER
12108   000001                          R1      =       %1      ;  NAMING
12109   000002                          R2      =       %2      ;   CONVENTIONS
12110   000003                          R3      =       %3
12111   000004                          R4      =       %4
12112   000005                          R5      =       %5
12113   000006                          SP      =       %6
12114   000007                          PC      =       %7
12115   177776                          ST      =       177776          ;STATUS REGISTER
12116                                   ;
12117   000014                          0.TVEC  =       14              ;TRT VECTOR LOCATION
12118   000340                          0.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
12119   000020                          0.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
12120   000003                          TRT     =       000003          ;TRT INSTRUCTION
12121   000006                          RTT     =       000006          ;RTT INSTRUCTION
12122                                   ;
12123                                   ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
12124                                   ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
12125                                   ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
12126                                   ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
12127                                   ;
12128   177562                          0.RDB   =       177562  ;R DATA BUFFER
12129   177560                          0.RCSR  =       177560  ;R C/SR
12130   177566                          0.TDB   =       177566  ;T DATA BUFFER
12131   177564                          0.TCSR  =       177564  ;T C/SR
12132                                   ;
12133                                   ;
12134                                   ; INITIALIZE ODT
12135                                   ;  USE 0.ODT FOR A NORMAL ENTRY
12136                                   ;  USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
12137                                   ;  USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
12138                                   ;
12139   067712  000413                  0.ODT:  BR      0.STRT          ;NORMAL ENTRY
12140   067714  000417                          BR      0.RST           ;RESTART
12141   067716  013737  177776  067672  0.ENTR: MOV     ST,0.UST        ;RE-ENTER  --  SAVE STATUS
12142   067724  013737  000016  177776          MOV     @0.TVEC+2,ST    ;SET UP LOCAL STATUS
Z 12143 067732  010737  067670                  MOV     PC,0.UPC        ;FAKE THE PC
12144   067736  000137  071070                  JMP     @0.BK1
12145                                   ;
12146   067742  012706  067652          0.STRT: MOV     #0.URO,SP       ;SET UP STACK
12147   067746  010637  067666                  MOV     SP,@0.USP       ;FAKE THE SAVED STACK
12148   067752  000414                          BR      0.RST1          ;CLEAR BREAKPOINT TABLES
12149   067754  004037  071276          0.RST:  JSR     0,@0.SVR        ;SAVE REGISTERS
12150   067760  013777  067710  177716          MOV     @0.UIN,@0.ADR1  ;REMOVE THE BREAKPOINT
12151   067766  113704  067674                  MOVB    @0.PRI,R4       ;GET ODT PRIORITY
12152   067772  106004                          RORB    R4              ;SHIFT
```

```
 12153  067774  106004                          RORB    R4              ; INTO
 12154  067776  106004                          RORB    R4              ;  POSITION
 12155  070000  110437  177776                  MOVB    R4,ST           ;STORE IN STATUS
Z12156  070004  000127                  0.RST1: JMP     (PC)+
 12157  070006  000403                          BR      0.45
 12158  070010  012737  000002  071000          MOV     #RTI,0.RTIT     ;SET TO RTI IF 11/20 OR /05
 12159  070016  105037  071717          0.45:   CLRB    0.P             ;DISALLOW PROCEED
 12160  070022  012737  000340  000016          MOV     #0.STM,0.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
 12161  070030  012737  071060  000014          MOV     #0.BRK,0.TVEC   ;PC TO TRT VECTOR
 12162  070036  000447                          BR      0.RALL          ;CLEAR BREAKPOINT TABLES
 12163                                  ;
 12164                                  ; SPECIAL NAME HANDLER
 12165                                  ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URO
 12166                                  ;
 12167  070040  004537  071520          0.REGT: JSR     5,0.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
 12168  070044  012704  071743                  MOV     #0.TL,R4        ;TABLE START ADDRESS
 12169  070050  120024                  0.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
 12170  070052  001413                          BEQ     0.SP            ;JUMP IF YES
 12171  070054  022704  071751                  CMP     #0.TL+0.LG,R4   ;IS THE SEARCH DONE?
 12172  070060  101373                          BHI     0.RSP           ;BRANCH IF NOT
 12173  070062  042700  177770                  BIC     #177770,R0      ;MASK OFF OCTAL
 12174  070066  010004                          MOV     R0,R4
 12175  070070  006304                  0.SP1:  ASL     R4
 12176  070072  062704  067652                  ADD     #0.URO,R4       ;GENERATE ADDRESS
 12177  070076  005202                          INC     R2              ;SET FOUND FLAG
 12178  070100  000444                          BR      0.SCAN          ;GO FIND NEXT CHARACTER
 12179  070102  162704  071734          0.SP:   SUB     #0.TL-7,R4      ;CORRECT CONSTANT
 12180  070106  000770                          BR      0.SP1
 12181                                  ;
 12182                                  ; ← HANDLER - OPEN INDEXED ON THE PC
 12183                                  ;
 12184  070110  004737  071644          0.ORPC: JSR     PC,0.TCLS
 12185  070114  010502                          MOV     R5,R2           ;CURRENT ADDRESS IN R2
 12186  070116  061202                          ADD     @R2,R2          ;COMPUTE
 12187  070120  006202                          ASR     R2              ;MOVE ONE BIT TO CARRY
 12188  070122  103421                          BCS     0.ERR           ;ERR IF ODD NUMBER
 12189  070124  006302                          ASL     R2              ;RESTORE WORD
 12190  070126  005722                          TST     (R2)+           ;  AND INCREMENT BY TWO
 12191  070130  010205                          MOV     R2,R5           ;UPDATE CAD
 12192  070132  000137  070404                  JMP     0.OP2           ;GO FINISH UP
 12193                                  ;
 12194                                  ; B HANDLER - SET AND REMOVE BREAKPOINTS
 12195                                  ;
 12196  070136  005702                  0.BKPT: TST     R2              ;IF NO NUMBER TYPED
 12197  070140  001406                          BEQ     0.RALL          ;  REMOVE BREAKPOINT
 12198  070142  006204                          ASR     R4              ;CHECK IF ODD
 12199  070144  103410                          BCS     0.ERR           ;JUMP IF ODD
 12200  070146  006304                          ASL     R4              ;RESTORE ONE BIT
 12201  070150  010437  067704                  MOV     R4,0.ADR1       ;SET A BREAKPOINT
 12202  070154  000412                          BR      0.DCD
 12203  070156  012737  071760  067704  0.RALL: MOV     #0.TRTC,0.ADR1  ;CLEAR BREAKPOINT
 12204  070164  000406                          BR      0.DCD
 12205                                  ;
 12206                                  ; CMD DECODER - ODT11
 12207                                  ;
 12208                                  ;   REGISTERS R0-R4 MAY BE USED,
```

```
12209                                   ;    REGISTER R5 WILL BE CONSIDERED SAFE
12210                                   ;
12211  070166  052705  000001  0.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
12212  070172  012700  000077          MOV     #'?,R0          ;   ? TO BE TYPED
12213  070176  004537  071576          JSR     5,0.FTYP        ; OUTPUT ?
12214  070202  004537  071676  0.DCD:  JSR     5,0.CRLS        ;TYPE  <CR><LF>*
12215  070206  005004          0.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
12216  070210  005002                  CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
12217  070212  004537  071520  0.SCAN: JSR     5,0.GET         ;GET A CHAR, RETURN IN R0
12218  070216  022700  000060          CMP     #'0,R0          ;COMPARE WITH ASCII 0
12219  070222  101013                  BHI     0.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
12220  070224  022700  000067          CMP     #'7,R0          ;COMPARE WITH ASCII 7
12221  070230  103410                  BLO     0.CLGL          ;CHECK LEGALITY IF NOT OCTAL
12222  070232  042700  177770          BIC     #177770,R0      ;CONVERT TO BCD
12223  070236  006304                  ASL     R4              ; MAKE ROOM
12224  070240  006304                  ASL     R4              ;   IN
12225  070242  006304                  ASL     R4              ;     R4
12226  070244  060004                  ADD     R0,R4           ;PACK THREE BITS IN R4
12227  070246  005202                  INC     R2              ;R2 HAS NUMERIC FLAG
12228  070250  000760                  BR      0.SCAN          ;    AND TRY AGAIN
12229  070252  005001  0.CLGL: CLR     R1              ;CLEAR INDEX
12230  070254  120061  071727  0.LGL1: CMPB    R0,0.LGCH(R1)   ;DO THE CODES MATCH?
12231  070260  001405                  BEQ     0.LGL2          ;JUMP IF YES
12232  070262  005201                  INC     R1              ; SET INDEX FOR NEXT SEARCH
12233  070264  020127  000014          CMP     R1,#0.CLGT      ;IS THE SEARCH DONE?
12234  070270  103336                  BHIS    0.ERR           ;    OOPS!
12235  070272  000770                  BR      0.LGL1          ;RE-LOOP
12236  070274  006301  0.LGL2: ASL     R1              ;MULTIPLY BY TWO
12237  070276  000171  070302          JMP     @0.LGDR(R1)     ;GO TO PROPER ROUTINE
12238                                   ;
12239  070302  070332  0.LGDR: 0.WRD    ;  /    OPEN WORD
12240  070304  070364         0.CRET   ;  CARRIAGE RETURN    CLOSE
12241  070306  070040         0.REGT   ;  $  REGISTER OPS
12242  070310  070674         0.GO     ;  G  GO TO ADDRESS K
12243  070312  070376         0.OP1    ;  <LF>  MODIFY, CLOSE, OPEN NEXT
12244  070314  070110         0.ORPC   ;  ↑  OPEN RELATED, INDEX - PC
12245  070316  070430         0.BACK   ;  ↑  OPEN PREVIOUS
12246  070320  070440         0.OFST   ;  O  OFFSET
12247  070322  070516         0.WSCH   ;  W  SEARCH WORD
12248  070324  070512         0.EFF    ;  E  SEARCH EFFECTIVE ADDRESS
12249  070326  070136         0.BKPT   ;  B  BREAKPOINTS
12250  070330  071002         0.PROC   ;  P  PROCEED
12251          000030  0.LGL   =       .-0.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS
12252                                   ;
12253                                   ; PROCESS / - OPEN WORD
12254                                   ;
12255  070332  005702  0.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
12256  070334  001410                  BEQ     0.WRDA          ;SKIP OTHERWISE
12257  070336  010405                  MOV     R4,R5           ;   PUT VALUE IN CAD
12258  070340  006205  0.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
12259  070342  103711  0.ERR2: BCS     0.ERR           ;JUMP IF ODD ADDRESS
12260  070344  006305                  ASL     R5              ;RESTORE THE CARRY BIT
12261  070346  011500                  MOV     @R5,R0          ;GET CONTENTS OF WORD
12262  070350  004537  071434          JSR     5,0.CADV        ;GO GET AND TYPE OUT @CAD
12263  070354  000714                  BR      0.DCD1          ;GO BACK TO DECODER
12264  070356  042705  000001  0.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
```

```
12265  070362  000766                  BR      0.WRD1          ;GO BACK TO MAIN-LINE
12266                                  ;
12267                                  ; PROCESS CARRIAGE RETURN
12268                                  ;
12269  070364  004737  071644          0.CRET: JSR     PC,0.TCLS       ;CLOSE LOCATION
12270  070370  052705  000001                  BIS     #1,R5           ;CLOSE EVERYTHING
12271  070374  000702                          BR      0.DCD           ;RETURN TO DECODER
12272                                  ;
12273                                  ; PROCESS <LF>, OPEN NEXT WORD
12274                                  ;
12275  070376  004737  071644          0.OP1:  JSR     PC,0.TCLS       ;CLOSE PRESENT CELL
12276  070402  005725                          TST     (R5)+           ;GENERATE NEW ADDRESS
12277  070404  004537  071670          0.OP2:  JSR     5,0.CRLF        ;<CR><LF>
12278  070410  010500                          MOV     R5,R0           ;NUMBER TO TYPE
12279  070412  004537  071434                  JSR     5,0.CADV        ; TYPE OUT ADDRESS
12280  070416  012700  000057                  MOV     #'/,R0          ;TYPE A /
12281  070422  004537  071576                  JSR     5,0.FTYP
12282  070426  000744                          BR      0.WRD1          ;GO PROCESS IT
12283                                  ;
12284                                  ; PROCESS ↑, OPEN PREVIOUS WORD
12285                                  ;
12286  070430  004737  071644          0.BACK: JSR     PC,0.TCLS
12287  070434  005745                          TST     -(R5)           ;GENERATE NEW ADDRESS
12288  070436  000762                          BR      0.OP2           ;GO DO THE REST
12289                                  ;
12290                                  ; PROCESS 0, COMPUTE OFFSET
12291                                  ;
12292  070440  006205                  0.OFST: ASR     R5              ;GET LOW ORDER BIT
12293  070442  103737                          BCS     0.ERR2          ;ERR IF CLOSED
12294  070444  006305                          ASL     R5              ;RESTORE WORD
12295  070446  012700  000040                  MOV     #' ,R0          ;TYPE ONE BLANK
12296  070452  004537  071576                  JSR     5,0.FTYP        ; AS A SEPARATOR
12297  070456  160504                          SUB     R5,R4           ;COMPUTE
12298  070460  005304                          DEC     R4
12299  070462  005304                          DEC     R4              ;     16 BIT OFFSET
12300  070464  010400                          MOV     R4,R0           ;TYPE A
12301  070466  010402                          MOV     R4,R2           ;SAVE R4
12302  070470  004537  071434                  JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
12303  070474  010200                          MOV     R2,R0
12304  070476  006200                          ASR     R0              ;DIVIDE BY TWO
12305  070500  103402                          BCS     0.OF1           ;BRANCH IF ODD
12306  070502  004537  071434                  JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
12307  070506  000137  070206          0.OF1:  JMP     0.DCD1          ;ALL DONE
12308                                  ;
12309                                  ; SEARCHES - $MSK   HAS THE MASK
12310                                  ;            $MSK+2 HAS THE FWA
12311                                  ;            $MSK+4 HAS THE LWA
12312                                  ;
12313  070512  005201                  0.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
12314  070514  000401                          BR      0.WDS
12315  070516  005001                  0.WSCH: CLR     R1              ;SET WORD SEARCH
12316  070520  005702                  0.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
12317  070522  001621                  0.ERR1: BEQ     0.ERR           ;ERR IF NO OBJECT
12318  070524  013702  067700                  MOV     0.MSK+2,R2      ;SET ORIGIN
12319  070530  013705  067676                  MOV     0.MSK,R5        ;SET MASK
12320  070534  005105                          COM     R5              ;AND COMPLEMENT IT
```

```
12321  070536  020237  067702          0.WDS2: CMP     R2,0.MSK+4      ; IS THE SEARCH ALL DONE?
12322  070542  101217                          BHI     0.DCD           ;  YES
12323  070544  011200                          MOV     @R2,R0          ; GET OBJECT
12324  070546  005701                          TST     R1              ;NO
12325  070550  001027                          BNE     0.EFF1          ;BRANCH IF EFFECTIVE SEARCH
12326  070552  010046                          MOV     R0,-(SP)
12327  070554  010403                          MOV     R4,R3           ;EXCLUSIVE OR
12328  070556  040400                          BIC     R4,R0           ; IS DONE
12329  070560  042603                          BIC     (SP)+,R3        ;  IN A VERY
12330  070562  050003                          BIS     R0,R3           ;    FANCY MANNER HERE
12331  070564  040503                          BIC     R5,R3           ;AND RESULT WITH MASK
12332  070566  001016                  0.WDS3: BNE     0.WDS4          ;RE-LOOP IF NO MATCH
12333  070570  010446                          MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
12334  070572  004537  071670                  JSR     5,0.CRLF        ;TYPE <CR,LF>
12335  070576  010200                          MOV     R2,R0           ;GET READY TO TYPE
12336  070600  004537  071434                  JSR     5,0.CADV        ;  TYPE ADDRESS
12337  070604  012700  000057                  MOV     #'/,R0          ;SLASH TO R0
12338  070610  004537  071576                  JSR     5,0.FTYP        ;TYPE IT
12339  070614  011200                          MOV     @R2,R0          ;GET CONTENTS
12340  070616  004537  071434                  JSR     5,0.CADV        ;TYPE CONTENTS
12341  070622  012604                          MOV     (SP)+,R4        ; RESTORE R4
12342  070624  005722                  0.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
12343  070626  000743                          BR      0.WDS2          ;    RETURN
12344  070630  020004                  0.EFF1: CMP     R0,R4           ; IS (X)=K?
12345  070632  001755                          BEQ     0.WDS3          ;TYPE IF EQUAL
12346  070634  010003                          MOV     R0,R3           ;(X) TO R3
12347  070636  060203                          ADD     R2,R3           ;(X)+X
12348  070640  005203                          INC     R3
12349  070642  005203                          INC     R3              ;(X)+X+2
12350  070644  020304                          CMP     R3,R4           ;IS (X)+X+2=K?
12351  070646  001747                          BEQ     0.WDS3          ;BRANCH IF EQUAL
12352  070650  042700  177400                  BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
12353  070654  110000                          MOVB    R0,R0           ;EXTEND SIGN
12354  070656  000257                          CCC
12355  070660  006300                          ASL     R0              ;MULTIPLY BY TWO
12356  070662  005200                          INC     R0              ;ADD TWO
12357  070664  005200                          INC     R0
12358  070666  060200                          ADD     R2,R0           ;ADD PC
12359  070670  020004                          CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
12360  070672  000735                          BR      0.WDS3
12361                                  ;
12362                                  ; PROCESS G - GO
12363                                  ;
12364  070674  105037  071717          0.GO:   CLRB    0.P             ;DISALLOW PROCEED
12365  070700  006204                          ASR     R4              ;CHECK LOW ORDER BIT
12366  070702  103617                          BCS     0.ERR2          ;ERR IF ODD NUMBER
12367  070704  006304                          ASL     R4              ;RESTORE WORD
12368  070706  010437  067670                  MOV     R4,0.UPC        ;SET UP NEW PC
12369  070712  112737  000340  177776          MOVB    #0.STM,ST       ;SET HIGH PRIORITY
12370  070720  004537  071366                  JSR     5,0.RSTT        ;RESTORE TELETYPE
12371  070724  105037  071716          0.TBIT: CLRB    0.T             ;CLEAR BOTH
12372  070730  042737  000020  067672          BIC     #0.TBT,0.UST    ;  T-BIT FLAGS
12373  070736  017737  176742  067710          MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
12374  070744  013777  071760  176732          MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
12375  070752  012600                  0.GO2:  MOV     (SP)+,R0        ;RESTORE
12376  070754  012601                          MOV     (SP)+,R1        ; R0
```

```
12377  070756  012602                          MOV     (SP)+,R2        ;   THRU
12378  070760  012603                          MOV     (SP)+,R3
12379  070762  012604                          MOV     (SP)+,R4        ;
12380  070764  012605                          MOV     (SP)+,R5        ;     R5
12381  070766  012606                          MOV     (SP)+,SP        ; AND SP
12382  070770  013746  067672                  MOV     0.UST,-(SP)     ;     AND STATUS
12383  070774  013746  067670                  MOV     0.UPC,-(SP)     ;     AND PC
12384  071000  000006                  0.RTIT: RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
12385                                          ;
12386                                          ; PROCESS P - PROCEED
12387                                          ;   ONLY ALLOWED AFTER A BREAKPOINT
12388                                          ;
12389  071002  105737  071717          0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
12390  071006  001645                          BEQ     0.ERR1          ;NOT LEGAL
12391  071010  105037  071717                  CLRB    0.P             ;CLEAR PROCEED FLAG
12392  071014  005702                          TST     R2              ;WAS COUNT SPECIFIED?
12393  071016  001402                          BEQ     0.PR1           ;NO
12394  071020  010437  067706                  MOV     R4,0.CT         ;YES, PUT AWAY COUNT
12395  071024  112737  000340  177776  0.PR1:  MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
12396  071032  004537  071366                  JSR     5,0.RSTT        ;RESTORE TTY
12397  071036  112737  000340  177776  0.C1:   MOVB    #0.STM,ST       ;SET HIGH PRIORITY
12398  071044  105237  071716                  INCB    0.T             ;SET T-BIT FLAG
12399  071050  052737  000020  067672          BIS     #0.TBT,0.UST    ;SET T-BIT
12400  071056  000735                          BR      0.G02
12401                                          ;
12402                                          ; BREAKPOINT HANDLER
12403                                          ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
12404                                          ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
12405                                          ;  AND GIVES CONTROL TO THE CMD DECODER
12406                                          ;
12407  071060  012637  067670          0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
12408  071064  012637  067672                  MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
12409  071070  004037  071276          0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
12410  071074  105737  071716                  TSTB    0.T             ;CHECK FOR T-BIT SET
12411  071100  001311                          BNE     0.TBIT          ;JUMP IF SET
12412  071102  013777  067710  176574          MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
12413  071110  105737  067674                  TSTB    0.PRI           ;CHECK IF PRIORITY
12414  071114  100003                          BPL     0.BK2           ;  IS AS SAME AS USER PGM
12415  071116  113705  067672                  MOVB    0.UST,R5        ;PICK UP USER UST IF SO
12416  071122  000407                          BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
12417  071124  113705  067674          0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
12418  071130  000257                          CCC                     ;CLEAR CARRY
12419  071132  106005                          RORB    R5              ;SHIFT LOW ORDER BITS
12420  071134  106005                          RORB    R5              ;   INTO
12421  071136  106005                          RORB    R5              ;     HIGH ORDER
12422  071140  106005                          RORB    R5              ;       POSITION
12423  071142  110537  177776          0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
12424  071146  013705  067670                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
12425  071152  005745                          TST     -(R5)           ;SUBTRACT TWO
12426  071154  010537  067670                  MOV     R5,0.UPC        ;FROM THE USER'S PC
12427  071160  020537  067704                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
12428  071164  001417                          BEQ     0.B2            ;JUMP IF FOUND
12429  071166  004537  071334                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
12430  071172  004537  071670                  JSR     5,0.CRLF
12431  071176  012704  071722                  MOV     #0.BD,R4        ;ERR, NOTHING FOUND
12432  071202  012703  071723                  MOV     #0.BD+1,R3
```

```
12433  071206  004537  071562                 JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
12434  071212  010500                         MOV     R5,R0
12435  071214  042737  000020  067672         BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
12436  071222  000420                         BR      0.B3            ; AND CONTINUE
12437  071224  005337  067706         0.B2:   DEC     0.CT
12438  071230  003302                         BGT     0.C1            ;JUMP IF REPEAT
12439  071232  012737  000001  067706         MOV     #1,0.CT         ;RESET COUNT TO 1
12440  071240  105237  071717                 INCB    0.P             ;ALLOW PROCEED
12441  071244  004537  071334                 JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
12442  071250  012700  000102                 MOV     #'B,R0
12443  071254  004537  071576                 JSR     5,0.FTYP        ;TYPE "B"
12444  071260  013700  067704                 MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
12445  071264  004537  071434         0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
12446  071270  005005                         CLR     R5              ;CLEAR CAD
12447  071272  000137  070202                 JMP     0.DCD           ;GO TO DECODER
12448                                 ;
12449                                 ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
12450                                 ;
12451  071276  012637  071714         0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
12452  071302  010637  067666                 MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
12453  071306  012706  067666                 MOV     #0.USP,SP       ;SET TO INTERNAL STACK
12454  071312  010546                         MOV     R5,-(SP)        ;SAVE
12455  071314  010446                         MOV     R4,-(SP)        ; REGISTERS
12456  071316  010346                         MOV     R3,-(SP)        ;1
12457  071320  010246                         MOV     R2,-(SP)        ; THRU
12458  071322  010146                         MOV     R1,-(SP)        ;    5
12459  071324  013746  071714                 MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
12460  071330  005746                         TST     -(SP)
12461  071332  000200                         RTS     R0
12462                                 ;
12463                                 ; SAVE TELETYPE STATUS
12464                                 ;
12465  071334  113737  177560  071720 0.SVTT:  MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
12466  071342  113737  177564  071721         MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
12467  071350  105037  177560                 CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
12468  071354  105037  177564                 CLRB    0.TCSR          ;  BITS IN BOTH C/SR
12469  071360  004537  071670                 JSR     5,0.CRLF        ;TYPE <CR,LF>
12470  071364  000205                         RTS     R5
12471                                 ;
12472                                 ; RESTORE TELETYPE STATUS
12473                                 ;
12474  071366  004537  071670         0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
12475  071372  105737  177564                 TSTB    0.TCSR          ;WAIT READY ON PRINTER
12476  071376  100375                         BPL     .-4
12477  071400  032737  004000  177560         BIT     #4000,0.RCSR    ;CHECK BUSY FLAG ON READER
12478  071406  001403                         BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
12479  071410  105737  177560                 TSTB    0.RCSR          ;WAIT READY
12480  071414  100375                         BPL     .-4             ;    ON READER
12481  071416  113737  071720  177560 0.RSE1: MOVB    0.CSR1,0.RCSR   ;RESTORE
12482  071424  113737  071721  177564         MOVB    0.CSR2,0.TCSR   ;  THE STATUS REGISTERS
12483  071432  000205                         RTS     R5
12484                                 ;
12485                                 ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
12486                                 ; WORD IS IN R0
12487                                 ;
12488  071434  010246                 0.CADV: MOV     R2,-(SP)        ;SAVE R2
```

```
12489  071436  012704  071757          MOV     #O.BUF+6,R4     ;BUFFER START ADDRESS
12490  071442  012746  000060          MOV     #'0,-(SP)               ;CONSTANT ASCII 0
12491  071446  010002          O.SPC:  MOV     R0,R2           ; GET
12492  071450  042702  177770          BIC     #177770,R2      ;    OCTAL CHARACTER
12493  071454  061602                  ADD     @SP,R2          ;CONVERT TO ASCII
12494  071456  110244                  MOVB    R2,-(R4)        ;STORE IN BUFFER
12495  071460  006200                  ASR     R0              ;SHIFT THIS MESS
12496  071462  006200                  ASR     R0              ; RIGHT
12497  071464  006200                  ASR     R0              ;   THREE WHOLE PLACES
12498  071466  020427  071752          CMP     R4,#O.BUF+1     ;DONE?
12499  071472  101365                  BHI     O.SPC           ;    NO
12500  071474  042700  177776          BIC     #177776,R0      ;GET LAST BIT
12501  071500  062600                  ADD     (SP)+,R0                ;CONVERT TO ASCII
12502  071502  110044                  MOVB    R0,-(R4)        ;AND PUT IT AWAY
12503  071504  012703  071757          MOV     #O.BUF+6,R3     ;LWA
12504  071510  004537  071562          JSR     5,O.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
12505  071514  012602                  MOV     (SP)+,R2        ;RESTORE R2
12506  071516  000205                  RTS     R5
12507                          ;
12508                          ; GENERAL CHARACTER INPUT ROUTINE
12509                          ; CHARACTER INPUT GOES TO R0
12510                          ;
12511  071520  105737  177560  O.GET:  TSTB    O.RCSR          ;WAIT FOR
12512  071524  100375                  BPL     .-4             ;    INPUT FROM KEYBOARD
12513  071526  113700  177562          MOVB    O.RDB,R0        ;GET A CHARACTER
12514  071532  004537  071576          JSR     5,O.FTYP        ;ECHO CHARACTER
12515  071536  042700  177600          BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
12516  071542  001766                  BEQ     O.GET           ;IGNORE NULLS
12517  071544  122700  000040          CMPB    #40,R0          ;CHECK FOR SPACES
12518  071550  001763                  BEQ     O.GET           ;IGNORE NULLS
12519  071552  122700  000073          CMPB    #';,R0          ;CHECK FOR SEMI-COLON
12520  071556  001760                  BEQ     O.GET           ;IGNORE THEM IF FOUND
12521  071560  000205                  RTS     R5
12522                          ;
12523                          ; GENERAL CHARACTER OUTPUT ROUTINE
12524                          ;  ADDRESS OF FIRST BYTE IN R4,
12525                          ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
12526                          ;
12527  071562  020304  O.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
12528  071564  103426                  BLO     O.TYP1          ;  EXIT WHEN DONE
12529  071566  112400                  MOVB    (R4)+,R0        ;GET A CHARACTER
12530  071570  004537  071576          JSR     5,O.FTYP        ;TYPE ONE CHARACTER
12531  071574  000772                  BR      O.TYPE          ;LOOP UNTIL DONE
12532                          ;
12533                          ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
12534                          ;
12535  071576  105737  177564  O.FTYP: TSTB    O.TCSR          ;CHECK STATUS
12536  071602  100375                  BPL     .-4             ;WAIT UNTIL READY
12537  071604  110037  177566          MOVB    R0,O.TDB        ;TYPE ONE CHARACTER
12538  071610  120037  000045          CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
12539  071614  001012                  BNE     O.TYP1          ;NO
12540  071616  113746  000044          MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
12541  071622  105737  177564  O.TYP2: TSTB    O.TCSR
12542  071626  100375                  BPL     O.TYP2
12543  071630  105037  177566          CLRB    O.TDB           ;GENERATE NULL FILLER
12544  071634  105316                  DECB    @SP
```

```
12545  071636  003371                          BGT     0.TYP2
12546  071640  005726                          TST     (SP)+           ;POP STACK
12547  071642  000205                  0.TYP1: RTS     R5
12548                                          ;
12549                                          ; CLOSE WORD OR BYTE AND EXIT,
12550                                          ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
12551                                          ;
12552  071644  006205                  0.TCLS: ASR     R5              ;GET LOW ORDER BIT
12553  071646  103405                          BCS     0.TC            ;JUMP IF ALREADY CLOSED
12554  071650  006305                          ASL     R5
12555  071652  005702                          TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
12556  071654  001401                          BEQ     0.CLS1          ;NO CHANGE TO THE OPEN CELL
12557  071656  010415                          MOV     R4,@R5          ;STORE WORD
12558  071660  000207                  0.CLS1: RTS     PC
12559  071662  005746                  0.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
12560  071664  000137  070166                  JMP     0.ERR           ;AND SCREAM BLOODY MURDER
12561                                          ;
12562                                          ; 0.CRLF - TYPE <CR,LF>
12563                                          ; 0.CRLS - TYPE <CR,LF>*
12564                                          ;
12565  071670  012703  071725          0.CRLF: MOV     #0.CR+1,R3      ;LWA <CR,LF>
12566  071674  000402                          BR      0.CRS
12567  071676  012703  071726          0.CRLS: MOV     #0.CR+2,R3      ;LWA <CR,LF>*
12568  071702  012704  071724          0.CRS:  MOV     #0.CR,R4        ;FWA
12569  071706  004537  071562                  JSR     5,0.TYPE        ;TYPE SOMETHING
12570  071712  000205                          RTS     R5
12571                                          ;
12572  071714  000000                  0.XXX:  .WORD   0               ;TEMPORARY STORAGE
12573  071716     000                  0.T:    .BYTE   0               ;  T-BIT FLAG
12574  071717     000                  0.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
12575                                          ;                               = 1 IF PROCEED ALLOWED
12576  071720     000                  0.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
12577  071721     000                  0.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
12578                                          ;
12579                                          .EVEN
12580  071722  042502                  0.BD:   .WORD   "BE
12581                                          ;
12582  071724     015                  0.CR:   .BYTE   015     ;  <CR>
12583  071725     012                          .BYTE   012     ;  <LF>
12584  071726     052                          .BYTE   '*      ;  *
12585                                          ;
12586  071727     057                  0.LGCH: .BYTE   '/      ;  /
12587  071730     015                          .BYTE   015     ;  CARRIAGE RETURN
12588  071731     044                          .BYTE   '$      ;  $
12589  071732     107                          .BYTE   'G      ;  G
12590  071733     012                          .BYTE   012     ;  <LF>
12591  071734     137                          .BYTE   '←      ;  ←
12592  071735     136                          .BYTE   '↑      ;  ↑
12593  071736     117                          .BYTE   'O      ;  O
12594  071737     127                          .BYTE   'W      ;  W
12595  071740     105                          .BYTE   'E      ;  E
12596  071741     102                          .BYTE   'B      ;  B
12597  071742     120                          .BYTE   'P      ;  P
12598             000014            0.CLGT  =       .-0.LGCH                ;TABLE LENGTH
12599                                          ;
12600  071743     123                  0.TL:   .BYTE   'S      ;DO      1
```

```
12601  071744      120                    .BYTE   'P      ;NOT     2
12602  071745      115                    .BYTE   'M      ;CHANGE  3
12603  071746      000                    .BYTE   0       ;THE     4
12604  071747      000                    .BYTE   0       ;ORDER   5
12605  071750      102                    .BYTE   'B      ;HERE    6
12606          000006            O.LG    =       .-O.TL
12607                            ;
12608  071751                    O.BUF:                          ;6 CHAR. BUFFER WITH
12609          071757            .       =       .+6
12610  071757      040                    .BYTE                   ;TRAILING BLANK
12611                                     .EVEN
12612                            ;
12613  071760  000003            O.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
12614                            ;
12615                            ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
12616                            ;
12617          067652                    =       O.ODT-40
12618  067652  000000            O.URO:  0       ;USER R0
12619  067654  000000                    0       ;       R1
12620  067656  000000                    0       ;       R2
12621  067660  000000                    0       ;       R3
12622  067662  000000                    0       ;       R4
12623  067664  000000                    0       ;       R5
12624  067666  000000            O.USP:  0       ;USER SP
12625  067670  000000            O.UPC:  0       ;USER PC
12626  067672  000000            O.UST:  0       ;USER ST
12627  067674  000007            O.PRI:  7       ;ODT PRIORITY
12628  067676  000000            O.MSK:  0       ;MASK
12629  067700  000000                    0       ;LOW LIMIT
12630  067702  000000                    0       ;HIGH LIMIT
12631                            ;
12632                            ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
12633                            ;   UIN = CONTENTS
12634                            ;
12635  067704  000000            O.ADR1: 0
12636  067706  000000            O.CT:   0
12637  067710  000000            O.UIN:  0
12638          000001                    .END
```

```
ABASE = 177440          2290    2331    2345#
ACDW1 = 000000          2290    2333
ACDW2 = 000000          2290    2334
ACLO  = 000010          1371#
ACPUOP= 000000          2290    2305
ACT11   003450          2516#   3808*
ADDW0 = 000000          2290    2335
ADDW1 = 000000          2290    2336
ADDW10= 000000          2290
ADDW11= 000000          2290
ADDW12= 000000          2290
ADDW13= 000000          2290
ADDW14= 000000          2290
ADDW15= 000000          2290
ADDW2 = 000000          2290    2337
ADDW3 = 000000          2290    2338
ADDW4 = 000000          2290    2339
ADDW5 = 000000          2290    2340
ADDW6 = 000000          2290    2341
ADDW7 = 000000          2290    2342
ADDW8 = 000000          2290
ADDW9 = 000000          2290
ADEVCT= 000000          2290    2296
ADEVM = 000000          2290    2332
AENV  = 000000          2290    2301
AENVM = 000000          2290    2302
AFATAL= 000000          2290    2293
AMADR1= 000000          2290    2318
AMADR2= 000000          2290    2322
AMADR3= 000000          2290    2325
AMADR4= 000000          2290    2328
AMAMS1= 000000          2290    2312
AMAMS2= 000000          2290    2320
AMAMS3= 000000          2290    2323
AMAMS4= 000000          2290    2326
AMSGAD= 000000          2290    2298
AMSGLG= 000000          2290    2299
AMSGTY= 000000          2290    2292
AMTYP1= 000000          2290    2313
AMTYP2= 000000          2290    2321
AMTYP3= 000000          2290    2324
AMTYP4= 000000          2290    2327
APASS = 000000          2290    2295
APRIOR= 000000          2290
APTCSU= 000040         10088   10260#
APTENV= 000001         10035   10081   10216   10258#
APTSIZ= 000200          3737   10257#
APTSPO= 000100         10083   10218   10259#
ASWREG= 000000          2290    2303
ATESTN= 000000          2290    2294
ATTN    003324          2448#   5231    9049    9068    9094
AUNIT = 000000          2290    2297
AUSWR = 000000          2290    2304
AVECT1= 000000          2290    2329
AVECT2= 000000          2290    2330
BADHDR  003320          2439#   3816*   5621*   7287*   9796    9848*
```

```
BADTMO  050146          3889    9863#
BAI     = 000020        1334#   5630    5779
BA16    = 000400        1320#
BA17    = 001000        1321#
BIT0    = 000001        1263#   1317    1349    1368    2506    3926    9718
BIT00   = 000001        1253#   1263
BIT01   = 000002        1252#   1262
BIT02   = 000004        1251#   1261
BIT03   = 000010        1250#   1260
BIT04   = 000020        1249#   1259
BIT05   = 000040        1248#   1258
BIT06   = 000100        1247#   1257
BIT07   = 000200        1246#   1256
BIT08   = 000400        1245#   1255    9970
BIT09   = 001000        1244#   1254    9978    10046
BIT1    = 000002        1262#   1350    2507
BIT10   = 002000        1243#   1322    1340    1359    1391    1405    1419    1432    1446    5207    10023
BIT11   = 004000        1242#   1323    1341    1360    1377    1392    1406    1420    1433    1447    5209    9985
BIT12   = 010000        1241#   1324    1342    1361    1393    1407    1421    1434    1448
BIT13   = 020000        1240#   1325    1343    1362    1378    1394    1408    1422    1435    1449    10030
BIT14   = 040000        1239#   1326    1344    1363    1379    1395    1409    1423    1436    1450    1463    9555    9587
                        9956
BIT15   = 100000        1238#   1327    1328    1345    1364    1380    1396    1466    4726    9555    9583
BIT2    = 000004        1261#   1351    1370    2508
BIT3    = 000010        1260#   1333    1352    1371
BIT4    = 000020        1259#   1334    1353    1372    1385    1413    1440    6506
BIT5    = 000040        1258#   1335    1354    1373    1386    1400    1414    1427    1441
BIT6    = 000100        1257#   1318    1336    1355    1374    1387    1401    1415    1428    1442
BIT7    = 000200        1256#   1319    1337    1356    1375    1388    1402    1416    1429    1443    3916
BIT8    = 000400        1255#   1320    1338    1357    1376    1389    1403    1417    1430    1444
BIT9    = 001000        1254#   1321    1339    1358    1390    1404    1418    1431    1445    5203
BPTVEC= 000014          1270#
BSE     = 000200        1356#
BSERR   001456          2413#   6787*   6793*   7065
BSE22H  002304          2427#   6738    6790    6809    9581
BSE22S  054472          6800    9585    10884#
BYP     044320          3996    4019    4023    4038    4090    4098    4116    4120    4124    4128    9121#
BYPCER  001462          2416#   3914*   4140*   4810*   4842*   5423*   5503*   6835*   6915*   8768*   9237
BYPFMT  001464          2417#   5911*   6003*   6234*   6326*   9553
BYPT16  001340          2361#   3650*   3655*   3660*   3665*   3670*   3674*   4863
BYT16   007026          1480    3654#
BYT16A  007046          1484    3659#
CALADD  001366          2374#   5917*   6240*   7077*   7108*   7130    7216    7234    7272    7279*   7290*   9549    9559
                        9561    9609
CALDIF  001360          2371#   4965*   4981    5013*   5044    6029*   6351*   6549*   7112*   7681*   8024*   8382*   8746*
                        11776   11800
CCLR    = 100000        1328#   4513    4542    4570    4583    4656    4736    4777    4882    4923    4973    4987    5021
                        5050    5102    5143    5246    5253    5270    5335    5446    5478    5541    6085    6357    6411
                        6611    6858    6890    6953    7190    7510    7731    7859    8074    8202    8432    8560    8789
                        8796    8813    9278    9288    9298    9308    9800    9818
CCYL    001354          2369#   4872*   4959*   4960    4966    5007*   5008    5014    7810*   8153*   8511*
CDT     = 002000        1322#   4404
CERR    = 100000        1327#   3976    4076    4367    4417    4464    5310    5646    5699    5789    5835    5929    5966
                        6060    6132    6177    6252    6289    6383    6447    6490    6586    6671    6750    7000    7094
                        7165    7242    7306    7370    7384    7485    7557    7608    7706    7777    7834    7905    7954
                        8049    8120    8177    8248    8297    8360    8407    8478    8535    8606    8655    8728    9239
```

```
                    9334
CFMT   = 010000     1324#  5784   5796   5831   5854
CHKFLG  001466      2421#  9131*  9213   9217   9221
CHKMSG  044334      4251   4504   4527   4561   4605   4632   4762   4796   4827   4908   4942   5128   5162
                    5212   5323   5354   5439   5465   5494   5526   5560   5661   5726   5814   5872   5944
                    6042   6073   6104   6193   6267   6369   6396   6430   6508   6570   6599   6630   6762
                    6851   6877   6906   6938   6972   7144   7178   7209   7257   7469   7498   7529   7694
                    7719   7750   7821   7847   7878   8037   8062   8093   8164   8190   8221   8395   8420
                    8451   8522   8548   8579   8774   9131#
CKCERR  045066      8999   9003   9015   9020   9237#
CKSWR  = 104407     9955   10019  10045  10873#
CLEAR  = 000005     1305#  4544   4779   4925   4989   5052   5145   5255   5337   5448   5543   6087   6413
                    6613   6860   6955   7192   7512   7733   7861   8076   8204   8434   8562   8798   9820
CLKOF   047400      4582   4596   4623   4649   4698   4703   9689#
CLKON   047304      4573   4587   4614   4640   9667#
CLOCK   047344      3838   9678#
CLRFLG  043254      3758   8876#
COE    = 001000     1358#
COUNT   001372      2378#  4571*  4585*  4612*  4638*  9679*  9681*
CR     = 000015     1178#  10127  10137  10887  10892  10897  10901  10911  10920  10927  10935  10940  10945
                   10950  10956  10963  10967  10972  10977  10981  10985  10988  10993  10998  11002  11006
                   11012  11017  11020  11025  11029  11031  11034  11038  11043  11048  11058  11066  11074
                   11079  11083  11087  11090  11097  11114  11121  11124  11336  11406  11538  11542  11548
                   11555
CRLF   = 000200     1179#  10098  10137
CTO    = 004000     1323#  9243
CYLADD  001364      2373#  4261   4537   4772   4834   4918   4976   5024   5138   5222   5292   5361   5536
                    5672   5737   5882   6111   6112   6199   6406   6514   6637   6948   7216   7536   7757
                    7885   8100   8228   8458   8586   8783   9413*  9414*  9415*  9416*  9417*  9418*  9436*
                    9437*  9438*  9439*  9440*  9441*
CYLDIF  001362      2372#  4258   4534   4769   4838   4915   4981   5033   5039   5044   5135   5219   5329
                    5533   5669   5734   5879   6049   6079   6375   6403   6577   6605   6945   7151   7184
                    7476   7504   7725   7853   8068   8196   8426   8554   8780   9398*  9399*  9400*  9401*
                    9402*  9403*  9404   9406*  9423*  9424*  9425*  9426*  9427*  9428*  9429   9431*
CYLINV  042352      8007   8697#
DATAO   001444      2406#  5639   5750
DATAO1  001446      2407#
DATAI   001450      2408#  5780   5889
DCK    = 100000     1364#
DCLO   = 000020     1372#
DCPAR  = 020000     1325#  4442   4461
DDISP  = 177570     1185#  2265   3725
DDPCH   003446      2515#  3804*  3984   4169
DDT    = 000400     1376#  4390   4411
DDUMP   003444      2514#  3771*  8876
DF1     066712      2556   2562   2568   2574   2579   2585   2591   2597   2665   2671   2682   2687   2692
                    2697   2702   2707   2712   2717   2722   2727   2732   2842   2867   2892   2917   2977
                    2992   3047   3052   3057   3122   11863#
DF10    067036      2603   2609   2614   2619   2624   2660   2677   2787   2837   2952   2957   2962   2967
                    2972   2982   2987   2997   3002   3007   3012   3062   3082   3087   3102   3107   3112
                    3117   3147   3152   3157   3167   3242   3247   3282   3297   3302   3307   3312   3347
                    3357   3362   3383   3388   3453   3549   3554   3559   3574   3579   3589   3594   3599
                    3604   3609   11912#
DF12    067056      3067   11921#
DF14    067102      3212   11932#
DF15    067122      3177   3182   3202   3207   3317   11942#
```

```
DF16    067142          3162    3197    11951#
DF17    067162          3322    3337    11960#
DF2     066726          3634    3639    3645    11870#
DF20    067206          2629    2634    2639    2644    2767    2772    2797    2802    2807    2812    2817    2822    2827
                        2832    2847    2852    2857    2862    2872    2877    2882    2887    2897    2902    2907    2912
                        2932    2937    2942    2947    3017    3022    3027    3032    3127    3132    3137    3142    3217
                        3222    3227    3232    3257    3262    3267    3272    3287    3292    3413    3418    3423    3428
                        3468    3473    3478    3483    3488    3493    3498    3503    3508    3513    3518    3523    3528
                        3533    3614    3619    11970#
DF21    067242          3564    3569    11984#
DF22    067262          2649    2654    2737    2742    2757    2762    2922    2927    3037    3042    3072    3077    3172
                        3187    3192    3463    3538    3543    3624    3629    11992#
DF23    067316          2777    2782    2792    12007#
DF3     066752          3352    11881#
DF4     066756          3584    11883#
DF5     066772          3332    11890#
DF6     067012          2747    2752    3237    3252    3277    3367    3372    3378    3393    3398    3403    3408    3433
                        3438    3458    11899#
DH1     063736          2554    2560    2566    2572    2577    2583    2589    2595    2663    2669    2680    2685    2690
                        2695    2700    2705    2710    2715    2720    2725    2730    2840    2865    2890    2915    2975
                        2990    3045    3050    3055    3120    3210    3562    3567    3582    11563#  11874   11892   11901
                        11914   11923   11944   11953   11964   11972   11994   12009
DH10    064217          2815    2820    2825    2830    2835    3185    3190    11596#
DH11    064246          2845    2850    2855    2860    3597    11600#
DH12    064305          2870    2875    2880    2885    3602    11606#
DH13    064346          2895    2900    2905    2910    3607    11612#
DH14    064415          2735    2740    2745    2750    2920    2925    2970    2985    3070    3075    3365    3370    3456
                        11619#
DH16    064445          2780    2940    2945    3612    3617    3622    3627    11624#
DH17    064507          2755    2760    2775    2785    2980    3105    3280    3285    3290    3295    3300    3305    3360
                        3506    3511    11630#
DH18    064527          2601    2607    2612    2617    2622    3587    11633#
DH19    064550          2675    2950    11636#
DH2     063753          11566#  11865   11876   11885   11894   11905   11916   11927   11936   11946   11955   11966   11980
                        11988   12002   12017
DH20    064567          2955    11639#
DH21    064616          2658    2960    2995    3000    11643#
DH22    064644          3085    3100    3355    3451    3466    3471    3496    3501    11647#
DH23    064672          3350    11651#
DH25    064713          3005    3010    3015    3020    3025    3030    3035    3040    3235    3240    3245    3330    3376
                        11654#
DH26    064732          2765    2770    2930    2935    3310    3315    11657#
DH28    064756          11661#  11974   11996   12011
DH29    064766          11663#  11978   12000   12015
DH3     064023          11573#  11867   11878   11887   11896   11907   11918   11929   11938   11948   11957   11968   11982
                        11990   12004   12019
DH30    064776          3165    3170    3175    3180    3461    3486    3491    3526    3531    3577    11665#
DH32    065024          11669#  11976   11998   12013
DH33    065051          3215    3220    3225    3230    3250    11673#
DH34    065071          3060    11676#
DH35    065115          3065    11680#
DH36    065143          11684#  11925
DH37    065166          3080    11688#
DH38    065232          3125    3130    3135    3140    3145    3150    3155    3547    11694#
DH39    065250          3200    3205    3476    3481    3516    3521    3536    3541    3572    11697#
DH40    065277          11701#  11934
```

```
DH41     065333          3255    3260    3265    3270    3275    3381    3557   11706#
DH42     065354          3335   11709#
DH44     065414          3345   11715#  11962
DH45     065464          3160   11722#
DH46     065543          3195   11730#
DH48     065610          3386    3391    3396    3401    3406    3411    3416    3421    3426    3431    3436   11737#
DH49     065642         11742#  11872
DH51     065716          3110    3115   11750#
DH52     065745         11754#  11986
DH53     065775          3320   11759#
DH6      064074         11580#  11903
DH8      064127          2627    2632    2637    2642    2647    2652    2790   11585#
DH9      064153          2795    2800    2805    2810    2965    3552    3592   11589#
DI     = 040000          1326#
DISPLA   001142          2265#   3725*   3733*  10000*  10022*
DISPRE   000174          1475#   3733
DLT    = 100000          1345#   5710    5846    5976    6143    6299    6458    7395    7568    7788    7916    8131    8259
                         8489    8617
DLY      044302          9112#   9115
DMD    = 000040          1386#
DOSEEK   036424          8006    8008#
DOTIM    003502          2534#   3837*   3846*   4510
DRA    = 000001          1368#   4013    4106
DRDY   = 000200          1375#
DRIVS    003454          2518#   3921*   3928*   3945    3960*   3986*   4003    4094*   4158    8839    8931*
DRIV0    003456          2523#   3819    3923    3962    4062    8914
DRIV1    003460          2524#
DRIV2    003462          2525#
DRIV3    003464          2526#
DRIV4    003466          2527#
DRIV5    003470          2528#
DRIV6    003472          2529#
DRIV7    003474          2530#
DROT   = 000040          1373#
DRPAR  = 000010          1352#
DRVMSK= 000007           1332#   3979    4079
DRVPTR   001346          2365#   3819*   4163    4173*
DSC    = 040000          1379#
DSWR   = 177570          1184#   2264    3724
DTE    = 010000          1361#
DTYE   = 000040          1354#   4393    4414
DT1      066044          2555    2561    2567    2573    2578    2584    2590    2596    2602    2608    2613    2618    2623
                         2659    2664    2670    2676    2681    2686    2691    2696    2701    2706    2711    2716    2721
                         2726    2731    2786    2836    2841    2866    2891    2916    2951    2956    2961    2966    2971
                         2976    2981    2986    2991    2996    3001    3006    3011    3046    3051    3056    3061    3081
                         3086    3101    3106    3111    3116    3121    3146    3151    3156    3161    3166    3176    3181
                         3196    3201    3206    3241    3246    3281    3296    3301    3306    3311    3316    3321    3331
                         3336    3346    3356    3361    3382    3387    3452    3548    3553    3558    3573    3578    3583
                         3588    3593    3598    3603    3608    3633    3638    3644   11769#
DT10     066376          3568   11816#
DT11     066442          3351   11824#
DT13     066446          2628    2633    2638    2643    2766    2771    2796    2801    2806    2811    2816    2821    2826
                         2831    2846    2851    2856    2861    2871    2876    2881    2886    2896    2901    2906    2911
                         2931    2936    2941    2946    3016    3021    3026    3031    3126    3131    3136    3141    3216
                         3221    3226    3231    3256    3261    3266    3271    3286    3291    3412    3417    3422    3427
                         3467    3472    3477    3482    3487    3492    3497    3502    3507    3512    3517    3522    3527
```

```
                    3532    3613    3618   11825#
DT14     066526     2648    2653    2736    2741    2756    2761    2921    2926    3036    3041    3071    3076    3171
                    3186    3191    3462    3537    3542    3623    3628   11836#
DT15     066616     2776    2781    2791   11848#
DT4      066104     2746    2751    3236    3251    3276    3392    3397    3402    3407    3432    3437   11776#
DT6      066152     3066   11784#
DT7      066216     3211   11792#
DT8      066264     3366    3371    3377    3457   11800#
DT9      066332     3563   11808#
D.ACLO= 000100      1428#
D.BRHM= 000100      1415#   4245    4501    4521    4555    4602    4629    4756    4790    4821    4902    4936    5122
                    5156    5197    5317    5348    5436    5459    5491    5520    5554    5655    5720    5808    5866
                    5938    6039    6067    6098    6187    6261    6366    6390    6424    6501    6567    6593    6624
                    6756    6848    6871    6903    6932    6966    7141    7172    7203    7251    7466    7492    7523
                    7691    7713    7744    7818    7841    7872    8034    8056    8087    8161    8184    8215    8392
                    8414    8445    8519    8542    8573    8771
D.CART= 000400      1417#   4245    4501    4521    4555    4602    4629    4756    4790    4821    4902    4936    5122
                    5156    5197    5317    5348    5436    5459    5491    5520    5554    5655    5720    5808    5866
                    5938    6039    6067    6098    6187    6261    6366    6390    6424    6501    6567    6593    6624
                    6756    6848    6871    6903    6932    6966    7141    7172    7203    7251    7466    7492    7523
                    7691    7713    7744    7818    7841    7872    8034    8056    8087    8161    8184    8215    8392
                    8414    8445    8519    8542    8573    8771
D.DDT = 000400      1403#   4387    4408
D.DOOR= 000200      1416#   4245    4501    4521    4555    4602    4629    4756    4790    4821    4902    4936    5122
                    5156    5197    5317    5348    5436    5459    5491    5520    5554    5655    5720    5808    5866
                    5938    6039    6067    6098    6187    6261    6366    6390    6424    6501    6567    6593    6624
                    6756    6848    6871    6903    6932    6966    7141    7172    7203    7251    7466    7492    7523
                    7691    7713    7744    7818    7841    7872    8034    8056    8087    8161    8184    8215    8392
                    8414    8445    8519    8542    8573    8771
D.DRA = 000040      1400#   4240    4243    4499    4519    4553    4600    4627    4754    4788    4819    4900    4934
                    5120    5154    5191    5315    5346    5434    5457    5489    5518    5552    5653    5718    5806
                    5864    5936    6037    6065    6096    6185    6259    6364    6388    6422    6499    6565    6591
                    6622    6754    6846    6869    6901    6930    6964    7139    7170    7201    7249    7464    7490
                    7521    7689    7711    7742    7816    7839    7870    8032    8054    8085    8159    8182    8213
                    8390    8412    8443    8517    8540    8571    8769
D.DRDY= 000200      1402#   4519    4553    4643    4754    4788    4819    4900    4934    5120    5154    5191    5315
                    5346    5434    5457    5518    5552    5653    5718    5806    5864    5936    6065    6096    6185
                    6259    6388    6422    6499    6591    6622    6754    6946    6869    6930    6964    7170    7201
                    7249    7490    7521    7711    7742    7839    7870    8054    8085    8182    8213    8412    8443
                    8540    8571    8769
D.DROT= 020000      1435#
D.DSC = 040000      1409#   4519    4754    4819    4900    5120    5191    5315    5434    5489    5518    6065    6185
                    6388    6591    6846    6901    6930    7170    7490    7711    7839    8054    8182    8412    8540
                    8764    8769
D.FLT = 000200      1429#   4455    4820    5196    5435    5490    6847    6902    8758    8770
D.FORM= 001000      1404#   5800    5858
D.FWD = 002000      1419#   4576    4590    4617    4629    6039    6567    7141    7691    8034    8519
D.HDFL= 000200      1443#
D.HDHM= 000040      1414#   4245    4501    9487
D.IDAE= 000040      1427#   5196    5266    8755    8770    8809
D.ILF = 000400      1430#
D.LIMD= 020000      1449#   5492    6904    9470
D.LOAD= 010000      1421#   9504
D.MHD = 000400      1444#
D.NMOV= 010000      1448#   5492    6904
D.OFF = 002000      1405#
```

```
D.PAR = 001000          1431#  4439   4458   4820   5435   6847
D.PIP = 020000          1408#  4600   4627   5489   6037   6364   6565   6901   7139   7464   7689   7816   8032
                        8159   8390   8517   8761
D.PLO = 004000          1447#
D.REV = 004000          1420#  4602   5491   6366   6903   7466   7818   8161   8392
D.RTZ = 020000          1422#  4602   4629   4747   4893   5113   6360   6366   9811
D.SECT= 000020          1440#
D.SKI = 002000          1432#  5196   5490   6902   8770
D.SPIN= 010000          1407#  4491   4499   4519   4553   4600   4627   4659   4754   4788   4819   4900   4934
                        5120   5154   5191   5315   5346   5434   5457   5489   5518   5552   5653   5718   5806
                        5864   5936   6037   6065   6096   6185   6259   6364   6388   6422   6499   6565   6591
                        6622   6754   6846   6869   6901   6930   6964   7139   7170   7201   7249   7464   7490
                        7521   7689   7711   7742   7816   7839   7870   8032   8054   8085   8159   8182   8213
                        8390   8412   8443   8517   8540   8571   8769
D.SPLS= 010000          1434#
D.SPOK= 001000          1418#  4521   4555   4602   4629   4756   4790   4821   4902   4936   5122   5156   5197
                        5317   5348   5436   5459   5491   5520   5554   5655   5720   5808   5866   5938   6039
                        6067   6098   6187   6261   6366   6390   6424   6501   6567   6593   6624   6756   6848
                        6871   6903   6932   6966   7141   7172   7203   7251   7466   7492   7523   7691   7713
                        7744   7818   7841   7872   8034   8056   8087   8161   8184   8215   8392   8414   8445
                        8519   8542   8573   8771   9520
D.SSP = 000020          1413#  4521   4555   4602   4629   4756   4790   4821   4902   4936   5122   5156   5197
                        5317   5348   5436   5459   5491   5520   5554   5655   5720   5808   5866   5938   6039
                        6067   6098   6187   6261   6366   6390   6424   6501   6567   6593   6624   6756   6848
                        6871   6903   6932   6966   7141   7172   7203   7251   7466   7492   7523   7691   7713
                        7744   7818   7841   7872   8034   8056   8087   8161   8184   8215   8392   8414   8445
                        8519   8542   8573   8771
D.SUNS= 040000          1450#
D.TIB = 002000          1446#
D.UNLD= 040000          1423#  5483   5491   6895   6903
D.UNS = 040000          1436#
D.VV  = 000100          1401#  4238   4243   4323   4337   4499   4519   4553   4600   4627   4754   4788   4819
                        4900   4934   5120   5154   5191   5315   5346   5434   5457   5489   5518   5552   5653
                        5718   5806   5864   5936   6037   6065   6096   6185   6259   6364   6388   6422   6499
                        6565   6591   6622   6754   6846   6869   6901   6930   6964   7139   7170   7201   7249
                        7464   7490   7521   7689   7711   7742   7816   7839   7870   8032   8054   8085   8159
                        8182   8213   8390   8412   8443   8517   8540   8571   8769
D.WCUR= 000040          1441#
D.WGAT= 000100          1442#
D.WLE = 004000          1433#
D.WRL = 004000          1406#
D.XERR= 001000          1445#
ECCW  = 020000          1394#
ECH   = 000100          1355#
EMTVEC= 000030          1273#   3708*  3709*
EM1     057002          8940   11097#
EM10    057575          2588   11165#
EM11    057640          2594   11171#
EM12    057704          2600    2949   2954   2959   2964   2969   2979   3004   3084   3109   3164   3199   3309
                       11177#
EM13    057735          2606    2784   2834   2984   3009   3114   3144   3385  11182#
EM14    057757          2611   11186#
EM15    060004          2616   11190#
EM16    060036          2621    3405  11195#
EM17    060063          2626    2764   2794   2814   2844   2869   2894   2919   2939   3014   3124   3214   3254
                        3284    3410   3495   3515   3525  11199#
```

```
EM18    060100          2631    2769    2799    2819    2849    2874    2899    2924    2944    3019    3129    3219    3259
                        3415    3465    3475    3485    3505   11202#
EM19    060115          2636    2804    2824    2854    2879    2904    2929    3024    3069    3079    3134    3224    3264
                        3289    3420    3500    3520    3530    3611   11205#
EM2     057054          2553   11104#
EM20    060132          2641    2809    2829    2859    2884    2909    2934    3029    3074    3139    3229    3269    3425
                        3470    3480    3490    3510    3616   11208#
EM21    060147          2657    3179    3204    3239    3279    3314   11211#
EM22    060171          2662   11215#
EM23    060230          2668   11221#
EM24    060310          2674   11229#
EM25    060334          2679   11233#
EM26    060362          2684   11237#
EM27    060402          2689   11240#
EM28    060423          2694   11243#
EM29    060450          2699   11247#
EM3     057143          9867   11114#
EM30    060517          2704   11254#
EM31    060550          2709   11259#
EM32    060575          2714    3395   11263#
EM33    060622          2719   11267#
EM34    060657          2724   11272#
EM35    060710          2729   11277#
EM36    060763          2734    2744    3234    3329    3364    3375    3435   11285#
EM37    061023          2739   11291#
EM38    061063          2749    3249    3274    3369    3430   11297#
EM39    061134          2754    2994    3034    3169    3184    3294    3535   11304#
EM4     057211          3159   11121#
EM40    061171          2759    2999    3039    3189   11309#
EM41    061217          3460    3540   11313#
EM43    061260          2774   11319#
EM44    061307          2779   11323#
EM45    061343          2914   11328#
EM46    061464          2646    3621   11342#
EM47    061477          2651    3626   11344#
EM48    061512          3059    3064   11346#
EM49    061525          2889   11348#
EM5     057324          2559    3642   11135#
EM50    061613          2839   11358#
EM51    061664          2864   11365#
EM52    061764          3044   11376#
EM53    062023          3049   11382#
EM54    062054          3054   11387#
EM55    062107          3099   11392#
EM56    062142          3349   11397#
EM57    062165          3581   11401#
EM58    062221          3586   11406#
EM59    062267          3119   11413#
EM6     057345          2565   11138#
EM60    062320          3149   11418#
EM61    062354          3154   11423#
EM63    062410          3174   11428#
EM64    062431          2789   11431#
EM65    062460          3209   11435#
EM66    062500          3299   11438#
EM67    062534          3244   11443#
```

```
EM68     062573            3304    3359   11449#
EM69     062632            3344   11455#
EM7      057366            2571   11141#
EM70     062652            3591    3596    3601    3606   11458#
EM71     062665            3354   11460#
EM72     062706            3194   11463#
EM73     062733            3631   11467#
EM74     062754            3380   11470#
EM75     063001            3390   11474#
EM76     063027            3400   11478#
EM77     063050            3450   11481#
EM78     063076            3455   11485#
EM79     063163            3636   11494#
EM8      057455            2576   11151#
EM80     063204            3546   11497#
EM81     063234            3551   11501#
EM82     063262            3556   11505#
EM83     063307            3561    3566   11509#
EM84     063347            3571    3576   11515#
EM85     063377            3104   11520#
EM86     063430            3632    3637    3643   11525#
EM87     063515            3319    3334   11534#
EM88     063542            2974   11538#
EM89     063633            2989   11548#
EM9      057525            2582   11158#
ERRVEC=  000004            1266#   3722    3723#   3734#   3741#   3742#   3755#   3756#   3822#   3826#   3833#   3871#   3889#
                           9961    9962#   9964#   9967#
ESEC     001404            2384#   5384#   5385#   5395   11784
E.A0     003424            2494#   4240#   4243#   4499#   4519#   4553#   4600#   4627#   4754#   4788#   4819#   4900#   4934#
                           5120#   5154#   5191#   5315#   5346#   5434#   5457#   5489#   5518#   5552#   5653#   5718#   5806#
                           5864#   5936#   6037#   6065#   6096#   6185#   6259#   6364#   6388#   6422#   6499#   6565#   6591#
                           6622#   6754#   6846#   6869#   6901#   6930#   6964#   7139#   7170#   7201#   7249#   7464#   7490#
                           7521#   7689#   7711#   7742#   7816#   7839#   7870#   8032#   8054#   8085#   8159#   8182#   8213#
                           8390#   8412#   8443#   8517#   8540#   8571#   8769#   9135#   9142    9144#   9173   11825   11836
                          11848
E.A1     003430            2496#   4245#   4501#   4521#   4555#   4602#   4629#   4756#   4790#   4821#   4902#   4936#   5122#
                           5156#   5197#   5317#   5348#   5436#   5459#   5491#   5520#   5554#   5655#   5720#   5808#   5866#
                           5938#   6039#   6067#   6098#   6187#   6261#   6366#   6390#   6424#   6501#   6567#   6593#   6624#
                           6756#   6848#   6871#   6903#   6932#   6966#   7141#   7172#   7203#   7251#   7466#   7492#   7523#
                           7691#   7713#   7744#   7818#   7841#   7872#   8034#   8056#   8087#   8161#   8184#   8215#   8392#
                           8414#   8445#   8519#   8542#   8573#   8771#   9136#   9146    9148#   9193   11825   11836   11848
E.A2     003434            2498#   4247#   4523#   4557#   4758#   4792#   4823#   4904#   4938#   5124#   5158#   5199#   5314#
                           5350#   5461#   5522#   5556#   5657#   5722#   5810#   5868#   5940#   6069#   6100#   6189#   6263#
                           6392#   6426#   6503#   6595#   6626#   6758#   6873#   6934#   6968#   7174#   7205#   7253#   7494#
                           7525#   7715#   7746#   7843#   7874#   8058#   8089#   8186#   8217#   8416#   8447#   8544#   8575#
                           9137#   9150    9152#  11836   11848
E.A3     003440            2500#   9138#
E.B0     003426            2495#   4244#   4500#   4520#   4554#   4601#   4628#   4755#   4789#   4820#   4901#   4935#   5121#
                           5155#   5194#   5196#   5316#   5347#   5435#   5458#   5490#   5519#   5553#   5654#   5719#   5807#
                           5865#   5937#   6038#   6066#   6097#   6186#   6260#   6365#   6389#   6423#   6500#   6566#   6592#
                           6623#   6755#   6847#   6870#   6902#   6931#   6965#   7140#   7171#   7202#   7250#   7465#   7491#
                           7522#   7690#   7712#   7743#   7817#   7840#   7871#   8033#   8055#   8086#   8160#   8183#   8214#
                           8391#   8413#   8444#   8518#   8541#   8572#   8770#   9154    9156#   9183   11825   11836   11848
E.B1     003432            2497#   4246#   4502#   4522#   4556#   4603#   4630#   4757#   4791#   4822#   4903#   4937#   5123#
                           5157#   5198#   5318#   5349#   5437#   5460#   5492#   5521#   5555#   5656#   5721#   5809#   5867#
                           5939#   6040#   6068#   6099#   6188#   6262#   6367#   6391#   6425#   6502#   6568#   6594#   6625#
```

```
                    6757*  6849*  6872*  6904*  6933*  6967*  7142*  7173*  7204*  7252*  7467*  7493*  7524*
                    7692*  7714*  7745*  7819*  7842*  7873*  8035*  8057*  8088*  8162*  8185*  8216*  8393*
                    8415*  8446*  8520*  8543*  8574*  8772*  9158   9160*  9203   11825  11836  11848
E.B2    003436      2499#  4248*  4524*  4558*  4759*  4793*  4824*  4825*  4905*  4939*  5125*  5159*  5200*
                    5320*  5351*  5462*  5523*  5557*  5658*  5723*  5811*  5869*  5941*  6070*  6101*  6190*
                    6264*  6393*  6427*  6504*  6506*  6596*  6627*  6759*  6874*  6935*  6969*  7175*  7206*
                    7254*  7495*  7526*  7716*  7747*  7844*  7875*  8059*  8090*  8187*  8218*  8417*  8448*
                    8545*  8576*  9162   9164*  11836  11848
E.B3    003442      2501#  4249*  4525*  4559*  4760*  4794*  4906*  4940*  5126*  5160*  5203*  5207*  5209*
                    5321*  5352*  5463*  5524*  5558*  5659*  5724*  5812*  5870*  5942*  6071*  6102*  6191*
                    6265*  6394*  6428*  6505*  6597*  6628*  6760*  6875*  6936*  6970*  7176*  7207*  7255*
                    7496*  7527*  7717*  7748*  7845*  7876*  8060*  8091*  8188*  8219*  8418*  8449*  8546*
                    8577*  9166   9168*  11848
FATT1   044126      4515   4751   4897   5117   5272   5513   5586   6925   7032   8815   9065#  9791   9815
FATT2   044222      5308   6058   6175   6381   6488   6584   6669   6998   7163   7304   7368   7483   7606
                    7704   7832   7952   8047   8175   8295   8358   8405   8533   8653   8726   9092#
FHDHM   046434      5508   6920   9484#  9492
FHDTAB  046632      5920   6243   7080   7105   7228   9535#
FLGTST  047076      9582   9586   9602#
FLIM    046356      5480   6892   9466#
FLOAD   046510      5510   6922   9501#
FMTE  = 000020      1353#
FMTI    001440      2403#  9545*  9546*  9547*  9552
FNS22   045736      5382   9378#
FORM    031234      7042#  9830
FORMAT  001436      2402#  5919*  6242*  7079*  7110*  9545   9579
FRCYL   001350      2367#  4870*  5006   5007   5022   5024   5028   5067*  5068   5070   6027*  6349*  6547*
                    6643*  7113*  7280*  7291*  7447*  7581*  7667*  7678   7809   7810   7885   7897   7924
                    8010*  8021   8152   8153   8228   8240   8267   8368*  8379   8510   8511   8586   8598
                    8625   8736*  11776  11800  11816
FRDY    043612      3955   3969   3974   4069   4074   4204   4334   4406   4487   4546   4675   4742   4781
                    4815   4888   4927   4969   4991   5017   5054   5108   5147   5187   5250   5257   5304
                    5339   5430   5450   5476   5545   5582   5643   5697   5786   5798   5833   5856   5926
                    5964   6035   6089   6130   6171   6249   6287   6355   6415   6445   6484   6563   6615
                    6665   6747   6842   6862   6888   6957   6994   7028   7091   7137   7194   7239   7300
                    7364   7382   7462   7514   7555   7602   7687   7735   7775   7814   7863   7903   7948
                    8030   8078   8118   8157   8206   8246   8291   8354   8388   8436   8476   8515   8564
                    8604   8649   8722   8750   8793   8800   8994#  8997   9267   9329   9787   9806   9822
FRDY1   043660      9009#  9012   9283   9293   9303   9312
FSPOK   046556      4213   4684   9517#  9525
FSO22   045652      5378   9358#
FTITLE  001344      2363#  8887   8889*
GBA     043514      3790   8954#
GDRVS   043354      3788   8909#
GINT    043542      3792   8967#
GNS   = ****** U    1474   10865  10866  10867  10868  10869  10871  10873  10874  10875  10876  10877  10878
                    10879
GO    = 000001      1317#
GSTAT   045150      4237   4289   4322   4365   4454   4490   4575   4589   4616   4642   4658   4746   4892
                    5112   5265   5479   5645   5712   5788   5848   5928   5978   6145   6251   6301   6359
                    6460   6749   6891   7093   7241   7397   7570   7790   7918   8133   8261   8491   8619
                    8808   9075   9079   9099   9103   9263#  9333   9344   9397   9412   9469   9486   9503
                    9519   9810
GSTAT1  045212      9133   9276#
GTSWR = 104406      8898   10871#
HASOF   003352      2466#  9033*  9049   11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
```

```
HBA     003342  2462#  9029*  11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
HCS1    003334  2459#  3976   4076   4367   4417   4442   4461   4464   5310   5646   5699   5789   5835
                5929   5966   6060   6132   6177   6252   6289   6383   6447   6490   6586   6671   6750
                7000   7094   7165   7242   7306   7370   7384   7485   7557   7608   7706   7777   7834
                7905   7954   8049   8120   8177   8248   8297   8360   8407   8478   8535   8606   8655
                8728   9026*  9239   9243   9334   11770  11778  11786  11794  11802  11810  11818  11829
                11841  11854
HCS2    003336  2460#  4009   4011   4015   4027   4102   4104   4108   4111   4353   4370   9027*  9248
                9252   11770  11778  11786  11794  11802  11810  11818  11829  11841  11854
HDA     003344  2463#  9030*  11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
HDB     003356  2468#
HDC     003354  2467#  9034*  11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
HDS     003346  2464#  4013   4106   4390   4411   9031*  11770  11778  11786  11794  11802  11810  11818
                11829  11841  11854
HDTAB   001470  2423#  5631   5639*  5913   5990   6236   6313   7083   7232   9537
HDWD    001454  2411#  5751*  5752   5890*  5891   5992*  5994   6156*  6158   6315*  6317   11792
HD1     001434  2401#  9539*  9540*  9541*  9542*  9543*  9544*  9551
HEAD    001430  2399#  5622*  5633*  5634   5635*  5637   5677*  5678   5681*  5690*  5691   5692*  5741*
                5742   5743*  5748   5759*  5760   5918*  6241*  7078*  7085*  7086   7087*  7101*  7102
                7109*  7131*  7132   7133*  7229*  7230   7231*  7269*  7270   7276   7278*  7354*  7355
                9539   9614
HEADA   001432  2400#  4265   5170   5232   9456*  9457*  9458*  9459*
HER     003350  2465#  4393   4414   9032*  11770  11778  11786  11794  11802  11810  11818  11829  11841
                11854
HMR1    003360  2469#  9035*  11770  11778  11786  11794  11802  11810  11818  11829  11841  11854
HMR2    003362  2470#  3978   4078   4238   4290   4323   4337   4387   4408   4491   4576   4590   4617
                4643   4659   4747   4893   5113   5483   5800   5858   6360   6895   8761   8764   9013*
                9018*  9036*  9285   9295   9305   9314   9398   9487   9504   9520   9811   11770  11778
                11786  11794  11802  11810  11818  11829  11841  11854
HMR3    003364  2471#  4439   4455   4458   5266   8755   8758   8809   9014*  9019*  9037*  9286   9296
                9306   9315   9345   9413   9470   11770  11778  11786  11794  11802  11810  11818  11829
                11841  11854
HOLD    043746  8998   9002   9026#  9277
HPAT    003370  2473#  9039*  11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
HPEN    050126  7328   7408   9848#
HPEND   003322  2441#  3817*  7282   7286*  9798*  9849
HPOS    003366  2472#  9038*  11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
HT    = 000011  1176#  10096  10137
HVRC  = 000400  1357#
HWC     003340  2461#  9028*  11773  11781  11789  11797  11805  11813  11821  11832  11844  11857
HZ      001370  2376#  4571   4585   4612   4638   9681
H.A0    003404  2483#  9173   9314*  11825  11839  11851
H.A1    003410  2485#  9193   9305*  11825  11839  11851
H.A2    003414  2487#  9295*  9423   11839  11851
H.A3    003420  2489#  9285*
H.B0    003406  2484#  9183   9315*  11825  11839  11851
H.B1    003412  2486#  9203   9306*  11825  11839  11851
H.B2    003416  2488#  9296*  9436   11839  11851
H.B3    003422  2490#  9286*  9446   9456   11851
IDAE  = 002000  1359#
IE    = 000100  1318#
ILF   = 000001  1349#
INTER   050276  8982   9907#
INVCYL  003304  2431#  8745
IOTVEC= 000020  1271#  3706*  3707*
IR    = 000100  1336#
```

```
LCLKF   003476          2532#  3824*
LCVEC   001330          2353#  3825
LF    = 000012          1177#  10131  10137  10887  10892  10897  10901  10911  10920  10927  10935  10940  10945
                        10950  10956  10963  10967  10972  10977  10981  10985  10988  10993  10998  11002  11006
                        11012  11017  11020  11025  11029  11031  11034  11038  11043  11048  11058  11066  11074
                        11079  11083  11087  11090  11097  11114  11121  11124  11336  11406  11538  11542  11548
                        11555
LIMERR  001460          2414#  5615   7061   9466*  9474*
LKS     001326          2351#  3823   9670*  9691*
LPFLG   001410          2387#  4691   4699*  4704*  5422*  5590*  6205   6211*  6216*  6519   6524*  6529*  6687
                        6695*  6701*  6834*  7016   7036*  7322   7336   7342*  7348*  7416*  7421*  7624   7632*
                        7638*  7970   7976*  7981*  7986*  7991*  8313   8319*  8324*  8329*  8334*  8671   8677*
                        8682*  8687*  8692*  9750*
MCLK  = 000400          1389#
MDS   = 001000          1339#  4009   4102   9252
MDTST   007070          1488   3664#
MDTSTA  007110          1490   3669#
MEMBAS= 172100          2357#  3744   3753
MEMERR  050234          3747   9893#
MEMPAR  007442          3741#
MEMVEC= 000114          2356#  3747*  3748*
MERD  = 001000          1390#
MESMSK= 000017          1384#
MEWD  = 002000          1391#
MIND  = 000200          1388#
MODTST  001342          2362#  3651*  3656*  3661*  3666*  3671*  3675*  7057   7125   7155   7265   7443   8005
MSG1    054636          8890   10887#
MSG10   056030          3920   3959   11002#
MSG11   056057          9934   11006#
MSG12   056117          7073   11012#
MSG13   056150          3847   11017#
MSG14   056226          9121   11025#
MSG15   056251          4029   4174   11029#
MSG16   056263          4287   11031#
MSG17   056304          6808   11034#
MSG18   056333          5649   5702   5714   5792   5838   5850   5932   5969   5980   6135   6147   6255   6292
                        6303   6450   6462   7097   7245   7387   7399   7560   7572   7780   7792   7908   7920
                        8123   8135   8251   8263   8481   8493   8609   8621   11038#
MSG19   056367          4160   11043#
MSG2    055526          3772   10967#
MSG20   056417          7059   11048#
MSG21   056566          7445   11066#
MSG22   056646          7442   11074#
MSG3    055564          3787   10972#
MSG4    055616          3789   10977#
MSG5    055644          3791   10981#
MSG6    055671          9909   10985#
MSG7    055707          3806   10988#
MSG74   056676          9777   11079#
MSG75   056725          9779   11083#
MSG76   056747          9834   11087#
MSG77   056765          9839   11090#
MSG8    055744          4868   10993#
MSG9    056002          4865   10998#
MSP   = 000100          1387#
MULT6   047262          9632   9658#
```

```
M.ALGN= 040000          1463#
M.CADD= 017760          1462#   9414    9437
M.CDIF= 017760          1455#   9399    9424
M.DRV = 000007          1454#
M.HEAD= 007000          1465#   9457
M.ID  = 000003          1461#
M.OFST= 017760          1456#
M.PAR = 100000          1466#   9720
M.SECT= 000760          1464#   9346    9447
M.SER = 077770          1457#
NED   = 010000          1342#   4015    4027    4108    4111    9248
NEM   = 004000          1341#
NUDRV   011362          3947    4041    4140#   8841
NXF   = 000004          1351#
OFFSET= 000015          1309#
OFST  = 000004          1370#
OPI   = 020000          1362#
OR    = 000200          1337#
O.ADR1  067704          12150#  12201#  12203#  12373   12374#  12412#  12427   12444   12635#
O.BACK  070430          12245   12286#
O.BD    071722          12431   12432   12580#
O.BKPT  070136          12196#  12249
O.BK1   071070          12144   12409#
O.BK2   071124          12414   12417#
O.BK3   071142          12416   12423#
O.BRK   071060          12161   12407#
O.BUF   071751          12489   12498   12503   12608#
O.B2    071224          12428   12437#
O.B3    071264          12436   12445#
O.CADV  071434          12262   12279   12302   12306   12336   12340   12445   12488#
O.CLGL  070252          12219   12221   12229#
O.CLGT= 000014          12233   12598#
O.CLS1  071660          12556   12558#
O.CR    071724          12565   12567   12568   12582#
O.CRET  070364          12240   12269#
O.CRLF  071670          12277   12334   12430   12469   12474   12565#
O.CRLS  071676          12214   12567#
O.CRS   071702          12566   12568#
O.CSR1  071720          12465#  12481   12576#
O.CSR2  071721          12466#  12482   12577#
O.CT    067706          12394#  12437#  12439#  12636#
O.C1    071036          12397#  12438
O.DCD   070202          12202   12204   12214#  12271   12322   12447
O.DCD1  070206          12215#  12263   12307
O.EFF   070512          12248   12313#
O.EFF1  070630          12325   12344#
O.ENTR  067716          12141#
O.ERR   070166          12188   12199   12211#  12234   12259   12317   12560
O.ERR1  070522          12317#  12390
O.ERR2  070342          12259#  12293   12366
O.FTYP  071576          12213   12281   12296   12338   12443   12514   12530   12535#
O.GET   071520          12167   12217   12511#  12516   12518   12520
O.GO    070674          12242   12364#
O.GO2   070752          12375#  12400
O.LG  = 000006          12171   12606#
O.LGCH  071727          12230   12586#  12598
```

```
0.LGDR  070302          12237   12239#  12251
0.LGL =  000030         12251#
0.LGL1  070254          12230#  12235
0.LGL2  070274          12231   12236#
0.MSK   067676          12318   12319   12321   12628#
0.ODT   067712           1486   12139#  12617
0.OFST  070440          12246   12292#
0.OF1   070506          12305   12307#
0.OP1   070376          12243   12275#
0.OP2   070404          12192   12277#  12288
0.ORPC  070110          12184#  12244
0.P     071717          12159*  12364*  12389   12391*  12440*  12574#
0.PRI   067674          12151   12413   12417   12627#
0.PROC  071002          12250   12389#
0.PR1   071024          12393   12395#
0.RALL  070156          12162   12197   12203#
0.RCSR=  177560         12129#  12465   12467*  12477   12479   12481*  12511
0.RDB =  177562         12128#  12513
0.REGT  070040          12167#  12241
0.RSE1  071416          12478   12481#
0.RSP   070050          12169#  12172
0.RST   067754          12140   12149#
0.RSTT  071366          12370   12396   12474#
0.RST1  070004          12148   12156#
0.RTIT  071000          12158*  12384#
0.SCAN  070212          12178   12217#  12228
0.SP    070102          12170   12179#
0.SPC   071446          12491#  12499
0.SP1   070070          12175#  12180
0.STM =  000340         12118#  12160   12369   12395   12397
0.STRT  067742          12139   12146#
0.SVR   071276          12149   12409   12451#
0.SVTT  071334          12429   12441   12465#
0.T     071716          12371*  12398*  12410   12573#
0.TBIT  070724          12371#  12411
0.TBT =  000020         12119#  12372   12399   12435
0.TC    071662          12553   12559#
0.TCLS  071644          12184   12269   12275   12286   12552#
0.TCSR=  177564         12131#  12466   12468*  12475   12482*  12535   12541
0.TDB =  177566         12130#  12537*  12543*
0.TL    071743          12168   12171   12179   12600#  12606
0.TRTC  071760          12203   12374   12613#
0.TVEC=  000014         12117#  12142   12160*  12161*
0.TYPE  071562          12433   12504   12527#  12531   12569
0.TYP1  071642          12528   12539   12547#
0.TYP2  071622          12541#  12542   12545
0.UIN   067710          12150   12373*  12412   12637#
0.UPC   067670          12143*  12368*  12383   12407*  12424   12426*  12625#
0.URD   067652          12146   12176   12618#
0.USP   067666          12147*  12452*  12453   12624#
0.UST   067672          12141*  12372*  12382   12399*  12408*  12415   12435*  12626#
0.WDS   070520          12314   12316#
0.WDS2  070536          12321#  12343
0.WDS3  070566          12332#  12345   12351   12360
0.WDS4  070624          12332   12342#
0.WRD   070332          12239   12255#
```

```
O.WRDA  070356          12256  12264#
O.WRD1  070340          12258# 12265  12282
O.WSCH  070516          12247  12315#
O.XXX   071714          12451* 12459  12572#
O.45    070016          12157  12159#
PACK  = 000003           1304#  4332
PARAM   001336           2360#  3649*  3654*  3659*  3664*  3669*  3673*  3784
PARSRT  007006           1482   3649#
PAT   = 000020           1385#  4453   4811   4957   5005   5426   6838
PCA   = 004000           1392#
PCD   = 010000           1393#
PCLKF   003500           2533#  3828*  3835*  9668   9689
PCVEC   001332           2354#  3829   3836
PCYL    001356           2370#  4873*  4961   4966*  5009   5014*
PFSRT   011620           4220#  9936
PGE   = 002000           1340#
PIP   = 020000           1378#
PIRQ  = 177772           1183#
PIRQVE= 000240           1277#
PKRB    001324           2350#
PKS     001320           2348#  3827   3834   9673*  9693*
PKSB    001322           2349#  9672*
PPTP    003452           2517#  3775*
PRGSRT  007146           3652   3657   3662   3667   3672   3676#  8941
PR0   = 000000           1200#  3678   3866
PR1   = 000040           1201#
PR2   = 000100           1202#
PR3   = 000140           1203#
PR4   = 000200           1204#
PR5   = 000240           1205#  2347
PR6   = 000300           1206#  3839
PR7   = 000340           1207#  3742   3748   9931   9932
PS    = 177776           1180#  1181
PSEC    001402           2383#  5381*  5384   5386*  5388   5391   9381   9384   11784
PSW   = 177776           1181#
PWRVEC= 000024           1272#  3712*  3713*  9921*  9930*  9931*
RCYLA   046234           9219   9436#
RCYLD   046162           9215   9423#
RDCHR = 104410          10539  10874#
RDCYLA  046112           4975   5023   5291   9411#
RDCYLD  046026           4980   5032   5038   5043   9396#
RDDATA= 000021           1311#  6745
RDGATE= 100000           1396#
RDHEAD= 000025           1313#  5695   5831   5962   6128   6285   6443   7380   7553   7773   7901   8116   8244
                         8474   8602
RDLIN = 104411           8909  10613  10875#
RDOCT = 104412           8954   8967  10876#
RDSEC   045602           5387   5390   9343#  9360   9363   9380   9383   9630
RDY   = 000200           1319#  8994   9009
RECAL = 000013           1308#  4738   4884   5104   5248   6353   8791   9802
RESREG= 104414          10673  10743  10878#  12087
RESVEC= 000010           1267#
RHEAD   046330           9224   9456#
RHTAB   001674           2424#  5694   5745   5830   5886   5960   5983   6127   6152   6156   6283   6306   6442
                         6466   6470   7379   7403   7552   7576   7772   7796   7900   7924   8115   8139   8243
                         8267   8473   8497   8601   8625   9637   9639   9643   9646  11808  11816
```

```
RKASOF= 000016  1290#  3880   9033   9068   9094
RKBA  = 000004  1285#  3876   5631*  5780*  5913*  6236*  6739*  6782*  6801*  7083*  7232*  9029
RKCS1 = 000000  1283#  3873   3972*  4072*  4202*  4332*  4404*  4485*  4513*  4542*  4544*  4570*  4583*
                4656*  4673*  4736*  4738*  4777*  4779*  4813*  4882*  4884*  4923*  4925*  4967*  4973*
                4987*  4989*  5015*  5021*  5050*  5052*  5102*  5104*  5143*  5145*  5185*  5246*  5248*
                5253*  5255*  5270*  5302*  5335*  5337*  5428*  5446*  5448*  5474*  5478*  5541*  5543*
                5580*  5641*  5695*  5784*  5796*  5831*  5854*  5924*  5962*  6033*  6085*  6087*  6128*
                6169*  6247*  6285*  6353*  6357*  6411*  6413*  6443*  6482*  6561*  6611*  6613*  6663*
                6745*  6840*  6858*  6860*  6886*  6890*  6953*  6955*  6992*  7026*  7089*  7135*  7190*
                7192*  7237*  7298*  7362*  7380*  7460*  7510*  7512*  7553*  7600*  7685*  7731*  7733*
                7773*  7812*  7859*  7861*  7901*  7946*  8028*  8074*  8076*  8116*  8155*  8202*  8204*
                8244*  8289*  8352*  8386*  8432*  8434*  8474*  8513*  8560*  8562*  8602*  8647*  8720*
                8748*  8789*  8791*  8796*  8798*  8813*  8994   9009   9026   9265*  9278*  9281*  9288*
                9291*  9298*  9301*  9308*  9310*  9785*  9800*  9802*  9818*  9820*
RKCS2 = 000010  1287#  3874   3953*  3967*  3971*  4067*  4071*  4364*  4543*  4737*  4778*  4883*  4924*
                4988*  5051*  5103*  5144*  5247*  5254*  5336*  5447*  5542*  5630*  5710   5779*  5846
                5976   6086*  6143   6299   6412*  6458   6612*  6859*  6954*  7191*  7395   7511*  7568
                7732*  7788   7860*  7916   8075*  8131   8203*  8259   8433*  8489   8561*  8617   8790*
                8797*  9027   9264*  9279*  9289*  9299*  9309*  9327*  9331*  9801*  9819*
RKDA  = 000006  1286#  3877   5182*  5634*  5691*  5742*  6741*  6784*  6803*  7086*  7132*  7230*  7359*
                9030
RKDB  = 000024  1292#  3882   5705   5706   5707   5841   5842   5843   5972   5973   5974   6138   6139
                6140   6295   6296   6297   6453   6454   6455   7390   7391   7392   7563   7564   7565
                7783   7784   7785   7911   7912   7913   8126   8127   8128   8254   8255   8256   8484
                8485   8486   8612   8613   8614
RKDC  = 000020  1291#  3881   4812*  4958*  4974*  5006*  5022*  5427*  5473*  5922*  5958*  6030*  6125*
                6245*  6281*  6480*  6559*  6661*  6743*  6839*  6885*  6990*  7130*  7234*  7296*  7458*
                7549*  7598*  7683*  7769*  7809*  7897*  7944*  8026*  8112*  8152*  8240*  8287*  8350*
                8384*  8470*  8510*  8598*  8645*  8747*  9034
RKDS  = 000012  1288#  3878   9031
RKECPS= 000030  1296#  3886   9038
RKECPT= 000032  1297#  3887   9039
RKER  = 000014  1289#  3879   9032
RKMR1 = 000026  1293#  3883   4288*  4321*  4453*  4574*  4588*  4615*  4641*  4657*  4745*  4811*  4891*
                4957*  5005*  5111*  5426*  6358*  6838*  9035   9074*  9078*  9098*  9102*  9280*  9290*
                9300*  9332*  9343*  9396*  9411*  9468*  9485*  9502*  9518*  9809*
RKMR2 = 000034  1294#  3884   9013   9018   9036
RKMR3 = 000036  1295#  3885   9014   9019   9037
RKPRI   001316  2347#  8983
RKVEC   001314  2346#  3811*  8971*  8974*  8981
RKWC  = 000002  1284#  3875   5632*  5781*  5914*  6237*  6744*  7084*  7233*  9028
RLS   = 000010  1333#  4364
RSEC    046272  9223   9446#
RTT   = 000006  12121#
SAVREG= 104413  10658  10717  10877#  12031
SBPAR   047422  9143   9147   9151   9155   9159   9163   9167   9706#
SCLR  = 000040  1335#  3953   3967   4067   9327
SCOP1 = 104415  3964   4064   4326   4358   4398   4447   4729   4804   4875   4950   4999   5094   5175
                5296   5624   5682   5823   5952   6118   6275   6552   7119   7221   7451   7541   7671
                7762   7802   7890   8014   8105   8145   8233   8372   8463   8503   8591   8739   10879#
SCOP1S  047474  9730#  10879
SDC   = ****** U  1163   1485   12097
SEC     001374  2380#  4572*  4586*  4613*  4639*  9682*
SECFLG  047026  9557   9578#
SECNT   001400  2382#  5376*  5398*  5821*  5951*  6274*
SECTOR  001406  2385#  5386   5388   5391   5395   9345*  9346*  9347*  9348*  9349*  9350*  9361   9364
```

```
                    9381    9384    9446*   9447*   9448*   9449*   9450*   9451*   9631*   9635    9636*   9637*   9648
                    9658*   9659    9660*   9661*
SEEK   = 000017     1310#   4813    4967    5015    5185    5302    5428    5474    6033    6169    6482    6561    6663
                    6840    6886    6992    7135    7298    7362    7460    7600    7685    7812    7946    8028    8155
                    8289    8352    8386    8513    8647    8720    8748
SELDRV= 000001      1303#   3972    4072    4404    5796    5854    9265    9281    9291    9301    9310
SETINT  043574      3812    8972    8981#
SIZFLG  003504      2535#   3813*   3957    8927*   8947*
SKI    = 000002     1350#
SORT    047154      5986    6309    9628#
SPACE2  067626     12065   12074   12085   12096#
SRTSPL= 000011      1307#   4485    5580    7026    9785
SRTTAB  002100      2425#   5989    6312    9640
ST     = 177776    12115#  12141   12142*  12155*  12369*  12395*  12397*  12423*
STACK  = 001100     1171#   3677    3704    3864    3913    3965    4060    4065    4154    4195    4232    4280    4319
                    4327    4349    4359    4383    4399    4435    4448    4480    4723    4730    4805    4861    4876
                    4951    5000    5090    5095    5176    5287    5297    5370    5417    5614    5625    5683    5775
                    5824    5906    5953    6024    6119    6229    6276    6345    6543    6553    6730    6829    7056
                    7120    7222    7440    7452    7542    7665    7672    7763    7803    7891    8003    8015    8106
                    8146    8234    8344    8373    8464    8504    8592    8715    8740    8837    9229    9902    9933
                   12093
START   007132      1478    3673#
START1  007550      3766#
STKLMT= 177774      1182#
STOP    047576      7288    9768#  10380   10463
ST2     007616      3773    3784#
ST3     007662      3774    3802#
ST4     007714      3786    3807    3810#
ST5     007740      3793    3815#   3951    4007    8866    9841
SUBCLR  045522      4199    4209    4234    4284    4329    4351    4361    4385    4401    4437    4450    4482    4670
                    4680    4687    4732    4807    4878    4953    5002    5097    5178    5289    5299    5372    5419
                    5505    5577    5627    5685    5777    5826    5908    5955    6025    6121    6166    6181    6231
                    6278    6347    6437    6478    6494    6545    6555    6658    6675    6683    6732    6769    6831
                    6917    6987    7004    7012    7023    7070    7122    7224    7293    7310    7318    7374    7454
                    7544    7595    7612    7620    7674    7765    7805    7893    7941    7958    7966    8017    8108
                    8148    8236    8284    8301    8309    8347    8364    8375    8466    8506    8594    8642    8659
                    8667    8717    8732    8742    9327#   9770    9782
SVAL   = 100000     1380#
SWR     001140      2264#   3702    3724*   3726    3732*   3739*   4700    4705    5591    6212    6217    6525    6530
                    6696    6702    7037    7343    7349    7417    7422    7633    7639    7977    7982    7987    7992
                    8320    8325    8330    8335    8678    8683    8688    8693    8896    9179    9189    9199    9209
                    9730    9751    9756    9758    9865    9873    9878    9898    9956    9970    9972    9978    9985
                   10023   10030   10042   10046   10383   10422   10477*  12089
SWREG   000176      1476#   3732    8896   10383   10422   10445
SWTST   047526      4664    5567    6162    6474    6649    6979    7331    7411    7587    7935    8278    8636    9749#
SW0    = 000001     1235#
SW00   = 000001     1225#   1235
SW01   = 000002     1224#   1234
SW02   = 000004     1223#   1233
SW03   = 000010     1222#   1232
SW04   = 000020     1221#   1231
SW05   = 000040     1220#   1230
SW06   = 000100     1219#   1229
SW07   = 000200     1218#   1228
SW08   = 000400     1217#   1227
SW09   = 001000     1216#   1226
```

```
SW1    = 000002          1234#
SW10   = 002000          1215#
SW11   = 004000          1214#
SW12   = 010000          1213#  12089
SW13   = 020000          1212#   9865
SW14   = 040000          1211#   9751   9878
SW15   = 100000          1210#
SW2    = 000004          1233#
SW3    = 000010          1232#
SW4    = 000020          1231#
SW5    = 000040          1230#
SW6    = 000100          1229#
SW7    = 000200          1228#
SW8    = 000400          1227#   9756
SW9    = 001000          1226#   4700   4705   5591   6212   6217   6525   6530   6696   6702   7037   7343   7349
                         7417    7422   7633   7639   7977   7982   7987   7992   8320   8325   8330   8335   8678
                         8683    8688   8693   9179   9189   9199   9209   9730   9873   9898

TBITVE= 000014           1268#
TEMP1  003372            2475#   3954*  3968*  3973*  3978*  3979*  3980   4068*  4073*  4078*  4079*  4080   4203*
                         4333*   4405*  4486*  4545*  4674*  4741*  4780*  4814*  4887*  4926*  4968*  4990*  5016*
                         5053*   5107*  5146*  5186*  5230*  5231*  5232   5249*  5256*  5303*  5306*  5338*  5429*
                         5449*   5475*  5544*  5581*  5642*  5696*  5750*  5752   5785*  5797*  5832*  5855*  5889*
                         5891    5925*  5963*  5993*  5994   6034*  6055*  6088*  6129*  6157*  6158   6170*  6174*
                         6248*   6286*  6316*  6317   6354*  6380*  6414*  6444*  6483*  6487*  6562*  6582*  6614*
                         6664*   6668*  6746*  6841*  6861*  6887*  6956*  6993*  6997*  7027*  7090*  7136*  7161*
                         7193*   7238*  7299*  7303*  7363*  7367*  7381*  7461*  7481*  7513*  7554*  7601*  7605*
                         7686*   7702*  7734*  7774*  7813*  7830*  7862*  7902*  7947*  7951*  8029*  8045*  8077*
                         8117*   8156*  8173*  8205*  8245*  8290*  8294*  8353*  8357*  8387*  8403*  8435*  8475*
                         8514*   8531*  8563*  8603*  8648*  8652*  8721*  8725*  8749*  8792*  8799*  8996*  9011*
                         9066*   9070*  9096*  9112   9114*  9140   9142*  9144   9146*  9148   9150*  9152   9154*
                         9156    9158*  9160   9162*  9164   9166*  9168   9170*  9263   9266*  9269*  9276   9282*
                         9292*   9302*  9311*  9317*  9328*  9358   9359*  9366*  9368*  9370*  9378   9379*  9386*
                         9388*   9390*  9467*  9472*  9484*  9489*  9501*  9506*  9517*  9522*  9559*  9562*  9563
                         9712*   9720*  9786*  9805*  9821*  11792

TEMP2  003374            2476#   4212*  4514*  4683*  4750*  4896*  5116*  5271*  5507*  5512*  5585*  6734*  6772*
                         6773    6799*  6919*  6924*  7031*  8814*  9072*  9491*  9524*  9560*  9561*  9563*  9564
                         9790*   9814*

TEMP3  003376            2477#   4960*  4962   5008*  5010   6735*  6775   6796*  6797   7678*  7680   8021*  8023
                         8379*   8381*  8382

TEMP4  003400            2478#   4961*  4962*  4964*  4965   5009*  5010*  5012*  5013   6738*  6739   6782   6800*
                         6801    7679*  7680*  7681   8022*  8023*  8024   8380*  8381

TEMP5  003402            2479#   4726*  4843*  6740*  6741   6783*  6784   6802*  6803
TIMUP  001376            2381#   4578   4592   4619   4645   9667*  9678*  9684*
TITLE  043274            3768    3805   8887#
TKVEC = 000060           1275#  10360* 10361*
TOCYL  001352            2368#   4871*  4958   4959   4974   4976   5028   5030   5062*  5063   5070*  5915*  5917
                         6028*   6238*  6240   6350*  6548*  6559   6637   6641   6644*  6661   6678   6680*  6990
                         7007    7009*  7075*  7077   7114*  7281*  7296   7313   7315*  7360*  7403   7448*  7458
                         7536    7549   7576   7583*  7598   7615   7617*  7668*  7679   7683   7757   7769   7796
                         7930    7932*  7944   7961   7963*  8011*  8022   8026   8100   8112   8139   8273   8275*
                         8287    8304   8306*  8369*  8380   8384   8458   8470   8497   8631   8633*  8645   8662
                         8664*   8745*  8746   8747   8780   8783  11776  11800  11808

TPVEC = 000064           1276#
TRAPPC 001334            2358#   9894*  11824
TRAPVE= 000034           1274#   3710*  3711*  9932*
TRT    = 000003         12120#  12613
```

```
TRTVEC= 000014          1269#
TSTATN  044074          4206    4496    4548    4597    4624    4650    4677    4783    4817    4929    4971    4993    5019
                        5056    5149    5189    5259    5341    5432    5452    5501    5547    6091    6417    6617    6844
                        6864    6913    6959    7196    7516    7737    7865    8080    8208    8438    8566    8752    8802
                        9047#   9824
TST1    010124          3840    3862#
TST10   012142          4283    4317#
TST11   012246          4324    4338    4347#
TST12   012354          4371    4381#
TST13   012542          4418    4433#
TST14   012706          4465    4478#
TST15   014262          4665    4721#
TST16   015072          4847    4859#
TST17   016362          5069    5088#
TST2    010300          3890    3911#
TST20   017424          5285#
TST21   020004          5362    5368#
TST22   020162          5415#
TST23   021260          5568    5612#
TST24   022170          5761    5773#
TST25   022766          5904#
TST26   023456          6022#
TST27   024602          6163    6227#
TST3    011040          3958    4058#
TST30   025272          6343#
TST31   026420          6475    6541#
TST32   027342          6650    6728#
TST33   030006          6788    6807    6827#
TST34   031234          6980    7052#
TST35   033356          7332    7412    7438#
TST36   034452          7588    7663#
TST37   036376          7936    8001#
TST4    011366          4088    4152#
TST40   040334          8279    8342#
TST41   042352          8637    8713#
TST5    011520          4193#
TST6    011620          4230#
TST7    012004          4266    4278#
TYPDS = 104405          8855    10869#
TYPE  = 104401          3772    3787    3789    3791    3806    3847    3920    3930    3959    3988    4029    4160    4174
                        4182    4287    4308    4309    4865    4868    5649    5702    5714    5792    5838    5850    5932
                        5969    5980    6135    6147    6255    6292    6303    6450    6462    6808    6812    6813    7059
                        7073    7097    7245    7387    7399    7442    7445    7560    7572    7780    7792    7908    7920
                        8123    8135    8251    8263    8481    8493    8609    8621    8853    8856    8890    8940    9121
                        9777    9779    9834    9839    9867    9909    9934    10025   10033   10101   10195   10321   10377
                        10389   10443   10444   10447   10458   10468   10479   10498   10546   10552   10557   10561   10566
                        10567   10569   10572   10576   10642   10644   10784   10865#  12041   12042   12046   12047   12056
                        12065   12069   12074   12076   12082   12085
TYPERR  067352          10032   12031#
TYPOC = 104402          10446   10866#  12062
TYPON = 104404          10868#
TYPOS = 104403          3933    3991    4032    4178    9124    9870    9912    10867#
T.A2  = 000001          2506#   4252    4528    4562    4763    4797    4828    4909    4943    5129    5163    5213    5324
                        5355    5466    5527    5561    5662    5727    5873    5945    6043    6074    6105    6194    6268
                        6370    6397    6431    6509    6571    6600    6631    6878    6939    6973    7145    7179    7210
                        7258    7470    7499    7530    7695    7720    7751    7822    7848    7879    8038    8063    8094
```

```
                    8165    8191    8222    8396    8421    8452    8523    8549    8580    8775    9213
T.B2  = 000002      2507#   4252    4528    4562    4763    4797    4828    4909    4943    5129    5163    5213    5324
                    5355    5466    5527    5561    5662    5727    5873    5945    6043    6074    6105    6194    6268
                    6397    6431    6509    6571    6600    6631    6878    6939    6973    7145    7179    7210    7258
                    7470    7499    7530    7695    7720    7751    7822    7848    7879    8038    8063    8094    8165
                    8191    8222    8396    8421    8452    8523    8549    8580    8775    9217
T.B3  = 000004      2508#   4252    4763    4909    5129    5213    9221
T1      001412      2391#   5271    8814
T10     001414      2392#   3954    3968    3973    4068    4073    4203    4212    4333    4405    4486    4545    4674
                    4683    4741    4750    4780    4814    4887    4896    4926    4968    4990    5016    5053    5107
                    5116    5146    5186    5249    5256    5338    5429    5449    5475    5507    5544    5581    5797
                    5855    6034    6088    6170    6354    6414    6483    6562    6614    6664    6841    6861    6887
                    6919    6956    6993    7027    7136    7193    7299    7363    7461    7513    7601    7686    7734
                    7813    7862    7947    8029    8077    8156    8205    8290    8353    8387    8435    8514    8563
                    8648    8721    8792    8799    9266    9282    9292    9302    9311    9328    9786    9805    9814
                    9821
T100    001416      2393#   4514    5512    6924
T2500   001422      2395#   6055    6582
T500    001420      2394#   5585    5963    6286    7031    9379    9790
T5000   001424      2396#   9359
T50000  001426      2397#   5642    5696    5785    5832    5925    6129    6174    6248    6444    6487    6668    6746
                    6997    7090    7238    7303    7367    7381    7554    7605    7702    7774    7830    7902    7951
                    8045    8117    8173    8245    8294    8357    8403    8475    8531    8603    8652    8725
UFE   = 000400      1338#   4011    4104    4353    4370
UNLD    003316      2437#   3815*   4197*   4663*   4668*   5424*   5516*   5589*   6836*   6928*   7035*   9773    9794*
UNLOAD= 000007      1306#   4202    4673
UNS   = 040000      1363#
UPE   = 020000      1343#
VV    = 000100      1374#
WCE   = 040000      1344#
WDCNT   001442      2404#   5747*   5755*   5756    5888*   5894*   5895    5988*   5998*   5999    6311*   6321*   6322
                   11792
WLE   = 004000      1360#
WORD    001452      2410#
WRDATA= 000023      1312#
WRHEAD= 000027      1314#   5641    5784    5924    6247    7089    7237
WRL   = 004000      1377#
WRTCHK= 000031      1315#
WRTGAT= 040000      1395#
$APTHD  001000      1512    1518#
$ASTAT= ****** U    10238   10253
$ATYC   051616      10209   10211#
$ATY1   051572      10207#
$ATY3   051600      10086   10208#
$ATY4   051610      10038   10210#
$AUTOB  001134      2261#   8900*   10436   10593
$BASE   001264      2331#   3810*   3872    8958*   8960*
$BOADR  001122      2256#
$BODAT  001126      2258#
$BELL   001200      2282#   10025   10058   10389   10586
$CDW1   001270      2333#
$CDW2   001272      2334#
$CHARC  051342      10103#  10113*  10120   10129*  10134#
$CKSWR  052564      10422#  10873
$CMTAG  001100      2244#   3699    3700    3708    3714    3715    3716
$CM3  = 000000      2274#
```

```
$CM4  = 000006          2274#   2275#   2276#   2277#   2278#   2279#   2280#
$CNTLC  053474         10377   10458   10586#
$CNTLG  053506         10443   10588#
$CNTLU  053501         10468   10561   10587#
$CPUOP  001236          2305#
$CRLF   001205          2284#   3930    3988    4182    4308    4309    6812    6813   10033   10058   10102   10137   10479
                       10566   10586   10647   12041   12046   12056   12069   12082
$DBLK   051562         10161   10195   10203#
$DB2D   054052         10699   10717#
$DB20   053676          6810   10658#
$DDW0   001274          2335#
$DDW1   001276          2336#
$DDW2   001300          2337#
$DDW3   001302          2338#
$DDW4   001304          2339#
$DDW5   001306          2340#
$DDW6   001310          2341#
$DDW7   001312          2342#
$DECVL  054232         10719   10765#
$DEVCT  001220          2296#   3759*   3820*   4164    8838*   8839
$DEVM   001266          2332#   3924
$DOAGN  043230          8849    8858    8864#
$DTBL   051552         10164   10199#
$ENDAD  043220          1498    8860#  10053
$ENDCT  043166          3714    8851#
$ENDMG  043237          8853    8868#
$ENULL  043234          8856    8867#
$ENV    001230          2301#   8894   10035   10081   10216   10240
$ENVM   001231          2302#   3737    3916   10083   10088   10218
$EOP    043076          4662    5618    5650    5703    5715    5793    5839    5851    5933    5970    5981    6136    6148
                        6256    6293    6304    6451    6463    7064    7098    7246    7388    7400    7561    7573    7781
                        7793    7909    7921    8124    8136    8252    8264    8482    8494    8610    8622    8821    8833#
                       12094
$EOPCT  043160          3714*   8848#   8852    9835*
$EOPI   043132          3894    4161    8840    8842#   9837    9889
$ERFLG  001103          2247#   9732    9945    9974    9976    9982*  10004   10020*  10058   12091
$ERMAX  001115          2253#   3717*   9976    9999*  10004
$ERROR  050664          3708   10018#
$ERRPC  001116          2254#  10027*  10028*  10029   10058   11769   11776   11784   11792   11800   11808   11816   11825
                       11836   11848
$ERRTB  003506          2550#  12038
$ERTTL  001112          2251#  10026*  10058
$ESCAP  001176          2281#   3716*   3818*   4495*   4512*   4652*   4655*   4690*   5374*   5402*   5488*   5515*   5573*
                        6031*   6054*   6124*   6204*   6352*   6379*   6440*   6518*   6558*   6581*   6686*   6900*   6927*
                        7015*   7127*   7129*   7157*   7160*   7227*   7321*   7335*   7457*   7480*   7547*   7623*   7677*
                        7701*   7768*   7808*   7829*   7896*   7969*   8020*   8044*   8111*   8151*   8172*   8239*   8312*
                        8378*   8402*   8469*   8509*   8530*   8597*   8670*   9171    9175*   9185*   9195*   9205*   9226*
                        9230*   9749*   9836*   9893*   9897*   9998*  10049   10051   10058
$ETABL  001230          2300#
$ETEND  001314          1524    2343#
$FATAL  001212          2293#  10244*
$FFLG   052036         10207*  10210*  10238   10247*  10255#
$FILLC  001156          2272#  10106   10137
$FILLS  001155          2271#  10137
$GDADR  001120          2255#
$GDDAT  001124          2257#
```

```
$GET42  043210          8857#
$GTSWR  052654         10444#  10871
$HD   = 000000          1140
$HIBTS  001000          1519#
$HIOCT  053674         10635*  10646#
$ICNT   001104          2248#   9989*   9990    9992*  10003
$INTAG  001135          2262#  10441*  10460   10480   10593
$ITEMB  001114          2252#  10029*  10037   10058   12032
$LF     001206          2285#  10058   10137   10576   10586  10647
$LFLG   052035         10248*  10254#
$LPADR  001106          2249#   3718*   4694    5574    6208   6522   6690   7019   7325   7339   7627   7973   8316
                        8674    9876    9980*   9996*  10001  10003
$LPERR  001110          2250#   3719*   4693    6207    6521   6689   7018   7324   7338   7626   7972   8315   8673
                        9231    9734    9737*   9903    9980   9997*  10003  10048
$MADR1  001242          2318#
$MADR2  001246          2322#
$MADR3  001252          2325#
$MADR4  001256          2328#
$MAIL   001210          1520    1524    2291#   3736    8894   9995   10035  10081
$MAMS1  001240          2312#
$MAMS2  001244          2320#
$MAMS3  001250          2323#
$MAMS4  001254          2326#
$MBADR  001002          1520#
$MFLG   052034         10208*  10214   10249*  10253#
$MNEW   053524         10447   10591#
$MSGAD  001224          2298#  10224*  10227
$MSGLG  001226          2299#  10229*
$MSGTY  001210          2292#  10222   10230*  10242   10246*
$MSWR   053513         10444   10589#
$MTYP1  001241          2313#
$MTYP2  001245          2321#
$MTYP3  001251          2324#
$MTYP4  001255          2327#
$MXCNT  050662          9993   10003#
$NULL   001154          2270#  10108   10137
$NWTST= 000001          3852#   3854    3897#   3899   4043#  4045   4143#  4145   4184#  4186   4223#  4225   4271#
                        4273    4311#   4313    4341#  4343   4374#  4376   4421#  4423   4468#  4470   4710#  4712
                        4850#   4852    5075#   5077   5276#  5278   5365#  5404#  5406   5600#  5602   5764#  5766
                        5901#   6008#   6010    6224#  6330#  6332   6533#  6535   6706#  6708   6815#  6817   7043#
                        7045    7431#   7433    7646#  7648   7998#  8339#  8700#  8702
$OCNT   052262         10293*  10322*  10335#
$OCTVL  054000          4291   10660   10685#
$OMODE  052264         10288*  10292*  10297   10300*  10311*  10337#
$OVER   050646          9957    9973    9981    9991   10000#
$PASS   001216          2295#   3736*   4282    6806    8845*  8846*  8854   8867   9987   10004
$PASTM  001006          1522#
$PWRCT  050402          9927*   9928*   9938#
$PWRDN  050316          3712    9921#   9930
$PWRUP  050330          9921    9927#
$QUES   001204          2283#  10058   10137   10498   10569  10586  10644  10647
$RDCHR  053126         10511#  10874
$RDDEC= ****** U        10877
$RDLIN  053216         10534#  10875
$RDOCT  053536         10608#  10876
$RDSZ = 000022         10527#
```

```
$RESRE  054344          10822#  10878
$RTNAD  043232           8866#
$R2A   = ****** U       10879
$SAVRE  054306          10806#  10877
$SB2D   054016          10697#
$SCOPE  050404           3706    9954#
$SETUP= 000137           3646#   3705    3706    3708    3710    3712    3714    3715    3716    3718    8843    8891    9955
                        10019   10045   10053   10381   10386   10387   10417   10593
$STUP  = 177777          3646#
$SUPRS  054246           4307    6811   10776#
$SVLAD  050612           9965    9994#
$SVPC  = 000274          1496#   1501
$SWR   = 167400          1125#   1140    1146    1147    1148    1149    1150    1151    1152    1153    2280    2281    2282
                         3715    3716    3718    3719    3863    3912    4059    4153    4194    4231    4279    4318    4348
                         4382    4434    4479    4722    4860    5089    5286    5369    5416    5613    5774    5905    6023
                         6228    6344    6542    6729    6828    7053    7439    7664    8002    8343    8714    8830    8844
                         8859    8865    8867    9946    9947    9948    9949    9950    9956    9968    9970    9971    9974
                         9975    9976    9983    9984    9985    9997   10000   10003   10010   10011   10012   10013   10014
                        10023   10030   10042   10046   10058
$SWREG  001232           2303#   3739
$SWRMK= 000000           1153    1154    9950    9951    9972
$TESTN  001214           2294#   4155*   9995*  11769   11776   11784   11792   11800   11808   11816   11824   11825   11836
                        11848
$TIMES  001174           2280#   3715*   3863*   3912*   4059*   4153*   4194*   4231*   4279*   4318*   4348*   4382*   4434*
                         4479*   4722*   4860*   5089*   5286*   5369*   5416*   5613*   5774*   5905*   6023*   6228*   6344*
                         6542*   6729*   6828*   7053*   7439*   7664*   8002*   8343*   8714*   8836*   8844*   9983*   9990
                         9993*  10003
$TKB    001146           2267#  10341   10362   10373   10398   10426   10453
$TKCNT  052266          10342#  10357*  10387   10404*  10518   10520*
$TKINT  052276           3682   10357#  10378   10439
$TKQEN= 052275          10346#  10412   10523
$TKQIN  052270          10343#  10358*  10359   10410*  10411*  10412   10414*
$TKQOU  052272          10344#  10359*  10521   10522*  10523   10525*
$TKQSR  052274          10345#  10358   10414   10525
$TKS    001144           2266#  10341   10363*  10394*  10396   10402*  10424   10440*  10450   10462*  10482*
$TKSRV  052346          10360   10373#
$TMP0   001160           2274#
$TMP1   001162           2275#
$TMP2   001164           2276#
$TMP3   001166           2277#
$TMP4   001170           2278#
$TMP5   001172           2279#   9171*   9226    9230
$TN    = 000042          1126#   1140    3840    3852    3863#   3890    3897    3912#   3958    4043    4059#   4088    4143
                         4153#   4155    4156    4184    4194#   4223    4231#   4266    4271    4279#   4283    4311    4318#
                         4324    4338    4341    4348#   4371    4374    4382#   4418    4421    4434#   4465    4468    4479#
                         4665    4710    4722#   4847    4850    4860#   5069    5075    5089#   5276    5286#   5362    5365
                         5369#   5404    5416#   5568    5600    5613#   5761    5764    5774#   5901    5905#   6008    6023#
                         6163    6224    6228#   6330    6344#   6475    6533    6542#   6650    6706    6729#   6788    6807
                         6815    6828#   6980    7043    7053#   7332    7412    7431    7439#   7588    7646    7664#   7936
                         7998    8002#   8279    8339    8343#   8637    8700    8714#
$TNPWR  054162          10724   10725   10745#
$TPB    001152           2269#  10126*  10137
$TPFLG  001157           2273#  10075   10137
$TPS    001150           2268#  10124   10137
$TRAP   054402           3710   10842#
$TRAP2  054424          10853#  10864
```

```
$TRP  = 000016          10857#  10866#  10867#  10868#  10869#  10870#  10871   10872#  10873   10874#  10875#  10876#  10877#
                        10878#  10879#  10880#
$TRPAD  054436          10847   10864#
$TSTM   001004           1521#
$TSTNM  001102           2246#   4156#   8843#   9758    9945    9972    9994#   9995    10000   10004   10022   10058
$TTYIN  053452          10536   10537   10549   10567   10581   10585#
$TYPBN= ****** U        10870
$TYPDS  051346          10149#  10869
$TYPE   051064          10075#  10235   10857   10865
$TYPEC  051276          10105   10112   10119   10124#  10125   10484
$TYPEX  051344          10130   10132   10135#
$TYPOC  052064          10291#  10866
$TYPON  052100          10290   10293#  10868
$TYPOS  052040          10286#  10867
$UNIT   001222           2297#   3760*   3821*   4166*   4171    4175    4543    4737    4778    4883    4924    4988    5051
                         5103    5144    5247    5254    5336    5447    5542    6086    6412    6612    6859    6954    7191
                         7511    7732    7860    8075    8203    8433    8561    8790    8797    9048    9067    9093    9135
                         9136    9137    9138    9264    9279    9289    9299    9309    9331    9801    9819
$UNITM  001010           1523#
$USWR   001234           2304#
$VECT1  001260           2329#
$VECT2  001262           2330#
$XTSTR  050416           9959#
$$GET4= 000000           8859#
$OFILL  052263          10287*  10291*  10301   10336#
$40CAT= ****** U         9956   10032
.       = 067712         1470#   1474#   1479#   1481#   1483#   1485#   1487#   1489#   1496    1497#   1499#   1501#   1502#
                         1508    1509#   1511#   1513#   2243#   2286    2423#   2424#   2425#   2427#   3703    3718    3719
                         8867    8871    9923   10003   10004   10058   10137   10203#  10256#  10341   10345#  10346   10347#
                        10585#  10586   10593#  10647   10685#  10765#  10884#  12105#  12106#  12251   12476   12480   12512
                        12536   12598   12606   12609#  12617#
.$ASTA= ****** U        10208   10211
.$X   = 001000           1508#   1513
```

```
CALIB   1648#  4735  4881  5101  9799
CHECK   1567#  4250  4503  4526  4560  4604  4631  4761  4795  4826  4907  4941  5127  5161  5210
        5322   5353  5438  5464  5493  5525  5559  5660  5725  5813  5871  5943  6041  6072  6103
        6192   6266  6368  6395  6429  6507  6569  6598  6629  6761  6850  6876  6905  6937  6971
        7143   7177  7208  7256  7468  7497  7528  7693  7718  7749  7820  7846  7877  8036  8061
        8092   8163  8189  8220  8394  8419  8450  8521  8547  8578  8773
CIDAE   1679#  5244  8787
COMMEN  1278#
CWD2    1581#  4257  4533  4768  4914  5134  5218  5532  5668  5733  5878  6402  6944
DRCLR   1622#  4541  4776  4922  4985  5048  5142  5252  5333  5445  5540  6083  6410  6609  6857
        6952   7188  7508  7729  7857  8072  8200  8430  8558  8795  9817
ENDCOM  1278#
EOPGM   2224#  8834
ERROR   1172#  3893  3949  3956  3970  3975  3997  4005  4020  4024  4035  4039  4070  4075  4091
        4099   4117  4121  4125  4129  4200  4205  4207  4210  4214  4235  4253  4254  4255  4256
        4260   4263  4267  4285  4330  4335  4339  4352  4355  4362  4369  4372  4386  4389  4392
        4395   4402  4407  4410  4413  4416  4419  4438  4441  4444  4451  4457  4460  4463  4466
        4483   4488  4493  4498  4506  4507  4508  4509  4516  4529  4530  4531  4532  4536  4539
        4547   4550  4563  4564  4565  4566  4580  4594  4599  4607  4608  4609  4610  4621  4626
        4634   4635  4636  4637  4647  4651  4661  4671  4676  4678  4681  4685  4688  4733  4743
        4749   4752  4764  4765  4766  4767  4771  4774  4782  4785  4798  4799  4800  4801  4808
        4816   4818  4829  4830  4831  4832  4836  4840  4879  4889  4895  4898  4910  4911  4912
        4913   4917  4920  4928  4931  4944  4945  4946  4947  4954  4970  4972  4978  4983  4992
        4995   5003  5018  5020  5026  5035  5041  5046  5055  5058  5098  5109  5115  5118  5130
        5131   5132  5133  5137  5140  5148  5151  5164  5165  5166  5167  5172  5179  5188  5190
        5214   5215  5216  5217  5221  5224  5236  5251  5258  5261  5268  5273  5290  5294  5300
        5305   5309  5312  5325  5326  5327  5328  5331  5340  5343  5356  5357  5358  5359  5363
        5373   5379  5383  5393  5397  5420  5431  5433  5441  5442  5443  5444  5451  5454  5467
        5468   5469  5470  5477  5481  5485  5496  5497  5498  5499  5502  5506  5509  5511  5514
        5528   5529  5530  5531  5535  5538  5546  5549  5562  5563  5564  5565  5578  5583  5587
        5617   5628  5644  5648  5663  5664  5665  5666  5671  5674  5686  5698  5701  5713  5728
        5729   5730  5731  5736  5739  5754  5778  5787  5791  5799  5802  5816  5817  5818  5819
        5827   5834  5837  5849  5857  5860  5874  5875  5876  5877  5881  5884  5893  5909  5927
        5931   5946  5947  5948  5949  5956  5965  5968  5979  5996  6026  6036  6044  6045  6046
        6047   6051  6059  6062  6075  6076  6077  6078  6081  6090  6093  6106  6107  6108  6109
        6114   6122  6131  6134  6146  6154  6160  6167  6172  6176  6179  6182  6195  6196  6197
        6198   6201  6232  6250  6254  6269  6270  6271  6272  6279  6288  6291  6302  6319  6348
        6356   6362  6371  6372  6373  6374  6377  6382  6385  6398  6399  6400  6401  6405  6408
        6416   6419  6432  6433  6434  6435  6438  6446  6449  6461  6468  6472  6479  6485  6489
        6492   6495  6510  6511  6512  6513  6516  6546  6556  6564  6572  6573  6574  6575  6579
        6585   6588  6601  6602  6603  6604  6607  6616  6619  6632  6633  6634  6635  6639  6659
        6666   6670  6673  6676  6684  6733  6748  6752  6764  6765  6766  6767  6770  6777  6779
        6792   6832  6843  6845  6853  6854  6855  6856  6863  6866  6879  6880  6881  6882  6889
        6893   6897  6908  6909  6910  6911  6914  6918  6921  6923  6926  6940  6941  6942  6943
        6947   6950  6958  6961  6974  6975  6976  6977  6988  6995  6999  7002  7005  7013  7024
        7029   7033  7063  7067  7071  7092  7096  7123  7138  7146  7147  7148  7149  7153  7164
        7167   7180  7181  7182  7183  7186  7195  7198  7211  7212  7213  7214  7218  7225  7240
        7244   7259  7260  7261  7262  7294  7301  7305  7308  7311  7319  7365  7369  7372  7375
        7383   7386  7398  7405  7455  7463  7471  7472  7473  7474  7478  7484  7487  7500  7501
        7502   7503  7506  7515  7518  7531  7532  7533  7534  7538  7545  7556  7559  7571  7578
        7596   7603  7607  7610  7613  7621  7675  7688  7696  7697  7698  7699  7705  7708  7721
        7722   7723  7724  7727  7736  7739  7752  7753  7754  7755  7759  7766  7776  7779  7791
        7798   7806  7815  7823  7824  7825  7826  7833  7836  7849  7850  7851  7852  7855  7864
        7867   7880  7881  7882  7883  7887  7894  7904  7907  7919  7926  7942  7949  7953  7956
        7959   7967  8018  8031  8039  8040  8041  8042  8048  8051  8064  8065  8066  8067  8070
        8079   8082  8095  8096  8097  8098  8102  8109  8119  8122  8134  8141  8149  8158  8166
```

```
            8167    8168    8169    8176    8179    8192    8193    8194    8195    8198    8207    8210    8223    8224    8225
            8226    8230    8237    8247    8250    8262    8269    8285    8292    8296    8299    8302    8310    8348    8355
            8359    8362    8365    8376    8389    8397    8398    8399    8400    8406    8409    8422    8423    8424    8425
            8428    8437    8440    8453    8454    8455    8456    8460    8467    8477    8480    8492    8499    8507    8516
            8524    8525    8526    8527    8534    8537    8550    8551    8552    8553    8556    8565    8568    8581    8582
            8583    8584    8588    8595    8605    8608    8620    8627    8643    8650    8654    8657    8660    8668    8718
            8723    8727    8730    8733    8743    8751    8753    8757    8760    8763    8766    8776    8777    8778    8779
            8782    8785    8794    8801    8804    8811    8816    9245    9250    9254    9268    9284    9294    9304    9313
            9330    9771    9783    9788    9792    9807    9813    9816    9823    9826    9895
ESCAPE      1278#
FSECA       1976#   9353
FSECB       2002#   9373
F.EAB       1548#   4517    4552    4753    4787    4899    4933    5119    5153    5313    5345    5456    5517    5551    5652
            5717    5804    5862    5935    6063    6095    6184    6258    6386    6421    6498    6589    6621    6753    6868
            6929    6963    7168    7200    7248    7488    7520    7709    7741    7837    7869    8052    8084    8180    8212
            8410    8442    8538    8570
GETPRI      1278#
GETSWR      1278#   8891
HDCHK3      1903#   7377    7550    7770    7898    8113    8241    8471    8599
HDTBL       1964#   5916    6239    7076
HEADER      2136#   5900    6223
LIMIT       2061#   5418    6830
LOOP        1532#   3963    4063    4326    4357    4397    4446    4728    4804    4875    4950    4999    5093    5174    5295
            5624    5682    5822    5952    6117    6275    6551    7118    7220    7450    7540    7670    7761    7802    7889
            8013    8104    8145    8232    8371    8462    8503    8590    8738
LPCHK       1593#   4690    6203    6517    6686    7015    7321    7335    7623    7969    8312    8670
MSG         3852#   3854    3897#   3899    4043#   4045    4143#   4145    4184#   4186    4223#   4225    4271#   4273    4311#
            4313    4341#   4343    4374#   4376    4421#   4423    4468#   4470    4710#   4712    4850#   4852    5075#   5077
            5276#   5278    5404#   5406    5599#   5602    5764#   5766    6007#   6010    6329#   6332    6533#   6535    6706#
            6708    6815#   6817    7043#   7045    7431#   7433    7646#   7648    8700#   8702
MULT        1278#
NEWTST      1278#   3852    3897    4043    4143    4184    4223    4271    4311    4341    4374    4421    4468    4710    4850
            5075    5276    5365    5404    5600    5764    5901    6008    6224    6330    6533    6706    6815    7043    7431
            7646    7998    8339    8700
OWNTAG      1526#   2345
POP         1278#   10190   10250   10251   10636   10827
PUSH        1278#   10149   10211   10213   10234   10610   10807
QKRPSK      1784#   6656    6985    7292    7593    7939    8282    8640
QKSEEK      1760#   6168    6481    6662    6991    7297    7361    7599    7945    8288    8351    8646    8719
QKSRT       1737#   5575    7021    9780
QKUNLD      1808#   4198    4669
RALLHD      1913#   5959    6282
RDHDR       1866#   5693    5829    6126    6441    7378    7551    7771    7899    8114    8242    8472    8600
REPORT      1278#
SCOPE       1173#   3862    3911    4058    4152    4193    4230    4278    4317    4347    4381    4433    4478    4721    4859
            5088    5285    5368    5415    5612    5773    5904    6022    6227    6343    6541    6728    6827    7052    7438
            7663    8001    8342    8713    8835    8842
SECTST      2029#   5377
SETPRI      1278#   10514
SETTRA      10857#  10866   10867   10868   10869   10871   10873   10874   10875   10876   10877   10878   10879
SETUP       1278#   3698
SKATN       1718#   5307    6057    6583    7162    7482    7703    7831    8046    8174    8404    8532
SKIP        1278#   3840    3890    3958    4088    4266    4283    4324    4338    4371    4418    4465    4665    4847    5069
            5362    5568    5761    6163    6475    6650    6788    6807    6980    7332    7412    7588    7936    8279    8637
SKOSC       2168#   7666    8009    8367
SKRDY       1704#   6032    6560    7134    7459    7684    7811    8027    8154    8385    8512
```

```
SLASH     1278#
SPACE     1278#
STARS     1278#  1494   1505   1507   1514   2239   2286   2289   3852   3861   3897   3910   4043   4057   4143
          4151   4184   4192   4223   4229   4271   4277   4311   4316   4341   4346   4374   4380   4421   4432
          4468   4477   4710   4720   4850   4858   5075   5087   5276   5284   5365   5367   5404   5414   5600
          5611   5764   5772   5901   5903   6008   6021   6224   6226   6330   6342   6533   6540   6706   6727
          6815   6826   7043   7051   7431   7437   7646   7662   7998   8000   8339   8341   8700   8712   8827
          9942  10006  10060  10139  10206  10263  10340  10417  10432  10503  10527  10596  10649  10688  10706
         10768  10791  10836  12021  12030
SWRSU     1278#  3720#
SW814     1601#  4664   5567   6161   6473   6648   6979   7331   7411   7586   7934   8277   8635
TRMTRP   10857#
TSTSW9    1611#  4699   4704   5590   6210   6215   6523   6528   6694   6700   7036   7341   7347   7415   7420
          7631   7637   7975   7980   7985   7990   8318   8323   8328   8333   8676   8681   8686   8691
TYPBIN    1278#
TYPDEC    1279#  8854
TYPNAM    1278#
TYPNUM    1278#
TYPOCS    1278#  3931   3989   4030   4176   9122   9868   9910
TYPOCT    1278#  10445
TYPTXT    1278#
WRHDR     1842#  5640   5783   5923   6246   7088   7236
$$CMRE    2237#
$$CMTM    2237#  2274   2275   2276   2277   2278   2279
$$ESCA    1278#
$$NEWT    1278#  3852   3897   4043   4143   4184   4223   4271   4311   4341   4374   4421   4468   4710   4850
          5075   5276   5365   5404   5600   5764   5901   6008   6224   6330   6533   6706   6815   7043   7431
          7646   7998   8339   8700
$$SET    10857#  10866  10867  10868  10869  10871  10873  10874  10875  10876  10877  10878  10879
$$SETM    3736#
$$SKIP    1278#  3840   3890   3958   4088   4266   4283   4324   4338   4371   4418   4465   4665   4847   5069
          5362   5568   5761   6163   6475   6650   6788   6807   6980   7332   7412   7588   7936   8279   8637
.EQUAT    1125#  1168
.HEADE    1125#  1130
.SETUP    1125#  3646
.SWRHI    1125#  1142
.SWRLO    1125#  1154#
.$ACTI    1125#  1492
.$APTB    2287#
.$APTH    1125#  1503
.$APTY    1125#  10204
.$CATC    1125#  1468
.$CMTA    1125#  2237
.$DB2D    1125#  10704
.$DB2O    1125#  10647
.$EOP     1125#  8825
.$ERRO    1125#  10004
.$RDOC    1125#  10594
.$READ    1125#  10338
.$SAVE    1125#  10789
.$SB2D    1125#  10686
.$SCOP    1125#  9940
.$SUPR    1125#  10766
.$TRAP    1125#  10834
.$TYPD    1125#  10137
.$TYPE    1125#  10058
```

```
.$TYPO   1125#  10261


. ABS.  071762     000


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

 DZR6HD,DZR6HD.SEQ/SOL/CRF/NL:TOC/DOC=DZR6HD.P11
 RUN-TIME: 28 29 3 SECONDS
 RUN-TIME RATIO: 465/61=7.5
 CORE USED:  32K  (64 PAGES)

 DOCUMENT PAGES:  260
```