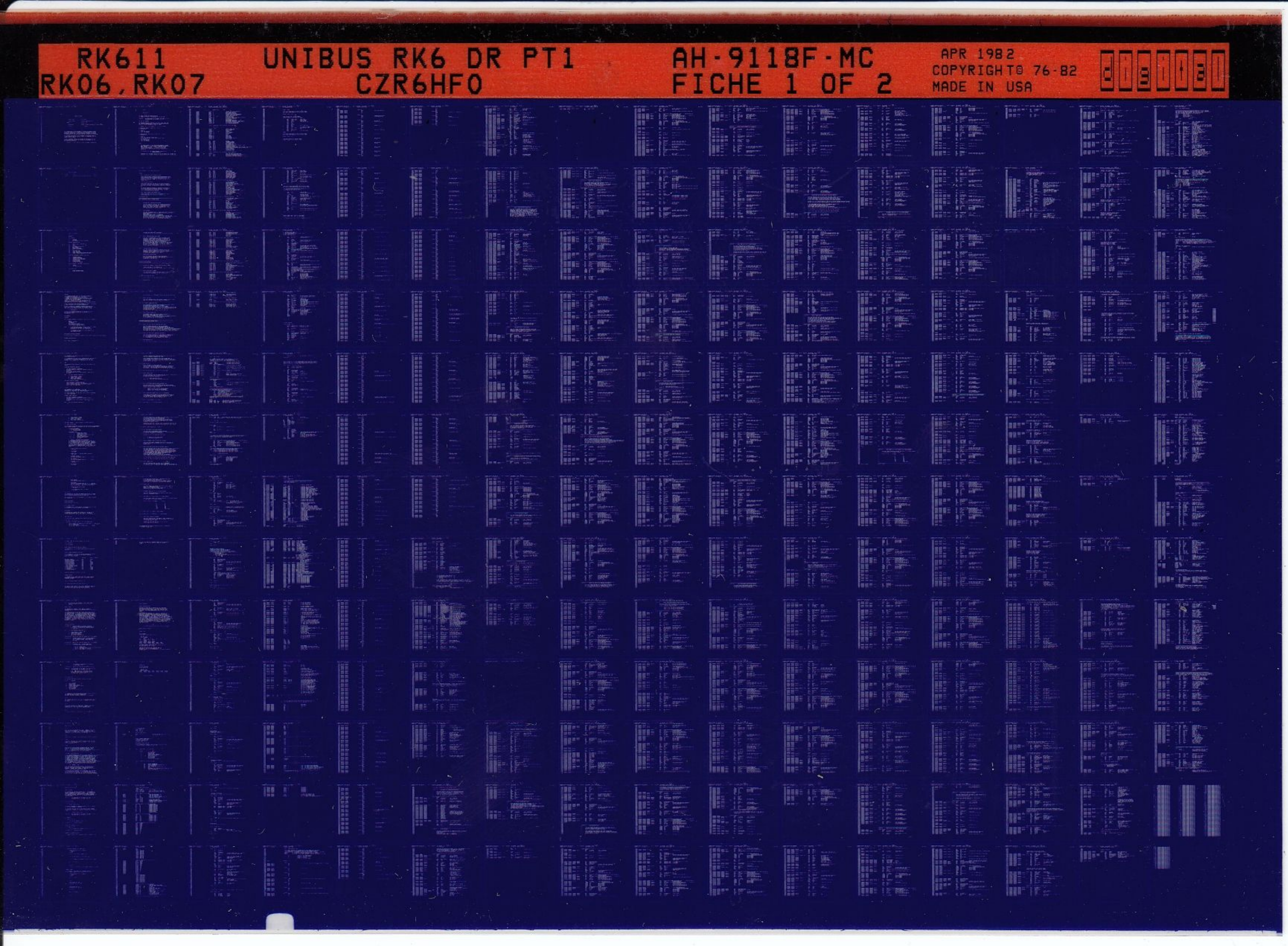
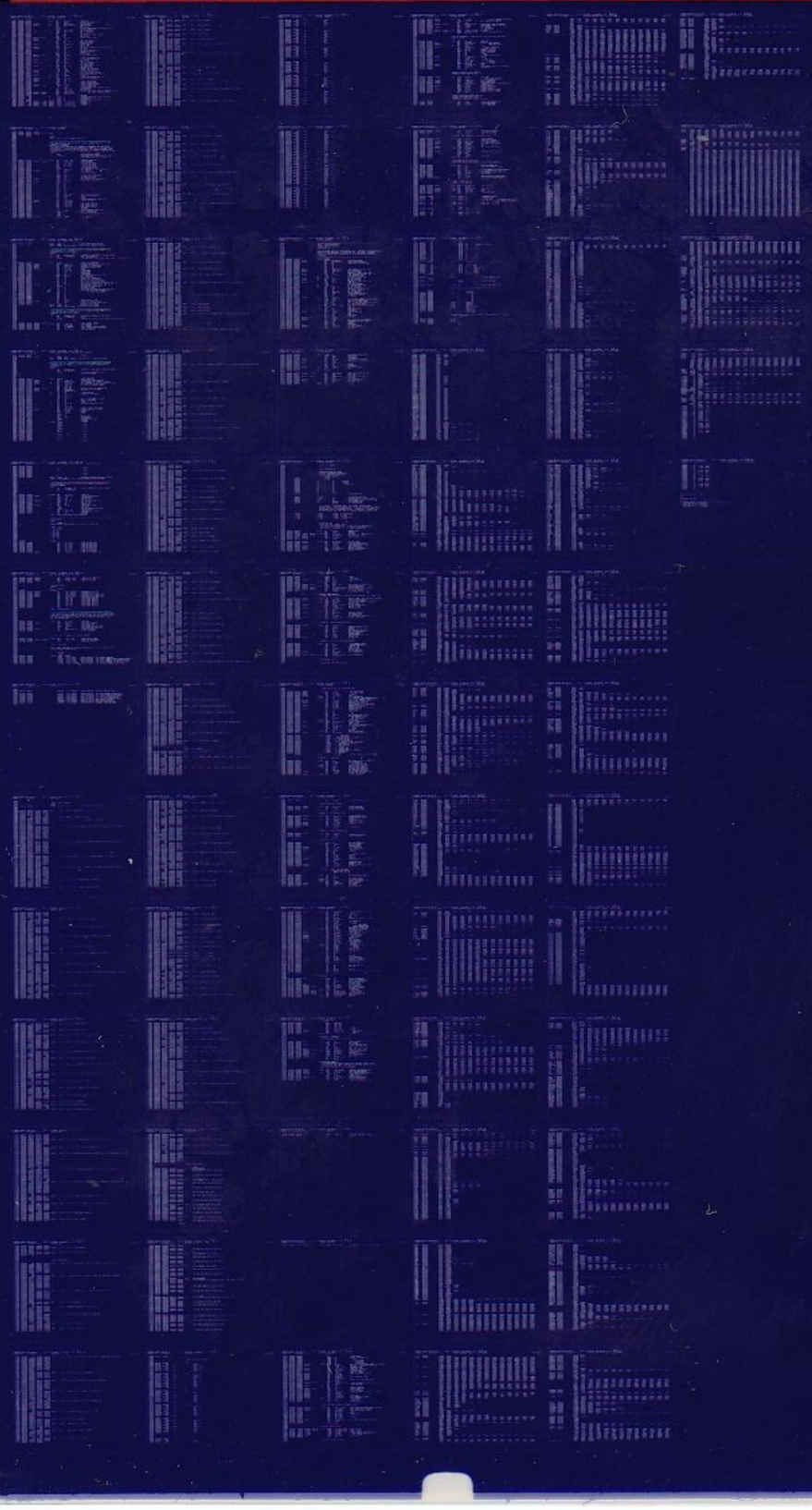
RK611 UNIBUS RK6 DR PT1 AH-9118F-MC
RK06,RK07 CZR6HF0 FICHE 1 OF 2



RK611 UNIBUS RK6 DR PT1 AH-9118F-MC
RK06,RK07 CZR6HF0 FICHE 2 OF 2



.REM %

IDENTIFICATION

PRODUCT CODE: AC-9116F-MC

PRODUCT NAME: CZR6HF0 UNIBUS RK6 DR PT1

DATE: JANUARY 1982

MAINTAINER: STORAGE SYSTEMS SOFTWARE TEST APPLICATIONS

AUTHOR: B. T. LEBLANC

REVISION HISTORY

REVISON	FIXES	DATE
CZR6HF0	IMPLEMENTED XXDP LOAD MEDIA OPTION	JAN 82

TABLE OF CONTENTS

```
                        1.0     ABSTRACT

                                THIS PROGRAM PERFORMS PART 1 OF THE DRIVE DIAGNOSTICS
                                 TO INSURE THAT THE RK06 & RK07 DISK DRIVE
                                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 (TOTAL) RK06/RK07 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 DR0 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
```

THE ABOVE TIMES ARE APPROX. DOUBLED FOR THE RK07.

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.

ALSO, THE PROGRAM WILL DETERMINE WHETHER
THE DRIVE TO BE TESTED IS AN RK06 OR RK07.
OPERATOR INPUTS ARE NOT REG'D.

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-RK07 DRIVE DIAGNOSTIC
                                           PART 1
                                        CZR6HF0

                                        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-RK07 DRIVE DIAGNOSTIC
                                           PART 1
                                        CZR6HF0

                                        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 BJT 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 AS AN RK06.  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.
         IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.


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 AS AN RK06
         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.
         IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.


TEST 4   FIND NEXT DRIVE TO BE TESTED

         THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
         ADDRESS IN 'DRVAD' & $TMP4 IS SET TO CDT IF DRIVE IS RK07.
         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) (512 FOR THE RK07)
                WITH EVEN PARITY SET. THIS FREEZES THE INFORMATION IN THE ABOVE REGISTER
                AND ALLOWS FOR CHECKING. 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 THE LAST CYLINDER

                        THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO THE LAST CYL
                        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 632 (1456 FOR RK07), 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 632 (1456) BY DETECTING INNER LIMIT AS THE ADJACENT CYL.
                        IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON THE LAST CYL
```

```
                        & 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 632 (1456) WITH WRITE HEADER CMDS (ALL TRACKS).
                        HEADERS WILL BE READ IN THE NEXT TEST


                TEST 35 DECREMENT FROM LAST CYL 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
                                        512             1000 (RK07)


                TEST 37 SEEK TO ALL CYLS FROM 0 & READ HEADERS


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


                TEST 41 SEEK TO ALL INVALID CYLINDERS

                        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 633 THRU 777 (8) FOR THE RK06
                        & 1457 THRU 1777 FOR THE RK07


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

```
1049
1050                    ANY TEST THAT MODIFIES STANDARD FORMATTING IS FOLLOWED BY A
1051                    'CLEAN UP' TEST TO PUT THOSE CYLS BACK TO STANDARD
1052                    FORMAT.
1053
```

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.

```
1110
1111                                    EXAMPLE #2:
1112
1113                                    NO ATTN IN RKASOF
1114                                    AFTER UNLOAD CMD
1115
1116                                      TEST NO.  PC
1117                                      000003  014330
1118                                    RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2   RKASOF
1119                                    000144  100000  000000  100101  000206  000104  000000
1120
1121
1122
1123                                            [ END OF DOCUMENT ]
1124
1125
1126                                %
```

```
                                ; *** PGM REV 039 ***
                                .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  CZR6HF0 UNIBUS RK6 DR PT1
                                ;*COPYRIGHT (C) 1976,1982
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY GARY PAPAZIAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                                ;*


                                .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
```

```
1236    000002          SW01=   2
1237    000001          SW00=   1
1238                    .EQUIV  SW09,SW9
1239                    .EQUIV  SW08,SW8
1240                    .EQUIV  SW07,SW7
1241                    .EQUIV  SW06,SW6
1242                    .EQUIV  SW05,SW5
1243                    .EQUIV  SW04,SW4
1244                    .EQUIV  SW03,SW3
1245                    .EQUIV  SW02,SW2
1246                    .EQUIV  SW01,SW1
1247                    .EQUIV  SW00,SW0
1248
1249                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1250    100000          BIT15=  100000
1251    040000          BIT14=  40000
1252    020000          BIT13=  20000
1253    010000          BIT12=  10000
1254    004000          BIT11=  4000
1255    002000          BIT10=  2000
1256    001000          BIT09=  1000
1257    000400          BIT08=  400
1258    000200          BIT07=  200
1259    000100          BIT06=  100
1260    000040          BIT05=  40
1261    000020          BIT04=  20
1262    000010          BIT03=  10
1263    000004          BIT02=  4
1264    000002          BIT01=  2
1265    000001          BIT00=  1
1266                    .EQUIV  BIT09,BIT9
1267                    .EQUIV  BIT08,BIT8
1268                    .EQUIV  BIT07,BIT7
1269                    .EQUIV  BIT06,BIT6
1270                    .EQUIV  BIT05,BIT5
1271                    .EQUIV  BIT04,BIT4
1272                    .EQUIV  BIT03,BIT3
1273                    .EQUIV  BIT02,BIT2
1274                    .EQUIV  BIT01,BIT1
1275                    .EQUIV  BIT00,BIT0
1276
1277                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1278    000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1279    000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1280    000014          TBITVEC=14              ;;"T" BIT
1281    000014          TRTVEC= 14              ;;TRACE TRAP
1282    000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1283    000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1284    000024          PWRVEC= 24              ;;POWER FAIL
1285    000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1286    000034          TRAPVEC=34              ;;"TRAP" TRAP
1287    000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
1288    000064          TPVEC=  64              ;;TTY PRINTER VECTOR
1289    000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1290
1291                    .SBTTL  RK06 CONTROLLER REGISTER DEFINITION
```

```
1292
1293                            ;       $BASE=177440
1294
1295            000000          RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1296            000002          RKWC=   2               ;WORD COUNT REGISTER
1297            000004          RKBA=   4               ;BUS ADDRESS REGISTER
1298            000006          RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1299            000010          RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1300            000012          RKDS=   12              ;DRIVE STATUS REGISTER
1301            000014          RKER=   14              ;ERROR REGISTER
1302            000016          RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1303            000020          RKDC=   20              ;DESIRED CYL REGISTER
1304            000024          RKDB=   24              ;DATA BUFFER
1305            000026          RKMR1=  26              ;MAINTENANCE REGISTER 1
1306            000034          RKMR2=  34              ;MAINTENANCE REGISTER 2 (MSG LINE A)
1307            000036          RKMR3=  36              ;MAINTENANCE REGISTER 3 (MSG LINE B)
1308            000030          RKECPS= 30              ;ECC POSITION INFORMATION
1309            000032          RKECPT= 32              ;ECC PATTERN INFORMATION
1310
1311                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1312
1313                            ;       DRIVE CMDS
1314
1315            000001          SELDRV= 1               ;SELECT DRIVE (GET STATUS)
1316            000003          PACK=   3               ;PACK ACKNOWLEDGE
1317            000005          CLEAR=  5               ;DRIVE CLEAR
1318            000007          UNLOAD= 7               ;UNLOAD
1319            000011          SRTSPL= 11              ;START SPINDLE
1320            000013          RECAL=  13              ;RECALIBRATE
1321            000015          OFFSET= 15              ;OFFSET
1322            000017          SEEK=   17              ;SEEK
1323            000021          RDDATA= 21              ;READ DATA
1324            000023          WRDATA= 23              ;WRITE DATA
1325            000025          RDHEAD= 25              ;READ HEADER
1326            000027          WRHEAD= 27              ;WRITE HEADER AND DATA
1327            000031          WRTCHK= 31              ;WRITE CHECK
1328
1329            000001          GO=     BIT0            ;GO BIT
1330            000100          IE=     BIT6            ;INTERRUPT ENABLE
1331            000200          RDY=    BIT7            ;CONTROLLER READY
1332            000400          BA16=   BIT8            ;BUS ADDRESS BIT 16
1333            001000          BA17=   BIT9            ;BUS ADDRESS BIT 17
1334            002000          CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06, 1=RK07
1335            004000          CTO=    BIT11           ;CONTROLLER TIMEOUT
1336            010000          CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1337            020000          DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1338            040000          DI=     BIT14           ;DRIVE INTERRUPT
1339            100000          CERR=   BIT15           ;CONTROLLER ERROR
1340            100000          CCLR=   BIT15           ;CONTROLLER CLEAR
1341
1342                            .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1343
1344            000007          DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
1345            000010          RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1346            000020          BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
1347            000040          SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
1348            000100          IR=     BIT6                    ;INPUT READY
1349            000200          OR=     BIT7                    ;OUTPUT READY
1350            000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1351            001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1352            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1353            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1354            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1355            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1356            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1357            100000          DLT=    BIT15                   ;DATA LATE ERROR
1358
1359                            .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1360
1361            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1362            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
1363            000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
1364            000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
1365            000020          FMTE=   BIT4                    ;FORMAT ERROR
1366            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1367            000100          ECH=    BIT6                    ;ECC HARD
1368            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
1369            000400          HVRC=   BIT8                    ;HEADER VRC ERROR
1370            001000          COE=    BIT9                    ;CYL ADDRESS OVERFLOW ERROR
1371            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1372            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1373            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
1374            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1375            040000          UNS=    BIT14                   ;DRIVE UNSAFE
1376            100000          DCK=    BIT15                   ;DATA CHECK
1377
1378                            .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1379
1380            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1381                                                            ;   THIS BIT IS RESET)
1382            000004          OFST=   BIT2                    ;DRIVE OFFSET
1383            000010          ACLO=   BIT3                    ;AC LOW
1384            000020          DCLO=   BIT4                    ;DC LOW
1385            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
1386            000100          VV=     BIT6                    ;VOLUME VALID
1387            000200          DRDY=   BIT7                    ;DRIVE READY
1388            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06, 1=RK07)
1389            004000          WRL=    BIT11                   ;WRITE LOCK
1390            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
1391            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
1392            100000          SVAL=   BIT15                   ;STATUS VALID
1393
1394                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1395
1396            000017          MESMSK= 17                      ;MSG MASK
1397            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MSG LINES
1398            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
1399            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1400            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
1401            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
1402            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1403            002000          MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
```

```
                        004000          PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
                        010000          PCD=    BIT12                   ;PRECOMPENSATION DELAY
                        020000          ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
                        040000          WRTGAT= BIT14                   ;WRITE GATE
                        100000          RDGATE= BIT15                   ;READ GATE

                                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MSG A (RKMR2:34)

                        000040          D.DRA=  BIT5                    ;DRIVE AVAILABLE
                        000100          D.VV=   BIT6                    ;VOLUME VALID
                        000200          D.DRDY= BIT7                    ;DRIVE READY
                        000400          D.DDT=  BIT8                    ;DRIVE TYPE (0=RK06,1 = RK07)
                        001000          D.FORM= BIT9                    ;DRIVE FORMAT
                        002000          D.OFF=  BIT10                   ;OFFSET ON
                        004000          D.WRL=  BIT11                   ;WRITE LOCK
                        010000          D.SPIN= BIT12                   ;SPINDLE ON
                        020000          D.PIP=  BIT13                   ;POSITIONING IN PROGRESS
                        040000          D.DSC=  BIT14                   ;DRIVE STATUS CHANGE

                                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MSG A (RKMR2:34)

                        000020          D.SSP=  BIT4                    ;SERVO SIG PRESENT
                        000040          D.HDHM= BIT5                    ;HEADS HOME
                        000100          D.BRHM= BIT6                    ;BRUSHES HOME
                        000200          D.DOOR= BIT7                    ;DOOR INTERLOCKED
                        000400          D.CART= BIT8                    ;CARTRAGE INTERLOCK
                        001000          D.SPOK= BIT9                    ;SPEED OK
                        002000          D.FWD=  BIT10                   ;FORWARD
                        004000          D.REV=  BIT11                   ;REVERSE
                        010000          D.LOAD= BIT12                   ;HEADS LOADING
                        020000          D.RTZ=  BIT13                   ;RETURN TO ZERO
                        040000          D.UNLD= BIT14                   ;HEADS UNLOADING

                                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MSG B (RKMR3:36)

                        000040          D.IDAE= BIT5                    ;INVALID DISK ADDRESS ERROR:HEAD/CYL
                        000100          D.ACLO= BIT6                    ;AC LOW
                        000200          D.FLT=  BIT7                    ;DRIVE FAULT
                        000400          D.ILF=  BIT8                    ;ILLEGAL FUNCTION CODE
                        001000          D.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
                        002000          D.SKI=  BIT10                   ;SEEK INCOMPLETE
                        004000          D.WLE=  BIT11                   ;WRITE LOCK ERROR
                        010000          D.SPLS= BIT12                   ;SPEED LOSS
                        020000          D.DROT= BIT13                   ;DRIVE OFF TRACK
                        040000          D.UNS=  BIT14                   ;R/W UNSAFE

                                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MSG B (RKMR3:36)

                        000020          D.SECT= BIT4                    ;SECTOR ERROR
                        000040          D.WCUR= BIT5                    ;WRITE CURRENT AND NO WRITE GATE
                        000100          D.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
                        000200          D.HDFL= BIT7                    ;HEAD FAULT
                        000400          D.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
                        001000          D.XERROR=       BIT9                    ;INDEX ERROR
                        002000          D.TIB=  BIT10                   ;TRIBIT ERROR
                        004000          D.PLO=  BIT11                   ;PLO ERROR
```

```
1460    010000          D.NMOV= BIT12                   ;SEEK AND NO MOTION
1461    020000          D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1462    040000          D.SUNS= BIT14                   ;SERVO UNSAFE
1463
1464                    .SBTTL  COMMON MASKS AND OTHER BITS: MSG A (RKMR2:34)
1465
1466    000007          M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1467    077770          M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1468
1469                    .SBTTL  COMMON MASKS AND OTHER BITS: MSG B (RKMR3:36)
1470
1471    000003          M.ID=   3                       ;BYTE ID, ALL BYTES
1472    040000          M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1473    000760          M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1474    007000          M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1475    100000          M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
                                .SBTTL  TRAP CATCHER

                        000000          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                        000174          .=174
000174  000000        DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000        SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  007120          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                        000204  .=204
000204  000137  007014          JMP     BYT16           ;BYPASS N-SQUARE TEST IN DEFAULT MODE
                        000220  .=220
000220  000137  006774          JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING
                        000230  .=230
000230  000137  007034          JMP     BYT16A  ;BYPASS N-SQUARE TEST IN PARAM MODE
                        000240  .=240
000240  000137  070144          JMP     0.ODT           ;ENTER ODT11
                        000260  .=260
000260  000137  007056          JMP     MDTST           ;MODULE TESTS DEFAULT MODE ONLY
                        000270  .=270                   ;BYPASS SEVERAL TESTS
000270  000137  007076          JMP     MDTSTA          ;SAME AS 260 & BYPASS N-SQUARE TEST ALSO

                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000274          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  043046                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
000052  100000                  .WORD   100000          ;;2)SET LOC.52 TO 100000
                        000274          .=$SVPC                 ;; RESTORE PC
                        001000          .=1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;**************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**************************************************************
                        001000          .$X=.   ;;SAVE CURRENT LOCATION
                        000024          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                  200     ;;FOR APT START UP
                        000044          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                  $APTHDR ;;POINT TO APT HEADER BLOCK
                        001000          .=.$X   ;;RESET LOCATION COUNTER
                                ;;**************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                  $APTHD:
001000  000000          $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001210          $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000430          $TSTM:  .WORD   280.    ;;RUN TIM OF LONGEST TEST
001006  001130          $PASTM: .WORD   600.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
```

```
1532  001010  001130                $UNITM: .WORD   600.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1533  001012  000042                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1534
1535
1536                                        .LIST   MD
1537                                ;
1538                                ;USE LOOP X TO OMIT JSR PC, SUBCLR
1539                                ;
1540
1541                                .MACRO  LOOP    A
1542                                        SCOP1
1543                                        MOV     #STACK,SP       ;RESTORE STK PTR
1544
1545                                .IF     B       A
1546                                        JSR     PC,SUBCLR
1547                                        ERROR   24              ;CERR AFTER SCLR
1548
1549                                .ENDC
1550                                .ENDM   LOOP
1551
1552                                ;
1553                                ; THIS MACRO FILLS EXPECTED MSG A0,B0,A1,B1,A2,B2 & B3 WITH STANDARD BITS SET
1554                                ; A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1555                                ; NOTE: A CAN BE ANY BIT COMBINATION DESIRED
1556
1557                                .MACRO  F.EAB   A
1558
1559                                        MOV     #<A!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
1560                                        CLR     E.B0            ;EXPECTED MSG B0
1561                                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
1562                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
1563                                        CLR     E.A2            ;EXPECTED MSG A2
1564                                        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
1565                                        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
1566                                .ENDM   F.EAB
1567                                ;
1568                                ; THIS MACRO ASSUMES DRIVE MSG A0,B0,A1,B1 WILL ALWAYS BE TESTED
1569                                ; USE   A,C,D,E FOR MSG A0,B0,A1,B1 ERROR NUMBERS RESP.
1570                                ; USE   G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1571                                ;       H=T.B2  TO READ MSG B2 & PUT INFOR INTO 'CYLADD'
1572                                ;       I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
1573                                ;
1574                                ;       F=      < ERROR DESCRIPTION>
1575                                ;
1576                                .MACRO  CHECK   A,C,D,E,F,G,H,I
1577
1578                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
1579                                        .WORD   G!H!I           ;& MSGS SPECIFIED HERE
1580                                        ERROR   A               ;MSG A0 ERROR F
1581                                        ERROR   C               ;MSG B0 ERROR
1582                                        ERROR   D               ;MSG A1 ERROR
1583                                        ERROR   E               ;MSG B1 ERROR
1584                                .ENDM   CHECK
1585                                ;
1586                                ; A=CYL DIFF/OFFSET ERROR #
1587                                ; B=CYL ADDR ERROR #
```

```
                        ; C= <ERROR DESCRIPTION>
                        ;
                        .MACRO  CWD2    A,B,C,?D,?E

                                MOV     #2,RKMR1(R5)    ;SELECT WORD 2
                                JSR     PC,GSTAT
                                TST     CYLDIF          ;SEE IF MSG A2=0
                                BEQ     D               ;BR IF YES
                                ERROR   A               ;MSG A2 NOT CLEARED C
                        D:      TST     CYLADD          ;SEE IF MSG B2=0
                                BEQ     E               ;BR IF YES
                                ERROR   B               ;MSG B2 NOT CLEARED C
                        E:
                        .ENDM   CWD2


                        .MACRO  LPCHK   ?A
                                CLR     $ESCAPE
                                TST     LPFLG
                                BEQ     A
                                JMP     @$LPERR         ;SW 9 WAS SET.
                        A:      JMP     @$LPADR         ;SW 14 OR 8 WAS SET
                        .ENDM   LPCHK

                        .MACRO  SW814
                                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
                                SKIP R,<GO TO NEXT TEST>        ;RETURN HERE IF NEITHER IS SET
                                                        ;RETURN HERE IF SW 14 IS SET OR
                                                        ;SW 8 WITH SWR <7:0> APPLY
                        .ENDM   SW814

                        ;
                        ;SWR9 (LOOP ON ERROR) TEST      A=BRANCH POINT TO RECONDITION DRIVE
                        ;                               B=JMP POINT TO RE-ENTER MAIN LINE
                        .MACRO  TSTSW9  A,B
                                INC     LPFLG
                                BIT     #SW9,@SWR       ;LOOP ON ERROR?
                                BNE     A               ;YES, RECONDITION DRIVE
                                JMP     B               ;RETURN TO MAINLINE
                        .ENDM   TSTSW9


                        ;
                        ; USE DRCLR     X    TO OMIT CHECKING MSG A0,B0,A1 & B1
                        ;
                        .MACRO  DRCLR   A,?C

                                MOV     #CCLR,RKCS1(R5)
                                MOV     $UNIT,RKCS2(R5) ;DRIVE#
                                MOV     #CLEAR,HCS1
                                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
                                ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                                JSR     PC,TSTATN       ;TEST FOR ATTN
                                BR      C
                                ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                        C:
```

```
1644                            .IF B   A
1645                                    F.EAB   0
1646                                    CHECK   273,265,274,266,<AFTER DRIVE CLEAR CMD>,T.A2,T.B2,0
1647                            .ENDC
1648
1649                            .ENDM   DRCLR
1650
1651
1652                            ;
1653                            ;A=BLANK TO CHECK A0 THRU B2
1654                            ;A=NON BLANK TO OMIT CHECKING A0 THRU B2
1655                            ;D=BLANK TO CHECK A0 THRU B2 IN DRCLR
1656                            ;D=NON-BLANK TO OMIT CHECKING A0 THRU B2 IN DRCLR
1657                            ;THE FOLLOWING MACRO DEFINITION IS CHANDGED ON 24-OCT-77
1658                            ;       1. ADD NEW LABEL ?F
1659                            ;       2. DEL THE MACRO CALL CWD2
1660                            ;       3. ADD 5 LINES LONG HAND CODING FOR SUBSTITUTING ITEM 2
1661                            ;       4.  SELECT MESSAGE 2
1662                            ;
1663                            .MACRO  CALIB   A,D,?C,?F
1664
1665                                    MOV     #CCLR,RKCS1(R5)
1666                                    MOV     $UNIT,RKCS2(R5)
1667                                    MOV     #RECAL,HCS1
1668                                    JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
1669                                    ERROR   124             ;RDY NOT SET AFTER RECAL CMD
1670
1671                                    MOV     #1,RKMR1(R5)    ;SELECT WORD 1
1672                                    JSR     PC,GSTAT
1673                                    BIT     #D.RTZ,HMR2
1674                                    BNE     C
1675                                    ERROR   244             ;RTZ NOT SET DURING RECAL CMD
1676                            C:      MOV     T10,TEMP2       ;SETUP TIMEOUT
1677                                    JSR     PC,FATT1        ;FIND ATTN
1678                                    ERROR   55              ;NO ATTN AFTER RECAL CMD
1679                            .IF B   A
1680                                    F.EAB   D.DSC
1681                                    CHECK   221,275,222,276,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
1682                            ;       CWD2    47,50,<AFTER RECAL CMD>
1683                                    MOV     #2,RKMR1(R5)    ;SELECT THE MESSAGE
1684                                    JSR     PC,GSTAT        ;GET THE STATUS
1685                                    TST     CYLADD          ;RECAL SUCCESSFUL ?
1686                                    BEQ     F               ;BRANCH IF SO
1687                                    ERROR   50              ;REPORT THE ERROR
1688                            F:
1689                            .ENDC
1690                                    DRCLR   D
1691
1692                            .ENDM   CALIB
1693
1694                            ;
1695                            ;IDAE IS CLEARED ONLY BY RECAL & DRIVE CLEAR
1696                            ;
1697                            .MACRO  CIDAE   ?A
1698
1699                                    MOV     #CCLR,RKCS1(R5)
```

```
1700                                    MOV     $UNIT,RKCS2(R5)
1701                                    MOV     #RECAL,HCS1
1702                                    JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
1703                                    ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
1704                                    DRCLR   X
1705
1706                                    JSR     PC,GSTAT
1707                                    BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
1708                                    BEQ     A               ;BR IF YES
1709                                    ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
1710
1711                                A:  MOV     #CCLR,RKCS1(R5)
1712                                    MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
1713                                    JSR     PC,FATT1
1714                                    ERROR   55              ;NO ATTN AFTER RECAL CMD
1715
1716                                .ENDM   CIDAE
1717
1718                                ;
1719                                ; A=D.FWD/D.REV
1720                                ;
1721                                .MACRO  SKRDY A
1722
1723                                        MOV     #SEEK,HCS1
1724                                        JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
1725                                        ERROR   131             ;NO RDY AFTER SEEK CMD
1726                                        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
1727                                        CLR     E.B0
1728                                        MOV     #<A!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
1729                                        MOV     #1,E.B1
1730                                        CHECK   203,204,205,206,<DURING SEEK CMD>,T.A2,T.B2,0
1731
1732                                .ENDM   SKRDY
1733
1734                                .MACRO  SKATN ?A,?B
1735
1736                                        JSR     PC,FATT2        ;FIND ATTN
1737                                        ERROR   132             ;NO ATTN AFTER SEEK CMD
1738                                        BIT     #CERR,HCS1
1739                                        BEQ     A
1740                                        ERROR   210             ;CERR AFTER SEEK CMD
1741                                A:      F.EAB   D.DSC
1742                                        CHECK   133,134,135,136,<AFTER SEEK CMD>,T.A2,T.B2,0
1743                                        TST     CYLDIF
1744                                        BEQ     B
1745                                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
1746
1747                                B:      DRCLR
1748                                .ENDM   SKATN
1749
1750                                ;
1751                                ;QUICK START SPINDLE.
1752                                ;
1753                                .MACRO  QKSRT   A
1754
1755                                        JSR     PC,SUBCLR
```

```
1756                                    ERROR   24              ;CERR AFTER SCLR
1757
1758                                    MOV     #SRTSPL,HCS1
1759                                    JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
1760                                    ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
1761
1762                                    MOV     T500,TEMP2      ;SETUP  TIMEOUT
1763                                    JSR     PC,FATT1        ;FIND ATTN
1764                                    ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
1765
1766                                    CLR     UNLD
1767                            .IF B   A
1768                                    TSTSW9  10$,2$
1769                            .ENDC
1770                            .ENDM   QKSRT
1771
1772                            ;
1773                            ;QUICK SEEK.    ENTER WITH CYL # IN RKDC
1774                            ;
1775                            .MACRO  QKSEEK  ?A
1776
1777                                    MOV     #SEEK,HCS1
1778                                    JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
1779                                    ERROR   131             ;NO RDY AFTER SEEK CMD.
1780
1781                                    MOV     T50000,TEMP1
1782                                    JSR     PC,FATT2        ;FIND ATTN
1783                                    ERROR   132             ;NO ATTN AFTER SEEK CMD
1784                                    BIT     #CERR,HCS1
1785                                    BEQ     A
1786                                    ERROR   210             ;CERR AFTER SEEK CMD.
1787
1788                            A:      JSR     PC,SUBCLR
1789                                    ERROR   24              ;CERR AFTER SCLR
1790
1791                            .ENDM   QKSEEK
1792
1793                            ;
1794                            ;QUICK REPETITIVE SEEKS
1795                            ;       A=INC/DEC    CYL#
1796                            ;       B=FINAL VALUE OF CYL# BEFORE EXITING
1797
1798                            .MACRO  QKRPSK  A,B,?C,?D
1799
1800                                    JSR     PC,SUBCLR
1801                                    ERROR   24              ;CERR AFTER SCRL
1802
1803                            C:      MOV     TOCYL,RKDC(R5)  ;CYL#
1804                                    QKSEEK
1805                                    CMP     TOCYL,B ;LAST CYL DONE?
1806                                    BEQ     D               ;BR IF YES
1807                                    A       TOCYL           ;ELSE DO ANOTHER
1808                                    BR      C
1809
1810                            D:      JSR     PC,SUBCLR
1811                                    ERROR   24              ;CERR AFTER SCLR
```

```
1812
1813                                    LPCHK
1814
1815                                .ENDM   QKRPSK
1816
1817                                ;
1818                                ;QUICK UNLOAD
1819                                ;D=BLANK TO DO SUBCLR & LPCHK
1820                                ;D=NON-BLANK TO BYPASS
1821                                ;
1822                                .MACRO  QKUNLD  D
1823
1824                                        JSR     PC,SUBCLR
1825                                        ERROR   24              ;CERR AFTER SCLR
1826
1827                                        MOV     #UNLOAD,HCS1
1828                                        JSR     PC,DOCMD        ;DO UNLOAD CMD & GET CONTR READY
1829                                        ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
1830                                        JSR     PC,TSTATN
1831                                        ERROR   12              ;NO ATTN AFTER UNLOAD CMD
1832
1833                                        JSR     PC,SUBCLR
1834                                        ERROR   24              ;CERR AFTER SCLR
1835
1836                                        MOV     T10,TEMP2
1837                                        JSR     PC,FSPOK
1838                                        ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
1839                                .IF     B       D
1840
1841                                        JSR     PC,SUBCLR
1842                                        ERROR   24              ;CERR AFTER SCLR
1843
1844                                        LPCHK
1845                                .ENDC
1846
1847
1848
1849                                .ENDM   QKUNLD
1850
1851                                ;
1852                                ; A=WRHEAD/<CFMT!WRHEAD>
1853                                ; USE WRHDR     <A>,X   TO OMIT CHECKING A0,B0,A1,B1
1854                                ;
1855                                .MACRO  WRHDR   A,C,?D
1856
1857                                        MOV     #<A>,HCS1
1858                                        JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
1859                                        ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
1860                                        JSR     PC,GSTAT        ;GET FRESH STATUS
1861                                        BIT     #CERR,HCS1
1862                                        BEQ     D
1863                                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
1864                                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
1865                                        JMP     $EOP            ;ABORT DRIVE
1866                                D:
1867                                .IF B   C
```

```
                        F.EAB   0
                        CHECK   277,267,300,270,<AFTER WRITE HEADER CMD>,T.A2,T.B2,0
                .ENDC

                .ENDM   WRHDR

                ;
                ; A=RDHEAD/<CFMT!RDHEAD>
                ; USE RDHDR <A>,X TO OMIT CHECKING A0,B0,A1,B1
                ;
                .MACRO  RDHDR   A,C,?D,?E

                        MOV     #RHTAB,R0
                        MOV     #<A>,HCS1
                        JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
                        BIT     #CERR,HCS1
                        BEQ     D
                        ERROR   174             ;CERR AFTER READ HEADER CMD
                        TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE

                D:      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)
                        BEQ     E
                        JSR     PC,GSTAT
                        ERROR   173             ;DLT AFTER READ HEADER CMD
                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE
                E:
                .IF     B       C
                        F.EAB   0
                        CHECK   301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
                .ENDC

                .ENDM   RDHDR

                ;
                ;A=TOCYL/FRCYL
                ;B=310 FOR TOCYL/311 FOR FRCYL
                ;

                .MACRO  HDCHK3  A,B,?C

                        RDHDR   RDHEAD,X
                        CMP     RHTAB,A ;CHECK WORD 0 (CYL#) ONLY
                        BEQ     C               ;BR IF SAME
                        ERROR   B               ;READ CYL WORD HEADER ERROR
                C:

                .ENDM   HDCHK3
```

```
1924                            .MACRO  RALLHD  ?A,?B,?C,?D,?E
1925
1926                                    MOV     #RHTAB,R0
1927
1928                            A:      MOV     #RDHEAD,HCS1
1929                                    JSR     PC,DATCMD       ;DO READ HEADER CMD & GET CONTR RDY
1930                                    ERROR   171             ;NO RDY AFTER READ HEADER CMD
1931                                    BIT     #CERR,HCS1
1932                                    BEQ     B
1933                                    ERROR   174             ;CERR AFTER READ HEADER  CMD
1934                                    TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
1935                                    JMP     $EOP            ;ABORT DRIVE
1936
1937                            B:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
1938                                    MOV     RKDB(R5),(R0)+  ;2'ND WORD
1939                                    MOV     RKDB(R5),(R0)+  ;3'RD WORD
1940
1941                                    BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
1942                                    BEQ     C
1943                                    JSR     PC,GSTAT
1944                                    ERROR   173             ;DATA LATE ON READ HEADER
1945                                    TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
1946                                    JMP     $EOP            ;ABORT DRIVE
1947
1948                            C:      CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
1949                                    BNE     A               ;BR IF NO
1950
1951                                    JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
1952                                                            ;BEGINS WITH SECTOR 0
1953                                    CLR     WDCNT           ;WORD COUNT
1954                                    MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
1955                                    MOV     #HDTAB,R1       ;CALC HEADER TABLE
1956
1957                            D:      MOV     (R0)+,HDWD
1958                                    MOV     (R1)+,TEMP1
1959                                    CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
1960                                    BEQ     E               ;BR IF COMPARE
1961                                    ERROR   202             ;READ HEADER MISMATCH
1962
1963                            E:      INC     WDCNT
1964                                    CMP     WDCNT,#66.      ;ALL WORDS DONE?
1965                                    BNE     D               ;BR IF NO
1966
1967                            .ENDM   RALLHD
1968
1969
1970
1971                            ;
1972                            ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1973                            ;
1974                            .MACRO  HDTBL   A,B,C
1975
1976                                    MOV     A,CALADD        ;SETUP
1977                                    MOV     #B,HEAD         ;TO FILL
1978                                    MOV     #C,FORMAT       ;HEADER
1979                                    JSR     PC,FHDTAB       ;TABLE
```

```
1980
1981                            .ENDM   HDTBL
1982                            ;
1983                            ;USE FSECA FS022,RDSEC,22 FOR 22 SECTOR FORMAT
1984                            ;USE FSECA FS020,R20SEC,20 FOR 20 SECTOR FORMAT.
1985                            ;
1986                            .MACRO  FSECA   A,B,C
1987
1988                            ;
1989                            ;FIND SECTOR 0 IN C SECTOR FORMAT.
1990                            ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
1991                            ;
1992                            A:      MOV     TEMP1,-(SP)     ;SAVE TEMP1
1993                                    MOV     T5000,TEMP1     ;SETUP TIMEOUT
1994                            1$:     JSR     PC,B            ;READ SECTOR
1995                                    TST     SECTOR          ;LOOK FOR SECTOR C
1996                                    BNE     2$
1997                                    JSR     PC,B
1998                                    TST     SECTOR
1999                                    BEQ     3$              ;BR IF SAME TWICE
2000                            2$:     DEC     TEMP1
2001                                    BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
2002                                    MOV     (SP)+,TEMP1     ;ELSE RESTORE TEMP1
2003                                    RTS     PC              ;EXIT
2004                            3$:     MOV     (SP)+,TEMP1
2005                                    ADD     #2,(SP)         ;SKIP OVER ERROR
2006                                    RTS     PC
2007                            .ENDM
2008                            ;
2009                            ;USE FSECB FNS22,RDSEC,22 FOR 22 SECTOR FORMAT
2010                            ;USE FSECB FNS20,R20SEC,20 FOR 20 SECTOR FORMAT
2011                            ;
2012                            .MACRO  FSECB   A,B,C
2013
2014                            ;
2015                            ;FIND NEXT SECTOR IN C SECTOR FORMAT
2016                            ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
2017                            ;
```

```
                        A:      MOV     TEMP1,-(SP)     ;SAVE TEMP 1
                                MOV     T500,TEMP1      ;SETUP TIMEOUT
                        1$:     JSR     PC,B            ;READ SECTOR
                                CMP     PSEC,SECTOR
                                BEQ     3$              ;BR IF SAME
                                JSR     PC,B            ;ELSE TRY READ DIFFERENT TWICE
                                CMP     PSEC,SECTOR
                                BNE     2$              ;BR IF DIFFERENT TWICE
                        3$:     DEC     TEMP1           ;ELSE TRY AGAIN IF TIME LEFT
                                BNE     1$
                                MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
                                RTS     PC
                        2$:     MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
                                ADD     #2,(SP)         ;SKIP OVER ERROR
                                RTS     PC
                        .ENDM

                        ;
                        ;USE SECTST FS022,FNS22,RDSEC FOR 22 SECTOR FORMAT
                        ;USE SECTST FS020,FNS20,R20SEC FOR 20 SECTOR FORMAT
                        ;
                        ;THE FOLLOWING MACRO DEFINTION IS CHANGED ON 24-OCT-77
                        ; 1. DETET 8 LINES FROM THE CODING
                        .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: FCP1,FC,D.REV,OUTER
                        ;DETECT INNER LIMIT: LCM1,LC,D.FWD,INNER
                        ;
                        ;THE FOLLOWING MACRO DEFINITION IS CHANGED ON 24-OCT-77
```

```
                        ;1. ADD NEW LOCAL LABEL ?F
                        ;2. DELET CHECK MACRO CALL ON LIMIT DETECT
                        ;3. DEL CDW2 MCARO CALL
                        .MACRO  LIMIT A,B,C,D,?F

                                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)
                                MOV     #SEEK,HCS1
                                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
                                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)
                                MOV     #SEEK,HCS1
                                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
                                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$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
                                MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
                                MOV     #<D.SKI!D.FLT>,E.B0
                                MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
                                MOV     #<D.LIMD!D.NMOV!1>,E.B1
                        ;       CHECK   161,162,163,164,<AFTER D LIMIT DETECT>,0,0,0
                                JSR     PC,CHKMSG       ;CHECK MESSAGE A0,B0,A1,B1
                                .WORD   0!0!0
                                ERROR   161             ;MAY BE A0 ERROR
                                ERROR   162             ;MAY BE B0 ERROR
                                ERROR   163             ;MAY BE B1 ERROR
                                NOP                     ;NEED FOR THE CALLIN SEQ
                                BIT     #D.LIMD,H.B1    ;SEE IF LIMIT DETECT BIT SET IN B1
                                BNE     3$              ;BRANCH IF SO
                                TYPE    ,EM42           ;OTHERWISE REPORT ERROR
                                ERROR   164
```

```
2130
2131                            3$:     JSR     PC,TSTATN
2132                                    ERROR   165             ;NO ATTN AFTER D LIMIT DETECT
2133                                    CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1
2134
2135                                    JSR     PC,SUBCLR       ;SUBSYS CLR
2136                                    ERROR   24              ;CERR AFTER SCLR
2137                                    MOV     T10,TEMP2       ;SET UP TIMEOUT
2138                                    JSR     PC,FHDHM        ;FIND HEAD HOME
2139                                    ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
2140                                    JSR     PC,FLOAD        ;FIND LOAD HEADS
2141                                    ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
2142                                    MOV     T100,TEMP2      ;SETUP TIMEOUT
2143                                    JSR     PC,FATT1        ;FIND ATTN
2144                                    ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
2145                            2$:     CLR     $ESCAPE
2146                                    CLR     UNLD            ;CLEAR FLAG
2147                                    F.EAB   D.DSC
2148                                    CHECK   63,64,65,66,<AT END OF HEAD LOADING>,T.A2,T.B2,0
2149                            ;       CWD2    175,176,<AT END OF HEAD LOADING>
2150                                    MOV     #2,RKMR1(R5)    ;SELECT MESSAGE
2151                                    JSR     PC,GSTAT        ;GET STATUS AND MR2,MR3
2152                                    TST     CYLADD          ;RECAL SUCCFUL ?
2153                                    BEQ     F               ;BRANCH IF SO
2154                                    ERROR   50              ;REPORT ERROR
2155                            F:      DRCLR
2156                                    SW814
2157
2158
2159                            .ENDM   LIMIT
2160
2161
2162                            ;
2163                            ; A=CYL#, B=HEAD#
2164                            ;
2165                            .MACRO  HEADER  A,B
2166
2167                                    NEWTST <<WRITE & READ HEADERS CYL A, HEAD B>>,1
2168                                    MOV     #STACK,SP       ;RESTORE STK PTR
2169
2170                                    JSR     PC,SUBCLR
2171                                    ERROR   24              ;CERR AFTER SCLR
2172
2173                                    INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
2174
2175                                    MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
2176                                    MOV     #-66.,RKWC(R5)  ;WORD COUNT.
2177                                    MOV     #A,TOCYL
2178                                    HDTBL   TOCYL,0,0
2179                                    MOV     #A,RKDC(R5)     ;CYL#
2180                                    WRHDR   WRHEAD
2181                                    CLR     SECNT           ;SECTOR COUNT
2182                                    LOOP
2183                                    MOV     #A,RKDC(R5)     ;CYL #
2184                                    RALLHD
2185
```

```
                        CLR BYPFMT              ;ALLOW CORRECT FORMATTING

                .ENDM HEADER



                ;
                ;SEEK TO MAJOR CYL: FC,FCP1,TEMP3,TEMP4,D.FWD,D.REV,ASL,MC,DEC,FC
                ;SEEK 0 TO ALL CYL: FC,FCP1,TEMP3,TEMP4,D.FWD,D.REV,INC,LC,DEC,FC
                ;SEEK 410 TO ALL CYL: LC,LCM1,TEMP4,TEMP3,D.REV,D.FWD,DEC,FC,INC,LC
                ;
                .MACRO  SKOSC   A,B,C,D,E,F,G,H,I,J

                        MOV     A,FRCYL ;SETUP FROM CYL
                        MOV     B,TOCYL ;SETUP TO CYL

                1$:     LOOP
                        MOV     #10$,$ESCAPE
                        MOV     FRCYL,TEMP3     ;SETUP
                        MOV     TOCYL,TEMP4     ;CYL DIFF
                        SUB     C,D     ;FOR
                        MOV     D,CALDIF        ;ERROR PRINTOUT

                        MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
                        SKRDY   E
                2$:     MOV     #12$,$ESCAPE
                        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
                        SKATN
                        CMP     CYLADD,TOCYL
                        BEQ     3$
                        ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC

                3$:     LOOP
                        CLR     $ESCAPE
                        MOV     TOCYL,RKDC(R5)  ;CYL #
                        HDCHK3  TOCYL,310

                        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?
```

```
2242                                    BEQ     6$              ;BR IF YES
2243                                    G       TOCYL           ;ELSE DO ANOTHER
2244                                    JMP     1$
2245                            6$:     SW814
2246                            8$:     QKRPSK  I,J
2247                            10$:    TSTSW9  8$,2$
2248                            12$:    TSTSW9  8$,3$
2249                            14$:    TSTSW9  8$,4$
2250                            16$:    TSTSW9  8$,5$
2251                            .ENDM   SKOSC
2252
2253                            .MACRO  EOPGM
2254
2255                                    SCOPE
2256                                    CLR     $ESCAPE
2257                                    MOV     #1,$TIMES
2258                                    MOV     #STACK,SP
2259                                    INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
2260                                    CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
2261                                    BEQ     1$              ;BR IF YES
2262                                    CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
2263                                    JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
2264                            1$:     CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
2265                                    BR      $EOP1+2         ;GO TO $EOP1+2
2266                            $EOP1:  SCOPE
2267                            .ENDM   EOPGM
2268
2269                                    .NLIST  MD
```

```
                        .SBTTL  COMMON TAGS

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

                001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $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
```

```
 2326  001212  000000                $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
 2327  001214  000000                $TESTN: .WORD   ATESTN  ;;TEST NUMBER
 2328  001216  000000                $PASS:  .WORD   APASS   ;;PASS COUNT
 2329  001220  000000                $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
 2330  001222  000000                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
 2331  001224  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
 2332  001226  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
 2333  001230                        $ETABLE:                ;;APT ENVIRONMENT TABLE
 2334  001230     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
 2335  001231     000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
 2336  001232  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
 2337  001234  000000                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
 2338  001236  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
 2339                                ;*                      BITS 15-11=CPU TYPE
 2340                                ;*                             11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
 2341                                ;*                             11/70=06,PDQ=07,Q=10
 2342                                ;*                      BIT 10=REAL TIME CLOCK
 2343                                ;*                      BIT  9=FLOATING POINT PROCESSOR
 2344                                ;*                      BIT  8=MEMORY MANAGEMENT
 2345  001240     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
 2346  001241     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
 2347                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
 2348                                ;*                             900 NSEC CORE=001
 2349                                ;*                             300 NSEC BIPOLAR=002
 2350                                ;*                             500 NSEC MOS=003
 2351  001242  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
 2352                                ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
 2353  001244     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
 2354  001245     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
 2355  001246  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
 2356  001250     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
 2357  001251     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
 2358  001252  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
 2359  001254     000                $MAMS4: .BYTE   AMAMS4  ;.HIGH ADDRESS,M.S.BYTE
 2360  001255     000                $MTYP4: .BYTE   AMTYP4  ;.MEM.TYPE,BLK#4
 2361  001256  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
 2362  001260  000000                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
 2363  001262  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
 2364  001264  177440                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
 2365  001266  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
 2366  001270  000000                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
 2367  001272  000000                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
 2368  001274  000000                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
 2369  001276  000000                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
 2370  001300  000000                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
 2371  001302  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
 2372  001304  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
 2373  001306  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
 2374  001310  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
 2375  001312  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
 2376  001314                        $ETEND:
 2377                                .MEXIT
 2378          177440                ABASE=  177440                  ;DEFAULT BUSS ADDRESS
 2379  001314  000210                RKVEC:  210                     ;DEFAULT CONTROLLER INTERRUPT VECTOR
 2380  001316  000240                RKPRI:  PR5                     ;PRIORITY
 2381  001320  172540                PKS:    172540                  ;P-CLOCK STATUS REG
```

```
2382  001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
2383  001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
2384  001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.
2385
2386  001330  000100          LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
2387  001332  000104          PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.
2388
2389          000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
2390          172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
2391  001334  000000          TRAPPC: 0               ;PC FOR MEMORY CHECK ENABLE TRAP
2392
2393  001336  000000          PARAM:  0               ;1 FOR 220 OR 230 START, NO DEFAULT
2394  001340  000000          BYPT16: 0               ;1 FOR 210, 230, 270 START
2395  001342  000000          MODTST: 0               ;1 FOR 260 OR 270 START
2396  001344  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
2397
2398  001346  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
2399                                                  ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
2400  001350  000000          FRCYL:  0               ;FROM CYL
2401  001352  000000          TOCYL:  0               ;TO CYL
2402  001354  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
2403  001356  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
2404  001360  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
2405  001362  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
2406  001364  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
2407  001366  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
2408
2409  001370  000074          HZ:     60.             ;60 FOR 60 CPS
2410                                                  ;50 FOR 50 CPS
2411  001372  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
2412                                                  ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
2413  001374  000000          SEC:    0               ;SECOND COUNTER
2414  001376  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
2415  001400  000000          SECNT:  0               ;SECTOR COUNT
2416  001402  000000          PSEC:   0               ;PREVIOUS SECTOR
2417  001404  000000          ESEC:   0               ;EXPECTED SECTOR
2418  001406  000000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
2419
2420  001410  000000          LPFLG:  0               ;SET TO 0 TO RETURN TO $LPADR
2421                                                  ;IF SW14 OR SW8 SET
2422                                                  ;SET TO 1 TO RETURN TO $LPERR
2423                                                  ;IF SW9 SET
2424  001412  000001          T1:     1               ;TIMEOUT CONSTANTS
2425  001414  000012          T10:    10.
2426  001416  000144          T100:   100.
2427  001420  000764          T500:   500.
2428  001422  004704          T2500:  2500.
2429  001424  011610          T5000:  5000.
2430  001426  141520          T50000: 50000.
2431
2432  001430  000000          HEAD:   0               ;HEAD NUMBER
2433  001432  000000          HEADA:  0               ;HEAD # FROM H.B3 RIGHT JUSTIFIED
2434  001434  000000          HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
2435  001436  000000          FORMAT: 0               ;FORMAT TYPE
2436  001440  000000          FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
2437  001442  000000          WDCNT:  0               ;WORD COUNT
```

```
2438
2439  001444  000000                DATA0:  0                       ;ALL 0'S
2440  001446  052525                DATA01: 52525                   ;0101 PATT
2441  001450  177777                DATA1:  177777                  ;ALL 1'S
2442
2443  001452  000000                WORD:   0                       ;HEADER/DATA WORD
2444  001454  000000                HDWD:   0                       ;HEADER WORD FROM RKDB
2445
2446  001456  000000                BSERR:  0                       ;CANNOT READ BSE INFO WHEN SET
2447  001460  000000                LIMERR: 0                       ;LIMIT DETECT ERROR FLAG
2448
2449  001462  000000                BYPCERR:0                       ;SET TO 1 TO BYPASS CKCERR IN GSTAT1 ROUTINE
2450  001464  000000                BYPFMT: 0                       ;BYPASS FORMAL FORMATTING OF HEADERS
2451                                                                ;UNTIL BSE INFO HAS BEEN STORED.
2452                                                                ;IF SET, BIT 14,15 = 1
2453
2454  001466  000000                CHKFLG: 0                       ;WORDS TO BE CHECKED
2455
2456  001470  000102                HDTAB:  .BLKW 66.               ;CALCULATED HEADER WORD TABLE
2457  001674  000102                RHTAB:  .BLKW 66.               ;FILLED AFTER READ HEADER CMD
2458  002100  000102                SRTTAB: .BLKW 66.               ;ABOVE RHTAB SORTED STARTING FORM
2459                                                                ;SECTOR 0 BY SORT ROUTINE
2460  002304  000400                BSE22H: .BLKW 256.              ;22 SECTOR HARDWARE BSE INFO.
2461                                                                ;22 SECTOR SOFTWARE BSE INFO
2462                                                                ;OVERLAYS MSG1
2463  003304  000000                UNLD:   0                       ;SET TO 0 IF HEADS ARE LOADED
2464                                                                ;SET TO 1 IF HEADS UNLOADED
2465  003306  000000                BADHDR: 0                       ;SET TO 0 IF FORMATTING OK
2466                                                                ;SET TO 1 IF FORMATTING ALTERED
2467  003310  000000                HPEND:  0                       ;SET TO 0 IF HALT NOT PENDING
2468                                                                ;SET TO 1 IF HALT PENDING
2469
2470                                                                ;THE ABOVE 3 FLAGS ARE USED
2471                                                                ;BY 'STOP' ROUTINE TO BRING
2472                                                                ;THE CPU TO A VALID HALT.
2473
2474  003312    001     002     004 ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
2475  003315    010     020     040
2476  003320    100     200
2477                                .EVEN
2478
2479
2480                                ;
2481                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
2482                                ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
2483                                ;
2484
2485  003322  000000                HCS1:   0                       ;HOLD RKCS1
2486  003324  000000                HCS2:   0                       ;HOLD RKCS2
2487  003326  00000C                HWC:    0                       ;HOLD RKWC
2488  003330  000000                HBA:    0                       ;ETC.
2489  003332  000000                HDA:    0
2490  003334  000000                HDS:    0
2491  003336  000000                HER:    0
2492  003340  000000                HASOF:  0
2493  003342  000000                HDC:    0
```

```
2494  003344  000000          HDB:    0
2495  003346  000000          HMR1:   0
2496  003350  000000          HMR2:   0
2497  003352  000000          HMR3:   0
2498  003354  000000          HPOS:   0
2499  003356  000000          HPAT:   0
2500
2501  003360  000000          TEMP1:  0                       ;TEMPORARY STORAGE.
2502  003362  000000          TEMP2:  0
2503  003364  000000          TEMP3:  0
2504  003366  000000          TEMP4:  0
2505  003370  000000          TEMP5:  0
2506                          ;
2507                          ; THE FOLLOWING ARE HOLDING REGISTERS FOR MSG A (0-3) & MSG B (0-3)
2508                          ;
2509  003372  000000          H.A0:   0
2510  003374  000000          H.B0:   0
2511  003376  000000          H.A1:   0
2512  003400  000000          H.B1:   0
2513  003402  000000          H.A2:   0
2514  003404  000000          H.B2:   0
2515  003406  000000          H.A3:   0
2516  003410  000000          H.B3:   0
2517                          ;
2518                          ; THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE
2519                          ;
2520  003412  000000          E.A0:   0
2521  003414  000000          E.B0:   0
2522  003416  000000          E.A1:   0
2523  003420  000000          E.B1:   0
2524  003422  000000          E.A2:   0
2525  003424  000000          E.B2:   0
2526  003426  000000          E.A3:   0
2527  003430  000000          E.B3:   0
2528
2529                          ;
2530                          ; THE FOLLOWING ARE IDENTITIES FOR DRIVE MSG WORDS TO BE TESTED
2531                          ;
2532          000001          T.A2=BIT0                       ;TEST MSG A2 IF SET
2533          000002          T.B2=BIT1
2534          000004          T.B3=BIT2
2535
2536                          ;
2537                          ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
2538                          ;
2539
2540  003432  000000          DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
2541  003434  000000          DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
2542  003436  000000          ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
2543  003440  000000          PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
2544  003442  000000          DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
2545
2546                          ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
2547                          ;IS PRESENT AND IS TO BE TESTED.
2548
2549  003444  000000          DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
```

```
2550  003446  000000          DRIV1:  0               ;FOR DRIVE 1
2551  003450  000000          DRIV2:  0               ;FOR DRIVE 2
2552  003452  000000          DRIV3:  0               ;FOR DRIVE 3
2553  003454  000000          DRIV4:  0               ;FOR DRIVE 4
2554  003456  000000          DRIV5:  0               ;FOR DRIVE 5
2555  003460  000000          DRIV6:  0               ;FOR DRIVE 6
2556  003462  000000          DRIV7:  0               ;FOR DRIVE 7
2557
2558  003464  000000          LCLKF:  0               ;L-CLOCK FLAG PRESENT FLAG
2559  003466  000000          PCLKF:  0               ;P-CLOCK FLAG PRESENT FLAG
2560  003470  000000          DOTIM:  0               ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
2561  003472  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


003474                          $ERRTB:

                                ;ERROR 1
003474  057217                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
003476  064170                          DH1
003500  066276                          DT1
003502  067144                          DF1

                                ;ERROR 2
003504  057467                          EM5                     ;DETECTED MDS
003506  064170                          DH1
003510  066276                          DT1
003512  067144                          DF1

                                ;ERROR 3
003514  057510                          EM6                     ;DETECTED UFE
003516  064170                          DH1
003520  066276                          DT1
003522  067144                          DF1

                                ;ERROR 4
003524  057531                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
003526  064170                          DH1
003530  066276                          DT1
003532  067144                          DF1
                                ;ERROR 5
003534  057620                          EM8                     ;DDT NOT SET IN MR2
003536  064170                          DH1
003540  066276                          DT1
003542  067144                          DF1

                                ;ERROR 6
003544  057652                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
003546  064170                          DH1
003550  066276                          DT1
003552  067144                          DF1

                                ;ERROR 7
003554  057722                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
003556  064170                          DH1
003560  066276                          DT1
003562  067144                          DF1
```

```
2618
2619                                    ;ERROR 10
2620  003564  057765                          EM11            ;DRA & NED BOTH SET
2621  003566  064170                          DH1
2622  003570  066276                          DT1
2623  003572  067144                          DF1
2624
2625                                    ;ERROR 11
2626  003574  060031                          EM12            ;CONTROLLER NOT READY
2627  003576  064761                          DH18            ;AFTER UNLOAD CMD.
2628  003600  066276                          DT1
2629  003602  067270                          DF10
2630
2631                                    ;ERROR 12
2632  003604  060062                          EM13            ;NO ATTN
2633  003606  064761                          DH18            ;AFTER UNLOAD CMD
2634  003610  066276                          DT1
2635  003612  067270                          DF10
2636                                    ;ERROR 13
2637  003614  060104                          EM14            ;WRONG ATTN
2638  003616  064761                          DH18
2639  003620  066276                          DT1
2640  003622  067270                          DF10
2641                                    ;ERROR 14
2642  003624  060131                          EM15            ;DRDY NOT CLEARED
2643  003626  064761                          DH18
2644  003630  066276                          DT1
2645  003632  067270                          DF10
2646                                    ;ERROR 15
2647  003634  060163                          EM16            ;DSC NOT SET
2648  003636  064761                          DH18
2649  003640  066276                          DT1
2650  003642  067270                          DF10
2651                                    ;ERROR 16
2652  003644  060210                          EM17            ;MSG A0 ERROR
2653  003646  064361                          DH8             ;IN UNLD
2654  003650  066700                          DT13
2655  003652  067440                          DF20
2656                                    ;ERROR 17
2657  003654  060225                          EM18            ;MSG B0 ERROR
2658  003656  064361                          DH8             ;IN UNLD
2659  003660  066700                          DT13
2660  003662  067440                          DF20
2661                                    ;ERROR 20
2662  003664  060242                          EM19            ;MSG A1 ERROR
2663  003666  064361                          DH8             ;IN UNLD
2664  003670  066700                          DT13
2665  003672  067440                          DF20
2666                                    ;ERROR 21
2667  003674  060257                          EM20            ;MSG B1 ERROR
2668  003676  064361                          DH8             ;IN UNLD
2669  003700  066700                          DT13
2670  003702  067440                          DF20
2671                                    ;ERROR 22
2672  003704  061672                          EM46            ;MSG A2 ERROR
2673  003706  064361                          DH8             ;IN UNLD
```

```
2674  003710  066760                    DT14
2675  003712  067514                    DF22
2676                              ;ERROR 23
2677  003714  061705                    EM47          ;MSG B2 ERROR
2678  003716  064361                    DH8           ;IN UNLD
2679  003720  066760                    DT14
2680  003722  067514                    DF22
2681
2682                              ;ERROR 24
2683  003724  060274                    EM21          ;CERR SET
2684  003726  065050                    DH21          ;AFTER SCLR
2685  003730  066276                    DT1
2686  003732  067270                    DF10
2687                              ;ERROR 25
2688  003734  060316                    EM22          ;RLS DID NOT SET CERR
2689  003736  064170                    DH1
2690  003740  066276                    DT1
2691  003742  067144                    DF1
2692
2693                              ;ERROR 26
2694  003744  060355                    EM23          ;SACK SET AFTER RLS SENT
2695  003746  064170                    DH1
2696  003750  066276                    DT1
2697  003752  067144                    DF1
2698
2699                              ;ERROR 27
2700  003754  060435                    EM24          ;VOL VALID NOT SET
2701  003756  065002                    DH19          ;AFTER PACK CMD
2702  003760  066276                    DT1
2703  003762  067270                    DF10
2704                              ;ERROR 30
2705  003764  060461                    EM25          ;DRIVE TYPE SET IN MR2
2706  003766  064170                    DH1
2707  003770  066276                    DT1
2708  003772  067144                    DF1
2709                              ;ERROR 31
2710  003774  060507                    EM26          ;DDT SET IN RKDS
2711  003776  064170                    DH1
2712  004000  066276                    DT1
2713  004002  067144                    DF1
2714                              ;ERROR  32
2715  004004  060527                    EM27          ;DTYE SET IN RKER
2716  004006  064170                    DH1
2717  004010  066276                    DT1
2718  004012  067144                    DF1
2719                              ;ERROR 33
2720  004014  060550                    EM28          ;DTYE NOT SET IN RKER
2721  004016  064170                    DH1
2722  004020  066276                    DT1
2723  004022  067144                    DF1
2724                              ;ERROR 34
2725  004024  060626                    EM29          ;DTYE DID NOT SET CERR
2726  004026  064170                    DH1
2727  004030  066276                    DT1
2728  004032  067144                    DF1
2729                              ;ERROR 35
```

```
2730  004034  060675                EM30                ;C-D PARITY ERROR SET IN MR3
2731  004036  064170                DH1
2732  004040  066276                DT1
2733  004042  067144                DF1
2734                          ;ERROR 36
2735  004044  060726                EM31                ;D-C PARITY  SET IN CS1
2736  004046  064170                DH1
2737  004050  066276                DT1
2738  004052  067144                DF1
2739                          ;ERROR 37
2740  004054  060753                EM32                ;FAULT NOT SET IN MR3
2741  004056  064170                DH1
2742  004060  066276                DT1
2743  004062  067144                DF1
2744                          ;ERROR 40
2745  004064  061000                EM33                ;C-D PARITY ERROR NOT SET IN MR3
2746  004066  064170                DH1
2747  004070  066276                DT1
2748  004072  067144                DF1
2749                          ;ERROR 41
2750  004074  061035                EM34                ;D-C PARITY  NOT SET IN CS1
2751  004076  064170                DH1
2752  004100  066276                DT1
2753  004102  067144                DF1
2754                          ;ERROR 42
2755  004104  061066                EM35                ;DCPAR DID NOT SET CERR
2756  004106  064170                DH1
2757  004110  066276                DT1
2758  004112  067144                DF1
2759                          ;ERROR 43
2760  004114  061141                EM36                ;CYL ADDR IN B2 NOT = RKDC
2761  004116  064647                DH14                ;AFTER SEEK WITH BAD PARITY
2762  004120  066760                DT14
2763  004122  067514                DF22
2764                          ;ERROR 44
2765  004124  061201                EM37                ;CYL DIFF IN A2 NOT=RKDC
2766  004126  064647                DH14
2767  004130  066760                DT14
2768  004132  067514                DF22
2769                          ;ERROR 45
2770  004134  061141                EM36                ;CYL ADDR IN RKMR3 NOT=RKDC
2771  004136  064647                DH14
2772  004140  066336                DT4
2773  004142  067244                DF6
2774                          ;ERROR 46
2775  004144  061241                EM38                ;CYL DIFF IN RKMR2 NOT=CALDIF
2776  004146  064647                DH14
2777  004150  066336                DT4
2778  004152  067244                DF6
2779                          ;ERROR 47
2780  004154  061312                EM39                ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2781  004156  064741                DH17                ;AFTER RECAL CMD
2782  004160  066760                DT14
2783  004162  067514                DF22
2784                          ;ERROR 50
2785  004164  061347                EM40                ;CYL ADDR IN RKMR3 NOT CLEARED
```

```
2786  004166  064741                DH17                    ;AFTER RECAL COMD
2787  004170  066760                DT14
2788  004172  067514                DF22
2789                          ;ERROR 51
2790  004174  060210                EM17                    ;A0 ERROR
2791  004176  065164                DH26                    ;AFTER READ DATA CMD
2792  004200  066700                DT13
2793  004202  067440                DF20
2794                          ;ERROR 52
2795  004204  060225                EM18                    ;B0 ERROR
2796  004206  065164                DH26
2797  004210  066700                DT13
2798  004212  067440                DF20
2799                          ;ERROR 53
2800  004214  061466                EM43                    ;HEAD DECODE IN B3 NOT CLEARED
2801  004216  064741                DH17                    ;AFTER RECAL CMD
2802  004220  067050                DT15
2803  004222  067550                DF23
2804                          ;ERROR 54
2805  004224  061515                EM44                    ;B3 HEAD DECODE INCORRECT
2806  004226  064677                DH16
2807  004230  067050                DT15
2808  004232  067550                DF23
2809                          ;ERROR 55
2810  004234  060062                EM13                    ;NO ATTN
2811  004236  064741                DH17                    ;AFTER RECAL CMD
2812  004240  066276                DT1
2813  004242  067270                DF10
2814                          ;ERROR 56
2815  004244  062637                EM64                    ;MSG B3 HEAD REG NOT CLEARED
2816  004246  064361                DH8                     ;IN UNLOAD
2817  004250  067050                DT15
2818  004252  067550                DF23
2819                          ;ERROR 57
2820  004254  060210                EM17                    ;MSG A0 ERROR
2821  004256  064405                DH9                     ;AFTER START SPINDLE CMD REC'D BY DRIVE
2822  004260  066700                DT13
2823  004262  067440                DF20
2824                          ;ERROR 60
2825  004264  060225                EM18                    ;MSG B0 ERROR
2826  004266  064405                DH9
2827  004270  066700                DT13
2828  004272  067440                DF20
2829                          ;ERROR 61
2830  004274  060242                EM19                    ;MSG A1 ERROR
2831  004276  064405                DH9
2832  004300  066700                DT13
2833  004302  067440                DF20
2834                          ;ERROR 62
2835  004304  060257                EM20                    ;MSG B1 ERROR
2836  004306  064405                DH9
2837  004310  066700                DT13
2838  004312  067440                DF20
2839                          ;ERROR 63
2840  004314  060210                EM17
2841  004316  064451                DH10                    ;AT END OF HEAD LOADING
```

```
2842  004320  066700                    DT13
2843  004322  067440                    DF20
2844                            ;ERROR 64
2845  004324  060225                    EM18
2846  004326  064451                    DH10
2847  004330  066700                    DT13
2848  004332  067440                    DF20
2849                            ;ERROR 65
2850  004334  060242                    EM19
2851  004336  064451                    DH10
2852  004340  066700                    DT13
2853  004342  067440                    DF20
2854                            ;ERROR 66
2855  004344  060257                    EM20
2856  004346  064451                    DH10
2857  004350  066700                    DT13
2858  004352  067440                    DF20
2859                            ;ERROR 67
2860  004354  060062                    EM13          ;NO ATTN
2861  004356  064451                    DH10          ;AT END OF HEAD LOADING.
2862  004360  066276                    DT1
2863  004362  067270                    DF10
2864                            ;ERROR 70
2865  004364  062021                    EM50          ;FWD NOT SET WITHIN 60 SEC FROM
2866  004366  064170                    DH1           ;START SPINDLE CMD.
2867  004370  066276                    DT1
2868  004372  067144                    DF1
2869                            ;ERROR 71
2870  004374  060210                    EM17
2871  004376  064500                    DH11          ;AFTER START SPINDLE CMD & FWD SET.
2872  004400  066700                    DT13
2873  004402  067440                    DF20
2874                            ;ERROR 72
2875  004404  060225                    EM18
2876  004406  064500                    DH11
2877  004410  066700                    DT13
2878  004412  067440                    DF20
2879                            ;ERROR 73
2880  004414  060242                    EM19
2881  004416  064500                    DH11
2882  004420  066700                    DT13
2883  004422  067440                    DF20
2884                            ;ERROR 74
2885  004424  060257                    EM20
2886  004426  064500                    DH11
2887  004430  066700                    DT13
2888  004432  067440                    DF20
2889                            ;ERROR 75
2890  004434  062072                    EM51          ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
2891  004436  064170                    DH1           ;FROM START SPINDLE CMD.
2892  004440  066276                    DT1
2893  004442  067144                    DF1
2894                            ;ERROR 76
2895  004444  060210                    EM17
2896  004446  064537                    DH12          ;AT INNER LIMIT FROM START SPINDLE CMD.
2897  004450  066700                    DT13
```

```
 2898  004452  067440                    DF20
 2899                          ;ERROR 77
 2900  004454  060225                    EM18
 2901  004456  064537                    DH12
 2902  004460  066700                    DT13
 2903  004462  067440                    DF20
 2904                          ;ERROR 100
 2905  004464  060242                    EM19
 2906  004466  064537                    DH12
 2907  004470  066700                    DT13
 2908  004472  067440                    DF20
 2909                          ;ERROR 101
 2910  004474  060257                    EM20
 2911  004476  064537                    DH12
 2912  004500  066700                    DT13
 2913  004502  067440                    DF20
 2914                          ;ERROR 102
 2915  004504  061733                    EM49          ;FWD NOT SET WITHIN 4 SEC IN RTZ PORTION
 2916  004506  064170                    DH1           ;OF START SPIN CMD.
 2917  004510  066276                    DT1
 2918  004512  067144                    DF1
 2919                          ;ERROR 103
 2920  004514  060210                    EM17
 2921  004516  064600                    DH13          ;FROM OUTER LIMIT TO CYL 0 DURING LOADING
 2922  004520  066700                    DT13
 2923  004522  067440                    DF20
 2924                          ;ERROR 104
 2925  004524  060225                    EM18
 2926  004526  064600                    DH13
 2927  004530  066700                    DT13
 2928  004532  067440                    DF20
 2929                          ;ERROR 105
 2930  004534  060242                    EM19
 2931  004536  064600                    DH13
 2932  004540  066700                    DT13
 2933  004542  067440                    DF20
 2934                          ;ERROR 106
 2935  004544  060257                    EM20
 2936  004546  064600                    DH13
 2937  004550  066700                    DT13
 2938  004552  067440                    DF20
 2939                          ;ERROR 107
 2940  004554  061551                    EM45          ;DRIVE READY NOT SET WITHIN 1 SEC
 2941  004556  064170                    DH1           ;FROM FWD IN RTZ PORTION OF START SPIN CMD.
 2942  004560  066276                    DT1
 2943  004562  067144                    DF1
 2944                          ;ERROR 110
 2945  004564  060210                    EM17          ;MSG A0 ERROR
 2946  004566  064647                    DH14          ;AFTER SEEK WITH BAD PARITY
 2947  004570  066760                    DT14
 2948  004572  067514                    DF22
 2949                          ;ERROR 111
 2950  004574  060225                    EM18          ;MSG B0 ERROR
 2951  004576  064647                    DH14          ;AFTER SEEK WITH BAD PARITY
 2952  004600  066760                    DT14
 2953  004602  067514                    DF22
```

```
2954
2955  004604  060242                ;ERROR 112
                                        EM19                    ;A1 ERROR
2956  004606  065164                   DH26                    ;AFTER READ DATA CMD
2957  004610  066700                   DT13
2958  004612  067440                   DF20
2959                                ;ERROR 113
2960  004614  060257                   EM20                    ;B1 ERROR
2961  004616  065164                   DH26
2962  004620  066700                   DT13
2963  004622  067440                   DF20
2964                                ;ERROR 114
2965  004624  060210                   EM17
2966  004626  064677                   DH16                    ;AFTER LOADING HEAD REGISTER & SEEK CMD
2967  004630  066700                   DT13
2968  004632  067440                   DF20
2969                                ;ERROR 115
2970  004634  060225                   EM18
2971  004636  064677                   DH16
2972  004640  066700                   DT13
2973  004642  067440                   DF20
2974                                ;ERROR 116
2975  004644  060031                   EM12                    ;CONT NOT RDY
2976  004646  065002                   DH19                    ;AFTER PACK CMD
2977  004650  066276                   DT1
2978  004652  067270                   DF10
2979                                ;ERROR 117
2980  004654  060031                   EM12                    ;CONT NOT RDY
2981  004656  065021                   DH20                    ;AFTER SEL DR CMD
2982  004660  066276                   DT1
2983  004662  067270                   DF10
2984                                ;ERROR 120
2985  004664  060031                   EM12
2986  004666  065050                   DH21                    ;AFTER SUBSYS CLEAR
2987  004670  066276                   DT1
2988  004672  067270                   DF10
2989                                ;ERROR 121
2990  004674  060031                   EM12
2991  004676  064405                   DH9                     ;AFTER START SPINDLE CMD
2992  004700  066276                   DT1
2993  004702  067270                   DF10
2994                                ;ERROR 122
2995  004704  060031                   EM12
2996  004706  064647                   DH14                    ;AFTER SEEK WITH BAD PARITY
2997  004710  066276                   DT1
2998  004712  067270                   DF10
2999                                ;ERROR 123
3000  004714  063750                   EM88                    ;NO DRIVES FOUND
3001  004716  064170                   DH1
3002  004720  066276                   DT1
3003  004722  067144                   DF1
3004                                ;ERROR 124
3005  004724  060031                   EM12
3006  004726  064741                   DH17                    ;AFTER RECAL CMD
3007  004730  066276                   DT1
3008  004732  067270                   DF10
3009                                ;ERROR 125
```

```
3010  004734  060062                    EM13                    ;NO ATTN
3011  004736  064647                    DH14                    ;FROM SEEK WITH BAD PARITY
3012  004740  066276                    DT1
3013  004742  067270                    DF10
3014                          ;ERROR 126
3015  004744  064041                    EM89                    ;NO DRVS FOUND IN DEVICE MAP
3016  004746  064170                    DH1
3017  004750  066276                    DT1
3018  004752  067144                    DF1
3019                          ;ERROR 127
3020  004754  061312                    EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3021  004756  065050                    DH21                    ;AFTER SCLR
3022  004760  066276                    DT1
3023  004762  067270                    DF10
3024                          ;ERROR 130
3025  004764  061347                    EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
3026  004766  065050                    DH21
3027  004770  066276                    DT1
3028  004772  067270                    DF10
3029                          ;ERROR 131
3030  004774  060031                    EM12                    ;NO RDY
3031  004776  065145                    DH25                    ;AFTER SEEK CMD
3032  005000  066276                    DT1
3033  005002  067270                    DF10
3034                          ;ERROR 132
3035  005004  060062                    EM13                    ;NO ATTN
3036  005006  065145                    DH25
3037  005010  066276                    DT1
3038  005012  067270                    DF10
3039                          ;ERROR 133
3040  005014  060210                    EM17                    ;MSG A0 ERROR
3041  005016  065145                    DH25
3042  005020  066700                    DT13
3043  005022  067440                    DF20
3044                          ;ERROR 134
3045  005024  060225                    EM18                    ;MSG B0 ERROR
3046  005026  065145                    DH25
3047  005030  066700                    DT13
3048  005032  067440                    DF20
3049                          ;ERROR 135
3050  005034  060242                    EM19                    ;MSG A1 ERROR
3051  005036  065145                    DH25
3052  005040  066700                    DT13
3053  005042  067440                    DF20
3054                          ;ERROR 136
3055  005044  060257                    EM20                    ;MSG B1 ERROR.
3056  005046  065145                    DH25
3057  005050  066700                    DT13
3058  005052  067440                    DF20
3059                          ;ERROR 137
3060  005054  061312                    EM39                    ;CYL DIFF/OFFSET IN A2 NOT CLEARED
3061  005056  065145                    DH25
3062  005060  066760                    DT14
3063  005062  067514                    DF22
3064                          ;ERROR 140
3065  005064  061347                    EM40                    ;CYL ADDR IN B2 NOT CLEARED
```

```
3066  005066  065145                    DH25
3067  005070  066760                    DT14
3068  005072  067514                    DF22
3069                          ;ERROR 141
3070  005074  062172                    EM52          ;20 SECTOR FORMAT NOT SET IN RKMR2
3071  005076  064170                    DH1
3072  005100  066276                    DT1
3073  005102  067144                    DF1
3074                          ;ERROR 142
3075  005104  062231                    EM53          ;SECTOR 0 NOT FOUND WITHIN 50 MS
3076  005106  064170                    DH1
3077  005110  066276                    DT1
3078  005112  067144                    DF1
3079                          ;ERROR 143
3080  005114  062262                    EM54          ;DIFF SECTOR NOT FOUND WITHIN 3MS
3081  005116  064170                    DH1
3082  005120  066276                    DT1
3083  005122  067144                    DF1
3084                          ;ERROR 144
3085  005124  061720                    EM48          ;MSG B3 ERROR
3086  005126  065323                    DH34          ;SECTOR REG UNSTABLE
3087  005130  066276                    DT1
3088  005132  067270                    DF10
3089                          ;ERROR 145
3090  005134  061720                    EM48
3091  005136  065347                    DH35          ;BETWEEN SECTOR COUNTS
3092  005140  066404                    DT6
3093  005142  067310                    DF12
3094                          ;ERROR 146
3095  005144  060242                    EM19          ;MSG A1 ERROR
3096  005146  064647                    DH14          ;AFTER SEEK WITH BAD PARITY
3097  005150  066760                    DT14
3098  005152  067514                    DF22
3099                          ;ERROR 147
3100  005154  060257                    EM20          ;MSG B1 ERROR
3101  005156  064647                    DH14
3102  005160  066760                    DT14
3103  005162  067514                    DF22
3104                          ;ERROR 150
3105  005164  060242                    EM19          ;MSG A1 ERROR
3106  005166  065420                    DH37
3107  005170  066276                    DT1
3108  005172  067270                    DF10
3109                          ;ERROR 151
3110  005174  060031                    EM12          ;NO RDY
3111  005176  065076                    DH22          ;AFTER CLEAR CMD
3112  005200  066276                    DT1
3113  005202  067270                    DF10
3114                          ;ERROR 152
3115  005204  064144                    EM90          ;DDT NOT SET IN RKDS
3116  005206  064170                    DH1
3117  005210  066276                    DT1
3118  005212  067144                    DF1
3119                          ;ERROR 153
3120  005214  000000                    0
3121  005216  000000                    0
```

```
3122  005220  000000                    0
3123  005222  000000                    0
3124                          ;ERROR 154
3125  005224  062315                    EM55          ;ATTN NOT CLEARED
3126  005226  065076                    DH22
3127  005230  066276                    DT1
3128  005232  067270                    DF10
3129                          ;ERROR 155
3130  005234  063605                    EM85          ;IDAE NOT CLEARED
3131  005236  064741                    DH17          ;AFTER RECAL CMD
3132  005240  066276                    DT1
3133  005242  067270                    DF10
3134                          ;ERROR 156
3135  005244  060031                    EM12          ;CONT NOT READY
3136  005246  066150                    DH51          ;AFTER SEEK TO SELF
3137  005250  066276                    DT1
3138  005252  067270                    DF10
3139                          ;ERROR 157
3140  005254  060062                    EM13          ;NO ATTN
3141  005256  066150                    DH51
3142  005260  066276                    DT1
3143  005262  067270                    DF10
3144                          ;ERROR 160
3145  005264  062475                    EM59          ;LIMIT DETECT NOT FOUND
3146  005266  064170                    DH1
3147  005270  066276                    DT1
3148  005272  067144                    DF1
3149                          ;ERROR 161
3150  005274  060210                    EM17          ;MSG A0 ERROR
3151  005276  065464                    DH38          ;AFTER LIMIT DETECT
3152  005300  066700                    DT13
3153  005302  067440                    DF20
3154                          ;ERROR 162
3155  005304  060225                    EM18          ;MSG B0 ERROR
3156  005306  065464                    DH38
3157  005310  066700                    DT13
3158  005312  067440                    DF20
3159                          ;ERROR 163
3160  005314  060242                    EM19          ;MSG A1 ERROR
3161  005316  065464                    DH38
3162  005320  066700                    DT13
3163  005322  067440                    DF20
3164                          ;ERROR 164
3165  005324  060257                    EM20          ;MSG B1 ERROR
3166  005326  065464                    DH38
3167  005330  066700                    DT13
3168  005332  067440                    DF20
3169                          ;ERROR 165
3170  005334  060062                    EM13          ;NO ATTN
3171  005336  065464                    DH38
3172  005340  066276                    DT1
3173  005342  067270                    DF10
3174                          ;ERROR 166
3175  005344  062526                    EM60          ;HEADS HOME NOT FOUND
3176  005346  065464                    DH38
3177  005350  066276                    DT1
```

```
3178  005352  067270                    DF10
3179                          ;ERROR 167
3180  005354  062562                    EM61          ;LOAD HEADS NOT FOUND
3181  005356  065464                    DH38
3182  005360  066276                    DT1
3183  005362  067270                    DF10
3184                          ;ERROR 170
3185  005364  057354                    EM4           ;FATAL ERROR
3186  005366  065716                    DH45          ;LIMIT DETECT ERROR ON PREVIOUS TEST
3187  005370  066276                    DT1
3188  005372  067374                    DF16
3189                          ;ERROR 171
3190  005374  060031                    EM12          ;NO RDY
3191  005376  065230                    DH30          ;AFTER READ HEADER CMD
3192  005400  066276                    DT1
3193  005402  067270                    DF10
3194                          ;ERROR 172
3195  005404  061312                    EM39          ;CYL DIFF/OFFSET NOT CLEARED
3196  005406  065230                    DH30          ;AFTER READ HEADER CMD
3197  005410  066760                    DT14
3198  005412  067514                    DF22
3199                          ;ERROR 173
3200  005414  062616                    EM63          ;DLT SET
3201  005416  065230                    DH30
3202  005420  066276                    DT1
3203  005422  067354                    DF15
3204                          ;ERROR 174
3205  005424  060274                    EM21          ;CERR SET
3206  005426  065230                    DH30
3207  005430  066276                    DT1
3208  005432  067354                    DF15
3209                          ;ERROR 175
3210  005434  061312                    EM39          ;CYL DIFF NOT CLEARED
3211  005436  064451                    DH10          ;AT END OF HEAD LOADING
3212  005440  066760                    DT14
3213  005442  067514                    DF22
3214                          ;ERROR 176
3215  005444  061347                    EM40          ;CYL ADDR NOT CLEARED.
3216  005446  064451                    DH10
3217  005450  066760                    DT14
3218  005452  067514                    DF22
3219                          ;ERROR 177
3220  005454  063114                    EM72          ;FORMAT TEST BYPASSED
3221  005456  065775                    DH46          ;COULD NOT READ BSE INFO
3222  005460  066276                    DT1
3223  005462  067374                    DF16
3224                          ;ERROR 200
3225  005464  060031                    EM12          ;NO RDY
3226  005466  065502                    DH39          ;AFTER WRITE HEADER CMD
3227  005470  066276                    DT1
3228  005472  067354                    DF15
3229                          ;ERROR 201
3230  005474  060274                    EM21          ;CERR SET
3231  005476  065502                    DH39
3232  005500  066276                    DT1
3233  005502  067354                    DF15
```

```
3234                                    ;ERROR 202
3235  005504  062666                        EM65                ;READ HEADER ERROR
3236  005506  064170                        DH1
3237  005510  066450                        DT7
3238  005512  067334                        DF14
3239                                    ;ERROR 203
3240  005514  060210                        EM17                ;MSG A0 ERROR
3241  005516  065303                        DH33                ;DURING SEEK CMD
3242  005520  066700                        DT13
3243  005522  067440                        DF20
3244                                    ;ERROR 204
3245  005524  060225                        EM18                ;MSG B0 ERROR
3246  005526  065303                        DH33
3247  005530  066700                        DT13
3248  005532  067440                        DF20
3249                                    ;ERROR 205
3250  005534  060242                        EM19                ;MSG A1 ERROR
3251  005536  065303                        DH33
3252  005540  066700                        DT13
3253  005542  067440                        DF20
3254                                    ;ERROR 206
3255  005544  060257                        EM20                ;MSG B1 ERROR
3256  005546  065303                        DH33
3257  005550  066700                        DT13
3258  005552  067440                        DF20
3259                                    ;ERROR 207
3260  005554  061141                        EM36                ;CYL ADDR IN RKMR3 INCORRECT
3261  005556  065145                        DH25                ;AFTER SEEK CMD
3262  005560  066336                        DT4
3263  005562  067244                        DF6
3264                                    ;ERROR 210
3265  005564  060274                        EM21                ;CERR SET
3266  005566  065145                        DH25
3267  005570  066276                        DT1
3268  005572  067270                        DF10
3269                                    ;ERROR 211
3270  005574  062742                        EM67                ;READ CYL 0 HEADERS ON CYL 1
3271  005576  065145                        DH25
3272  005600  066276                        DT1
3273  005602  067270                        DF10
3274                                    ;ERROR 212
3275  005604  061241                        EM38                ;CYL DIFF IN RKMR2 NOT = CALDIF
3276  005606  065303                        DH33                ;DURING SEEK CMD
3277  005610  066336                        DT4
3278  005612  067244                        DF6
3279                                    ;ERROR 213
3280  005614  060210                        EM17                ;MSG A0 ERROR
3281  005616  065565                        DH41                ;DURING RECAL CMD
3282  005620  066700                        DT13
3283  005622  067440                        DF20
3284                                    ;ERROR 214
3285  005624  060225                        EM18                ;MSG B0 ERROR
3286  005626  065565                        DH41
3287  005630  066700                        DT13
3288  005632  067440                        DF20
3289                                    ;ERROR 215
```

```
3290  005634  060242                    EM19                    ;MSG A1 ERROR
3291  005636  065565                    DH41
3292  005640  066700                    DT13
3293  005642  067440                    DF20
3294                          ;ERROR 216
3295  005644  060257                    EM20                    ;MSG B1 ERROR
3296  005646  065565                    DH41
3297  005650  066700                    DT13
3298  005652  067440                    DF20
3299                          ;ERROR 217
3300  005654  061241                    EM38                    ;CYL DIFF IN RKMR2 NOT=CALDIF
3301  005656  065565                    DH41
3302  005660  066336                    DT4
3303  005662  067244                    DF6
3304                          ;ERROR 220
3305  005664  060274                    EM21                    ;CERR SET
3306  005666  064741                    DH17                    ;AFTER RECAL CMD
3307  005670  066276                    DT1
3308  005672  067270                    DF10
3309                          ;ERROR 221
3310  005674  060210                    EM17                    ;MSG A0 ERROR
3311  005676  064741                    DH17
3312  005700  066700                    DT13
3313  005702  067440                    DF20
3314                          ;ERROR 222
3315  005704  060242                    EM19                    ;MSG A1 ERROR
3316  005706  064741                    DH17
3317  005710  066700                    DT13
3318  005712  067440                    DF20
3319                          ;ERROR 223
3320  005714  061312                    EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3321  005716  064741                    DH17
3322  005720  066276                    DT1
3323  005722  067270                    DF10
3324                          ;ERROR 224
3325  005724  062706                    EM66                    ;CYL ADDR IN RKMR3 INCORRECT
```

```
 3326  005726  064741                    DH17
 3327  005730  066276                    DT1
 3328  005732  067270                    DF10
 3329                          ;ERROR 225
 3330  005734  063001                    EM68          ;READING CYL 1 HEADERS ON CYL 0
 3331  005736  064741                    DH17
 3332  005740  066276                    DT1
 3333  005742  067270                    DF10
 3334                          ;ERROR 226
 3335  005744  060031                    EM12          ;NO RDY
 3336  005746  065164                    DH26          ;AFTER READ DATA CMD
 3337  005750  066276                    DT1
 3338  005752  067270                    DF10
 3339                          ;ERROR 227
 3340  005754  060274                    EM21          ;CERR SET
 3341  005756  065164                    DH26
 3342  005760  066276                    DT1
 3343  005762  067354                    DF15
 3344                          ;ERROR 230
 3345  005764  063723                    EM87          ;CANT READ BSE INFO
```

```
3346  005766  066227                    DH53                ;ON SECT 10,12,14,16,18,20
3347  005770  066276                    DT1
3348  005772  067414                    DF17
3349                            ;ERROR 231
3350  005774  000000                    0
3351  005776  000000                    0
3352  006000  000000                    0
3353  006002  000000                    0
3354                            ;ERROR 232
3355  006004  061141                    EM36                ;CYL ADDR IN RKMR3 NOT=RKDC
3356  006006  065145                    DH25                ;AFTER SEEK CMD
3357  006010  066276                    DT1
3358  006012  067224                    DF5
3359                            ;ERROR 233
3360  006014  063723                    EM87                ;CANNOT READ BSE INFO
3361  006016  065606                    DH42                ;ON SECT 0,2,4,6,8
3362  006020  066276                    DT1
3363  006022  067414                    DF17
3364                            ;ERROR 234
3365  006024  000000                    0
3366  006026  000000                    0
3367  006030  000000                    0
3368  006032  000000                    0
3369                            ;ERROR 235
3370  006034  063040                    EM69                ;ALIGN CARTRIDGE USED
3371  006036  065646                    DH44                ;WILL BYPASS FORMAT & ALL R/W TESTS
3372  006040  066276                    DT1
3373  006042  067270                    DF10
3374                            ;ERROR 236
3375  006044  062350                    EM56                ;UNEXP MEM PARITY TRAP
3376  006046  065124                    DH23                ;TEST #, TRAP PC
3377  006050  066674                    DT11
3378  006052  067204                    DF3
3379                            ;ERROR 237
3380  006054  063073                    EM71                ;DSC SET
3381  006056  065076                    DH22                ;AFTER DRIVE CLEAR CMD
3382  006060  066276                    DT1
3383  006062  067270                    DF10
3384                            ;ERROR 240
3385  006064  063001                    EM68                ;READ CYL 1 HEADERS ON CYL 0
3386  006066  064741                    DH17                ;AFTER RECAL CMD
3387  006070  066276                    DT1
3388  006072  067270                    DF10
3389                            ;ERROR 241
3390  006074  061141                    EM36                ;RKMR3 NOT = RKDC
3391  006076  064647                    DH14                ;AFTER SEEK WITH BAD PARITY
3392  006100  066516                    DT8
3393  006102  067244                    DF6
3394                            ;ERROR 242
3395  006104  061241                    EM38                ;CYL DIFF IN RKMR2 INCORRECT
3396  006106  064647                    DH14
3397  006110  066516                    DT8
3398  006112  067244                    DF6
3399
3400                            ;ERROR 243
3401  006114  061141                    EM36                ;CYL ADDR IN RKMR3 INCORRECT
```

```
3402  006116  065145                DH25            ;AFTER SEEK CMD
3403  006120  066516                DT8
3404  006122  067244                DF6
3405                          ;ERROR 244
3406  006124  063162                EM74            ;RTZ NOT SET
3407  006126  065565                DH41            ;DURING RECAL CMD
3408  006130  066276                DT1
3409  006132  067270                DF10
3410                          ;ERROR 245
3411  006134  060062                EM13            ;NO ATTN
3412  006136  066042                DH48            ;AFTER SEEK TO INVALID CYL
3413  006140  066276                DT1
3414  006142  067270                DF10
3415                          ;ERROR 246
3416  006144  063207                EM75            ;IDAE NOT SET
3417  006146  066042                DH48
3418  006150  066336                DT4
3419  006152  067244                DF6
3420                          ;ERROR 247
3421  006154  060753                EM32            ;FAULT NOT SET
3422  006156  066042                DH48
3423  006160  066336                DT4
3424  006162  067244                DF6
3425                          ;ERROR 250
3426  006164  063235                EM76            ;PIP SET
3427  006166  066042                DH48
3428  006170  066336                DT4
3429  006172  067244                DF6
3430                          ;ERROR 251
3431  006174  060163                EM16            ;DSC NOT SET
3432  006176  066042                DH48
3433  006200  066336                DT4
3434  006202  067244                DF6
3435                          ;ERROR 252
3436  006204  060210                EM17            ;MSG A0 ERROR
3437  006206  066042                DH48
3438  006210  066700                DT13
3439  006212  067440                DF20
3440                          ;ERROR 253
3441  006214  060225                EM18            ;MSG B0 ERROR
3442  006216  066042                DH48
3443  006220  066700                DT13
3444  006222  067440                DF20
3445                          ;ERROR 254
3446  006224  060242                EM19            ;MSG A1 ERROR
3447  006226  066042                DH48
3448  006230  066700                DT13
3449  006232  067440                DF20
3450                          ;ERROR 255
3451  006234  060257                EM20            ;MSG B1 ERROR
3452  006236  066042                DH48
3453  006240  066700                DT13
3454  006242  067440                DF20
3455                          ;ERROR 256
3456  006244  061241                EM38            ;CYL DIFF IN RKMR2 NOT='CYL DIF'
3457  006246  066042                DH48
```

```
3458  006250  066336                    DT4
3459  006252  067244                    DF6
3460                            ;ERROR 257
3461  006254  061141                    EM36          ;CYL ADDR IN RKMR3 NOT=RKDC
3462  006256  066042                    DH48
3463  006260  066336                    DT4
3464  006262  067244                    DF6
3465                            ;ERROR 260
3466  006264  000000                    0
3467  006266  000000                    0
3468  006270  000000                    0
3469  006272  000000                    0
3470                            ;ERROR 261
3471  006274  000000                    0
3472  006276  000000                    0
3473  006300  000000                    0
3474  006302  000000                    0
3475                            ;ERROR 262
3476  006304  063256                    EM77          ;FAULT NOT CLEARED
3477  006306  065076                    DH22          ;AFTER DRIVE CLEAR CMD
3478  006310  066276                    DT1
3479  006312  067270                    DF10
3480                            ;ERROR 263
3481  006314  063304                    EM78          ;CYL DIFF IN RKMR2 NOT=1 IN SEEK TO SELF
3482  006316  064647                    DH14          ;AFTER SEEK WITH BAD PARITY
3483  006320  066516                    DT8
3484  006322  067244                    DF6
3485                            ;ERROR 264
3486  006324  061375                    EM41          ;CYL ADDR NOT CLEARED
3487  006326  065230                    DH30          ;AFTER READ HEADER CMD
3488  006330  066760                    DT14
3489  006332  067514                    DF22
3490                            ;ERROR 265
3491  006334  060225                    EM18          ;MSG B0 ERROR
3492  006336  065076                    DH22          ;AFTER DRIVE CLEAR CMD
3493  006340  066700                    DT13
3494  006342  067440                    DF20
3495                            ;ERROR 266
3496  006344  060257                    EM20          ;MSG B1 ERROR
3497  006346  065076                    DH22
3498  006350  066700                    DT13
3499  006352  067440                    DF20
3500                            ;ERROR 267
3501  006354  060225                    EM18
3502  006356  065502                    DH39          ;AFTER WRITE HEADER CMD
3503  006360  066700                    DT13
3504  006362  067440                    DF20
3505                            ;ERROR 270
3506  006364  060257                    EM20
3507  006366  065502                    DH39
3508  006370  066700                    DT13
3509  006372  067440                    DF20
3510                            ;ERROR 271
3511  006374  060225                    EM18
3512  006376  065230                    DH30          ;AFTER READ HEADER CMD
3513  006400  066700                    DT13
```

```
3514  006402  067440                    DF20
3515                          ;ERROR 272
3516  006404  060257                    EM20
3517  006406  065230                    DH30
3518  006410  066700                    DT13
3519  006412  067440                    DF20
3520                          ;ERROR 273
3521  006414  060210                    EM17          ;MSG A0 ERROR
3522  006416  065076                    DH22          ;AFTER DRV CLR CMD
3523  006420  066700                    DT13
3524  006422  067440                    DF20
3525                          ;ERROR 274
3526  006424  060242                    EM19          ;MSG A1 ERROR
3527  006426  065076                    DH22
3528  006430  066700                    DT13
3529  006432  067440                    DF20
3530                          ;ERROR 275
3531  006434  060225                    EM18          ;MSG B0 ERROR
3532  006436  064741                    DH17          ;AFTER RECAL CMD
3533  006440  066700                    DT13
3534  006442  067440                    DF20
3535                          ;ERROR 276
3536  006444  060257                    EM20          ;MSG B1 ERROR
3537  006446  064741                    DH17
3538  006450  066700                    DT13
3539  006452  067440                    DF20
3540                          ;ERROR 277
3541  006454  060210                    EM17          ;MSG A0 ERROR
3542  006456  065502                    DH39          ;AFTER WRITE HEADER CMD
3543  006460  066700                    DT13
3544  006462  067440                    DF20
3545                          ;ERROR 300
3546  006464  060242                    EM19          ;MSG A1 ERROR
3547  006466  065502                    DH39
3548  006470  066700                    DT13
3549  006472  067440                    DF20
3550                          ;ERROR 301
3551  006474  060210                    EM17
3552  006476  065230                    DH30          ;AFTER READ HEADER CMD
3553  006500  066700                    DT13
3554  006502  067440                    DF20
3555                          ;ERROR 302
3556  006504  060242                    EM19
3557  006506  065230                    DH30
3558  006510  066700                    DT13
3559  006512  067440                    DF20
3560                          ;ERROR 303
3561  006514  061312                    EM39          ;CYL DIFF/OFFSET NOT CLEARED
3562  006516  065502                    DH39          ;AFTER WRITE HEADER CMD
3563  006520  066760                    DT14
3564  006522  067514                    DF22
3565                          ;ERROR 304
3566  006524  061375                    EM41          ;CYL ADDR NOT CLEARED
3567  006526  0655C2                    DH39
3568  006530  066760                    DT14
3569  006532  067514                    DF22
```

```
3570
3571                                    ;ERROR 305
3572  006534  063412                            EM80            ;UNLD NOT SET
3573  006536  065464                            DH38            ;AFTER LIMIT DETECT
3574  006540  066276                            DT1
3575  006542  067270                            DF10
3576                                    ;ERROR 306
3577  006544  063442                            EM81            ;SPIN NOT SET
3578  006546  064405                            DH9             ;AFTER START SPIN CMD.
3579  006550  066276                            DT1
3580  006552  067270                            DF10
3581                                    ;ERROR 307
3582  006554  063470                            EM82            ;RTZ NOT SET
3583  006556  065565                            DH41            ;DURING RECAL CMD
3584  006560  066276                            DT1
3585  006562  067270                            DF10
3586                                    ;ERROR 310
3587  006564  063515                            EM83            ;READ HEADER ERROR
3588  006566  064170                            DH1
3589  006570  066564                            DT9
3590  006572  067474                            DF21
3591                                    ;ERROR 311
3592  006574  063515                            EM83
3593  006576  064170                            DH1
3594  006600  066630                            DT10
3595  006602  067474                            DF21
3596                                    ;ERROR 312
3597  006604  063555                            EM84            ;FORMAT NOT SET
3598  006606  065502                            DH39            ;AFTER WRITE HEADER CMD
3599  006610  066276                            DT1
3600  006612  067270                            DF10
3601                                    ;ERROR 313
3602  006614  063555                            EM84
3603  006616  065230                            DH30            ;AFTER READ HEADER CMD
3604  006620  066276                            DT1
3605  006622  067270                            DF10
3606                                    ;ERROR 314
3607  006624  062373                            EM57            ;WCE AT CYL 411,TRK 2,SEC 21
3608  006626  064170                            DH1
3609  006630  066276                            DT1
3610  006632  067210                            DF4
3611                                    ;ERROR 315
3612  006634  062427                            EM58            ;SPOK NOT CLEARED
3613  006636  064761                            DH18            ;AFTER UNLD CMD
3614  006640  066276                            DT1
3615  006642  067270                            DF10
3616                                    ;ERROR 316
3617  006644  063060                            EM70            ;UNEXP ATTN
3618  006646  064405                            DH9             ;AFTER START SPIN CMD
3619  006650  066276                            DT1
3620  006652  067270                            DF10
3621                                    ;ERROR 317
3622  006654  063060                            EM70
3623  006656  064500                            DH11            ;AFT SPIN CMD &  FWD SET
3624  006660  066276                            DT1
3625  006662  067270                            DF10
```

```
3626
3627  006664  063060                ;ERROR 320
                                          EM70
3628  006666  064537                       DH12          ;AT INNER LIMIT FROM ST SPIN CMD
3629  006670  066276                       DT1
3630  006672  067270                       DF10
3631                                ;ERROR 321
3632  006674  063060                       EM70
3633  006676  064600                       DH13          ;FROM OUTER LIM TO CYL 0 DURING LOADING
3634  006700  066276                       DT1
3635  006702  067270                       DF10
3636                                ;ERROR 322
3637  006704  060242                       EM19          ;MSG A1 ERROR
3638  006706  064677                       DH16          ;AFT LD HEAD REG & SEEK
3639  006710  066700                       DT13
3640  006712  067440                       DF20
3641                                ;ERROR 323
3642  006714  060257                       EM20          ;MSG B1 ERROR.
3643  006716  064677                       DH16
3644  006720  066700                       DT13
3645  006722  067440                       DF20
3646                                ;ERROR 324
3647  006724  061672                       EM46          ;MSG A2 ERROR
3648  006726  064677                       DH16
3649  006730  066760                       DT14
3650  006732  067514                       DF22
3651                                ;ERROR 325
3652  006734  061705                       EM47          ;MSG B2 ERROR
3653  006736  064677                       DH16
3654  006740  066760                       DT14
3655  006742  067514                       DF22
3656                                ;ERROR 326
3657  006744  063141                       EM73          ;CTO SET
3658  006746  063636                       EM86          ;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
3659  006750  066276                       DT1
3660  006752  067160                       DF2
3661                                ;ERROR 327
3662  006754  063371                       EM79          ;NED SET
3663  006756  063636                       EM86
3664  006760  066276                       DT1
3665  006762  067160                       DF2
3666
3667                                ;ERROR 330
3668  006764  057467                       EM5           ;MDS SET
3669  006766  063636                       EM86
3670  006770  066276                       DT1
3671  006772  067160                       DF2
```

```
3672
3673                                    .SBTTL  PROGRAM SETUP
3674
3675  006774  012737  000001  001336  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
3676  007002  005037  001340                  CLR     BYPT16
3677  007006  005037  001342                  CLR     MODTST
3678  007012  000450                    BR      PRGSRT          ;START PROGRAM
3679
3680  007014  005037  001336          BYT16:  CLR     PARAM
3681  007020  012737  000001  001340          MOV     #1,BYPT16       ;SET FLAG TO BYPASS TEST 16
3682  007026  005037  001342                  CLR     MODTST
3683  007032  000440                    BR      PRGSRT
3684
3685  007034  012737  000001  001336  BYT16A: MOV     #1,PARAM
3686  007042  012737  000001  001340          MOV     #1,BYPT16
3687  007050  005037  001342                  CLR     MODTST
3688  007054  000427                    BR      PRGSRT
3689
3690  007056  005037  001336          MDTST:  CLR     PARAM
3691  007062  005037  001340                  CLR     BYPT16
3692  007066  012737  000001  001342          MOV     #1,MODTST
3693  007074  000417                    BR      PRGSRT
3694
3695  007076  005037  001336          MDTSTA: CLR     PARAM
3696  007102  012737  000001  001340          MOV     #1,BYPT16
3697  007110  012737  000001  001342          MOV     #1,MODTST
3698  007116  000406                    BR      PRGSRT
3699  007120  005037  001336          START:  CLR     PARAM           ;CLEAR FOR 200 START
3700  007124  005037  001340                  CLR     BYPT16
3701  007130  005037  001342                  CLR     MODTST
3702  007134  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
3703  007136  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3704  007142  012746  000000                  MOV     #PR0,-(SP)      ;PSW LOADED TO BE
3705  007146  012746  007154                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
3706  007152  000002                    RTI                     ;ENABLE ALL INTERRUPTS
3707
3708  007154  004737  052266          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
3709                                                                  ;& TURN ON KB INTERRUPT
3710
3711
3712                                          ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3713                                          ;*** ANY DEVICE WHICH SETS ITS   ***
3714                                          ;*** INTERRUPT ENABLE BIT WILL   ***
3715                                          ;*** SERVICED.                   ***
3716
3717                                          ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3718                                          ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3719                                          ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3720
3721                                          ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3722
3723                                                                  ;SYSMAC 'SETUP'
3724                                    .SBTTL  INITIALIZE THE COMMON TAGS
3725                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3726  007160  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3727  007164  005026                    CLR     (R6)+           ;;CLEAR MEMORY LOCATION
```

```
3728  007166  022706  001140            CMP     #SWR,R6 ;;DONE?
3729  007172  001374                    BNE     .-6             ;;LOOP BACK IF NO
3730  007174  012706  001100            MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3731                          ;;INITIALIZE A FEW VECTORS
3732  007200  012737  050322  000020    MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3733  007206  012737  000340  000022    MOV     #340,@#IOTVEC+2 ;;LEVEL 7
3734  007214  012737  050602  000030    MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3735  007222  012737  000340  000032    MOV     #340,@#EMTVEC+2 ;;LEVEL 7
3736  007230  012737  054404  000034    MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3737  007236  012737  000340  000036    MOV     #340,@#TRAPVEC+2;LEVEL 7
3738  007244  012737  050234  000024    MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
3739  007252  012737  000340  000026    MOV     #340,@#PWRVEC+2 ;;LEVEL 7
3740  007260  013737  043014  043006    MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
3741  007266  005037  001174            CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
3742  007272  005037  001176            CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3743  007276  112737  000001  001115    MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
3744  007304  012737  007304  001106    MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3745  007312  012737  007312  001110    MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
3746                          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3747                          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3748  007320  013746  000004            MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3749  007324  012737  007360  000004    MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3750  007332  012737  177570  001140    MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
3751  007340  012737  177570  001142    MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3752  007346  022777  177777  171564    CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3753  007354  001012                    BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3754                                                            ;;AND  THE HARDWARE SWR IS NOT = -1
3755  007356  000403                    BR      65$             ;;BRANCH IF NO TIMEOUT
3756  007360  012716  007366    64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
3757  007364  000002                    RTI
3758  007366  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
3759  007374  012737  000174  001142          MOV     #DISPREG,DISPLAY
3760  007402  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
3761
3762  007406  005037  001216            CLR     $PASS           ;;CLEAR PASS COUNT
3763  007412  132737  000200  001231    BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
3764  007420  001403                    BEQ     67$             ;;YES,USE NON-APT SWITCH
3765  007422  012737  001232  001140    MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
3766  007430                    67$:
3767  007430  012737  000000  000032    MOV     #PR0,EMTVEC+2   ;SET EMT PRIORITY TO 0
3768  007436  012737  000000  000036    MOV     #PR0,TRAPVEC+2  ;SET TRAP PRIORITY TO 0
3769  007444  012737  007510  000004  MEMPAR: MOV     #1$,ERRVEC      ;SET TIMEOUT VECTOR
3770  007452  012737  000340  000006          MOV     #PR7,ERRVEC+2
3771
3772  007460  012701  172100            MOV     #MEMBAS,R1      ;ADDR OF MEM CSR
3773  007464  005011                3$:     CLR     (R1)            ;SEE IF CAN REFERENCE
3774  007466  012711  000001            MOV     #1,(R1)         ;SET ENABLE BIT IF YES
3775  007472  012737  050152  000114    MOV     #MEMERR,MEMVEC  ;LOAD MEMORY CHECK VECTOR IF NO TIMEOUT
3776  007500  012737  000340  000116    MOV     #PR7,MEMVEC+2
3777  007506  000401                    BR      2$
3778
3779  007510  022626                1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
3780  007512  062701  000002          2$:     ADD     #2,R1           ;TRY NEXT CSR
3781  007516  020127  172140            CMP     R1,#MEMBAS+40   ;SEE IF TRIED ALL
3782  007522  001360                    BNE     3$              ;BR IF NO
3783  007524  012737  000006  000004    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 DR0 PACK W/SCRATCH & DO<CR>
                        HALT                    ;HALT HERE
                        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
```

```
3840  007726  012737  000210  001314          MOV     #210,RKVEC      ;DEFAULT VALUE
3841  007734  004737  043454                  JSR     PC,SETINT
3842  007740  005237  003472                  INC     SIZFLG          ;DO "SIZE THE BUSS" TEST
3843
3844  007744  005037  003304          ST5:    CLR     UNLD            ;INITIALIZE FLAGS
3845  007750  005037  003306                  CLR     BADHDR                  ;USED IN 'STOP ROUTINE
3846  007754  005037  003310                  CLR     HPEND                   ;FOR VALID PROGRAM HALTS
3847  007760  005037  001176                  CLR     $ESCAPE
3848  007764  005037  001170                  CLR     $TMP4           ;CLR RK06 IND. FLAG
3849  007770  012737  003444  001346          MOV     #DRIV0,DRVPTR   ;SETUP
3850  007776  005037  001220                  CLR     $DEVCT          ;NO. OF DRVS DONE
3851  010002  005037  001222                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
3852  010006  012737  010054  000004          MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
3853  010014  005777  171306                  TST     @LKS            ;SEE IF L-CLOCK THERE
3854  010020  005237  003464                  INC     LCLKF           ;PRESENT, SET FLAG.
3855  010024  013700  001330                  MOV     LCVEC,R0        ;VECTOR ADDR
3856  010030  012737  010116  000004          MOV     #2$,ERRVEC
3857  010036  005777  171256                  TST     @PKS            ;SEE IF P-CLOCK THERE
3858  010042  005237  003466                  INC     PCLKF           ;PRESENT, SET FLAG
3859  010046  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3860  010052  000412                          BR      3$
3861
3862  010054  022626                  1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
3863  010056  012737  010122  000004          MOV     #4$,ERRVEC
3864  010064  005777  171230                  TST     @PKS            ;SEE IF P-CLOCK THERE
3865  010070  005237  003466                  INC     PCLKF           ;PRESENT, SET FLAG
3866  010074  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3867  010100  005237  003470          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
3868  010104  012720  047304                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
3869  010110  012710  000300                  MOV     #PR6,(R0)
3870  010114  000407                          BR      TST1            ;;GO TO NEXT TEST
3871
3872  010116  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
3873  010120  000767                          BR      3$
3874
3875  010122  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
3876  010124  005037  003470                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
3877  010130  104401  056253                  TYPE    ,MSG13          ;ALL TIMING TESTS BYPASSED
3878
3879
```

```
                                        .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'
                                        ;*
                                        ;;**********************************************************************
010134  000004                        TST1:   SCOPE
010136  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
010144  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR
                                
010150  012746  000000                        MOV     #PR0,-(SP)      ;RESET PSW TO PRIORITY 0
010154  012746  010162                        MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
010160  000002                              RTI
010162                                  5$:
                                
010162  012737  010306  000004              MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
010170  013705  001264                        MOV     $BASE,R5        ;SETUP INDEX REG.
010174  005765  000000                        TST     RKCS1(R5)       ;REFERENCE ALL THE
010200  005765  000010                        TST     RKCS2(R5)       ;CONTROLLER REGISTERS
010204  005765  000002                        TST     RKWC(R5)
010210  005765  000004                        TST     RKBA(R5)
010214  005765  000006                        TST     RKDA(R5)
010220  005765  000012                        TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
010224  005765  000014                        TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
010230  005765  000016                        TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
010234  005765  000020                        TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
010240  005765  000024                        TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
010244  005765  000026                        TST     RKMR1(R5)
010250  005765  000034                        TST     RKMR2(R5)
010254  005765  000036                        TST     RKMR3(R5)
010260  005765  000030                        TST     RKECPS(R5)
010264  005765  000032                        TST     RKECPT(R5)
                                
010270  012737  050064  000004              MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
010276  012737  000340  000006              MOV     #PR7,ERRVEC+2
010304  000404                              BR      TST2            ;;GO TO NEXT TEST
                                
010306  022626                        1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
010310  104007                              ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
010312  000137  042760                        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 AS AN RK06.  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.
                                        ;*      IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.
                                        ;*
                                        ;;*****************************************************************
010316  000004                TST2:   SCOPE
010320  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
010326  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
010332  005237  001462                  INC     BYPCERR         ;DO NOT TEST CERR IN 'FRDY'

010336  132737  000200  001231          BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
010344  001002                          BNE     14$             ;BR IF YES
010346  000137  010466                  JMP     12$             ;ELSE DO NORM SIZING OR VERIFY

010352  104401  056133          14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
010356  005037  003442                  CLR     DRIVS           ;# OF DRIVES PRESENT
010362  005000                          CLR     R0              ;DRV ADDR
010364  012701  003444                  MOV     #DRIV0,R1       ;DRV FLAG
010370  013702  001266                  MOV     $DEVM,R2        ;APT DEVICE MAP

010374  032702  000001          15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
010400  001410                          BEQ     16$             ;BR IF NO
010402  005237  003442                  INC     DRIVS           ;ELSE INCR DRIVE COUNT
010406  005211                          INC     (R1)            ;& SET DRIVE PRESENT FLAG
010410  104401  001205                  TYPE    ,$CRLF
010414  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DRIVE #
010416  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
010420     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
010421     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

010422  005721                  16$:    TST     (R1)+           ;ADV POINTER TO NEXT FLAG
010424  005200                          INC     R0              ;INC DRIVE #
010426  022700  000010                  CMP     #8.,R0          ;ALL 8 TESTED?
010432  001402                          BEQ     17$             ;BR IF YES

010434  006002                          ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
010436  000756                          BR      15$             ;& TRY AGAIN

010440  005737  003442          17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
010444  001402                          BEQ     18$             ;BR IF NO
010446  000137  011152                  JMP     VERIFY          ;ELSE EXIT TEST & SETUP FOR RK07'S

010452  104126                  18$:    ERROR   126             ;NO DRIVES FOUND IN $DEVM
010454  000000                          HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
010456  000137  007744                  JMP     ST5             ;TO TRY AGAIN
010462  000137  011152          20$:    JMP     VERIFY          ;DO NOT SIZE, GO TO NEXT TEST
010466  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
010474  013737  001414  003360          MOV     T10,TEMP1
010502  004737  043566                  JSR     PC,FRDY         ;FIND RDY
010506  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
010510  005737  003472                  TST     SIZFLG          ;SIZE BUS?
010514  001762                          BEQ     20$             ;BR IF NO
010516  104401  056133                  TYPE    ,MSG10          ;WILL TEST DRIVES
```

```
3992  010522  005037  003442                CLR     DRIVS           ;# OF DRIVES PRESENT
3993  010526  005000                        CLR     R0              ;DRV ADDR
3994  010530  012701  003444                MOV     #DRIV0,R1       ;DRV FLAG
3995  010534                        1$:
3996  010534  104415                        SCOP1
3997  010536  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
3998
3999  010542  012765  000040  000010        MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
4000  010550  013737  001414  003360        MOV     T10,TEMP1
4001  010556  004737  043566                JSR     PC,FRDY         ;FIND RDY
4002  010562  104120                        ERROR   120             ;RDY NOT SET BY END OF SCLR
4003  010564  010065  000010                MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
4004  010570  012737  000001  003322        MOV     #SELDRV,HCS1
4005  010576  053737  001170  003322        BIS     $TMP4,HCS1      ;ADD CDT IF RK07
4006  010604  013765  003322  000000        MOV     HCS1,RKCS1(R5)  ;GET STATUS
4007  010612  013737  001426  003360        MOV     T50000,TEMP1
4008  010620  004737  044256                JSR     PC,DLY          ;DO DELAY TO CATCH MDS
4009  010624  013737  001414  003360        MOV     T10,TEMP1
4010  010632  004737  043566                JSR     PC,FRDY         ;FIND RDY
4011  010636  104117                        ERROR   117             ;NO RDY AFTER SELECT DR. CMD
4012  010640  032737  100000  003322        BIT     #CERR,HCS1
4013  010646  001056                        BNE     2$
4014  010650  013737  003350  003360        MOV     HMR2,TEMP1
4015  010656  042737  177770  003360        BIC     #^C<DRVMSK>,TEMP1
4016  010664  020037  003360                CMP     R0,TEMP1        ;S/B SAME
4017  010670  001024                        BNE     3$
4018  010672  005700                        TST     R0
4019  010674  001007                        BNE     4$
4020  010676  005737  003434                TST     DDPCH           ;SEE IF XXDP CHAIN MODE
4021  010702  001022                        BNE     5$
4022  010704  123727  000041  000013        CMPB    41,#13          ;IS DRIVE 0 TO BE TESTED
4023  010712  001416                        BEQ     5$              ;BRANCH IF NOT
4024  010714  005237  003442        4$:     INC     DRIVS           ;INC DRIVE COUNT.
4025  010720  005211                        INC     (R1)            ;SET DRIVE PRESENT FLAG
4026  010722  053711  001170                BIS     $TMP4,(R1)      ;ADD CDT IF SET
4027  010726  104401  001205                TYPE    ,$CRLF
4028  010732  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
4029                                                                ;;TYPE DR #
4030  010734  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
4031  010736     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
4032  010737     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
4033  010740  000403                        BR      5$
4034
4035  010742  004737  044274        3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
4036  010746  104001                        ERROR   1               ;SELECTED DR # DOES NOT MATCH RKMR2 DR #
4037
4038  010750  005721        5$:             TST     (R1)+           ;ADVANCE PTR TO NEXT DR. FLAG
4039  010752  005200                        INC     R0              ;INC DR #
4040  010754  005037  001170                CLR     $TMP4           ;CLEAR RK07 FLAG FOR NEXT DRIVE
4041  010760  022700  000010                CMP     #8.,R0
4042  010764  001263                        BNE     1$              ;MORE LEFT.
4043  010766  005737  003442                TST     DRIVS
4044  010772  001065                        BNE     10$
4045  010774  104123                        ERROR   123             ;NO DRIVES FOUND
4046  010776  000000                        HALT                    ;SETUP CORRECTLY
4047  011000  000137  007744                JMP     ST5             ;& PRESS 'CONT'
```

```
 4048
 4049  011004  032737  000040  003336  2$:     BIT     #DTYE,HER
 4050  011012  001405                          BEQ     13$
 4051  011014  012737  002000  001170          MOV     #CDT,$TMP4      ;ADD CDT
 4052  011022  000137  010534                  JMP     1$              ;TRY AGAIN
 4053  011026  032737  001000  003324  13$:    BIT     #MDS,HCS2
 4054  011034  001015                          BNE     6$
 4055  011036  032737  000400  003324          BIT     #UFE,HCS2
 4056  011044  001015                          BNE     7$
 4057  011046  032737  000001  003334          BIT     #DRA,HDS
 4058  011054  001015                          BNE     8$
 4059  011056  032737  010000  003324          BIT     #NED,HCS2
 4060  011064  001424                          BEQ     9$
 4061  011066  000730                          BR      5$
 4062
 4063  011070  004737  044274          6$:     JSR     PC,BYP          ;TYPE BYP DR #
 4064  011074  104002                          ERROR   2               ;MDS DETECTED
 4065  011076  000724                          BR      5$
 4066
 4067  011100  004737  044274          7$:     JSR     PC,BYP
 4068  011104  104003                          ERROR   3               ;UFE DETECTED
 4069  011106  000720                          BR      5$
 4070
 4071  011110  032737  010000  003324  8$:     BIT     #NED,HCS2
 4072  011116  001676                          BEQ     4$
 4073  011120  104401  056354                  TYPE    ,MSG15          ;DRV#
 4074  011124  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 4075                                                                  ;;TYPE DR#
 4076  011126  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 4077  011130     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
 4078  011131     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 4079  011132  104010                          ERROR   10              ;DRA & NED BOTH SET
 4080  011134  000705                          BR      5$
 4081
 4082  011136  004737  044274          9$:     JSR     PC,BYP
 4083  011142  104004                          ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
 4084  011144  000701                          BR      5$
 4085  011146  000137  011544          10$:    JMP     NUDRV
 4086
 4087  011152                          VERIFY:
 4088
 4089
 4090                                  ;;*********************************************************************
 4091                                  ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
 4092                                  ;*
 4093                                  ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
 4094                                  ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
 4095                                  ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
 4096                                  ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06
 4097                                  ;*      IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
 4098                                  ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
 4099                                  ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
 4100                                  ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFO TO
 4101                                  ;*      VERIFY IT WAS NOT SPECIFIED.
 4102                                  ;*      IF CERR DUE TO DTYE , THE DRIVE WILL BE TESTED AS AN RK07.
 4103                                  ;*
```

```
4104                                        ;;*****************************************************************
4105  011152  000004                  TST3: SCOPE
4106  011154  012737  000001  001174        MOV   #1,$TIMES       ;;DO 1 ITERATION
4107  011162  012706  001100               MOV   #STACK,SP       ;RESTORE STK PTR
4108  011166  005000                        CLR   R0              ;DRIVE ADDR
4109  011170  012701  003444               MOV   #DRIV0,R1       ;DRIVE FLAG
4110  011174                           1$:
4111  011174  104415                        SCOP1
4112  011176  012706  001100               MOV   #STACK,SP       ;RESTORE STK PTR
4113
4114  011202  012765  000040  000010       MOV   #SCLR,RKCS2(R5)
4115  011210  013737  001414  003360       MOV   T10,TEMP1
4116  011216  004737  043566               JSR   PC,FRDY         ;FIND RDY
4117  011222  104120                        ERROR 120             ;NO RDY AFTER SCLR
4118  011224  010065  000010               MOV   R0,RKCS2(R5)    ;DRV ADDR
4119  011230  012737  000001  003322       MOV   #SELDRV,HCS1
4120  011236  053737  001170  003322       BIS   $TMP4,HCS1      ;ADD CDT IF RK07
4121  011244  013765  003322  000000       MOV   HCS1,RKCS1(R5)  ;GET STATUS
4122  011252  013737  001426  003360       MOV   T50000,TEMP1
4123  011260  004737  044256               JSR   PC,DLY          ;DO DELAY TO CATCH MDS
4124  011264  013737  001414  003360       MOV   T10,TEMP1
4125  011272  004737  043566               JSR   PC,FRDY         ;FIND RDY
4126  011276  104117                        ERROR 117             ;NO RDY AFTER SELDRV CMD
4127  011300  032737  100000  003322       BIT   #CERR,HCS1
4128  011306  001036                        BNE   2$
4129  011310  013737  003350  003360       MOV   HMR2,TEMP1
4130  011316  042737  177770  003360       BIC   #^C<DRVMSK>,TEMP1
4131  011324  020037  003360               CMP   R0,TEMP1        ;S/B SAME
4132  011330  001014                        BNE   3$
4133  011332  005711                  11$:  TST   (R1)
4134  011334  001402                        BEQ   4$
4135  011336  053711  001170               BIS   $TMP4,(R1)      ;ADD CDT IF SET
4136  011342  005721                  4$:   TST   (R1)+           ;SHIFT PTR TO NEXT DR FLAG
4137  011344  005200                        INC   R0              ;INC DR#
4138  011346  005037  001170               CLR   $TMP4           ;CLEAR CDT FLAG FOR NEXT DRIVE
4139  011352  022700  000010               CMP   #8.,R0
4140  011356  001306                        BNE   1$              ;MORE LEFT
4141  011360  000475                        BR    TST4            ;;GO TO NEXT TEST
4142
4143  011362  004737  044274           3$: JSR   PC,BYP          ;TRY BYPASS DRIVE#
4144  011366  104001                        ERROR 1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
4145  011370  005711                        TST   (R1)
4146  011372  001763                        BEQ   4$              ;BRANCH IF NOT SPEC BY INPUT
4147  011374  005337  003442           12$: DEC   DRIVS           ;DECREMENT TOTAL DRIVS
4148  011400  005011                        CLR   (R1)            ;CLEAR DRIVE FLAG
4149  011402  000757                        BR    4$
4150
4151  011404  032737  000040  003336  2$:  BIT   #DTYE,HER
4152  011412  001405                        BEQ   13$
4153  011414  012737  002000  001170       MOV   #CDT,$TMP4      ;ADD CDT
4154  011422  000137  011174               JMP   1$              ;TRY AGAIN
4155
4156  011426  032737  001000  003324  13$: BIT   #MDS,HCS2
4157  011434  001027                        BNE   6$
4158  011436  032737  000400  003324       BIT   #UFE,HCS2
4159  011444  001027                        BNE   7$
```

```
 4160  011446  032737  000001  003334          BIT     #DRA,HDS
 4161  011454  001005                          BNE     8$
 4162  011456  032737  010000  003324          BIT     #NED,HCS2
 4163  011464  001423                          BEQ     9$
 4164  011466  000404                          BR      10$
 4165  011470  032737  010000  003324  8$:     BIT     #NED,HCS2
 4166  011476  001715                          BEQ     11$
 4167  011500  005711                  10$:    TST     (R1)
 4168  011502  001717                          BEQ     4$
 4169
 4170  011504  004737  044274                  JSR     PC,BYP          ;TYPE BYPASS DRIVE#
 4171  011510  104006                          ERROR   6
 4172  011512  000730                          BR      12$
 4173
 4174  011514  004737  044274          6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
 4175  011520  104002                          ERROR   2               ;MDS DETECTED
 4176  011522  000724                          BR      12$
 4177
 4178  011524  004737  044274          7$:     JSR     PC,BYP
 4179  011530  104003                          ERROR   3               ;UFE DETECTED
 4180  011532  000720                          BR      12$
 4181
 4182  011534  004737  044274          9$:     JSR     PC,BYP
 4183  011540  104004                          ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
 4184  011542  000714                          BR      12$
 4185
 4186
 4187                                          ;
 4188                                          ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
 4189                                          ;DRIVE PRESENT
 4190                                          ;
 4191                                          ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
 4192                                          ;UNDER TEST
 4193
 4194  011544  005037  001462          NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
 4195                                                                  ;& TEST CERR IN 'FRDY'
 4196  011550  005037  001170                  CLR     $TMP4           ;CLEAR RK07 IND FLAG
 4197
 4198                                  ;;**************************************************************
 4199                                  ;*TEST 4        FIND NEXT DRIVE TO BE TESTED
 4200                                  ;*
 4201                                  ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
 4202                                  ;*      ADDRESS IN '$UNIT' & $TMP4 IS SET TO CDT IF DRIVE IS RK07.
 4203                                  ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
 4204                                  ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
 4205                                  ;*
 4206                                  ;;**************************************************************
 4207  011554  000004                  TST4:   SCOPE
 4208  011556  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4209  011564  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4210  011570  012737  000004  001214          MOV     #$TN-1,$TESTN
 4211  011576  012737  000004  001102          MOV     #$TN-1,$TSTNM
 4212
 4213  011604  005737  003442                  TST     DRIVS           ;ANY DRIVES PRESENT?
 4214  011610  001004                          BNE     4$              ;YES BRANCH
 4215  011612  104401  056472                  TYPE    ,MSG19          ;ALL DRIVES TESTED
```

```
                                                JMP     $EOP1           ;NO, GO TO END
                                        4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
                                                TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
                                                BEQ     2$              ;YES, BRANCH
                                        1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
                                        2$:     TST     (R1)            ;IS DRIVE PRESENT?
                                                BNE     5$              ;BR IF YES
                                                TST     (R1)+           ;ELSE FIND NEXT DRIVE
                                                BR      1$
                                        5$:     TST     DDPCH           ;DDP CHAIN MODE?
                                                BEQ     3$              ;BR IF NO
                                                TST     $UNIT           ;ELSE IS IT DRIVE 0?
                                                BNE     3$              ;BR IF NO
                                                TST     (R1)+           ;ELSE FIND NEXT DRIVE PRESENT
                                                BR      1$

                                        3$:     BIT     #CDT,(R1)+      ;SEE IF DRIVE UNDER TEST IS RK07
                                                BEQ     6$              ;BR IF NO
                                                MOV     #CDT,$TMP4      ;ELSE SET RK07 FLAG
                                        6$:     MOV     R1,DRVPTR       ;STORE POINTER TO NEXT DR FLAG
                                                TYPE    ,MSG15          ;"DRIVE"
                                                MOV     $UNIT,R0
                                                MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                        ;;DRIVE #
                                                TYPOS                   ;;GO TYPE--OCTAL ASCII
                                                .BYTE   1               ;;TYPE 1 DIGIT(S)
                                                .BYTE   0               ;;SUPPRESS LEADING ZEROS

                                        ;       TYPE    ,$CRLF

                                                TST     $TMP4           ;SEE IF RK07 UNDER TEST
                                                BNE     7$              ;BR IF YES
                                                MOV     #631,LCM1       ;ELSE LOAD RK06 PARAMETERS
                                                MOV     #632,LC
                                                MOV     #633,LCP1
                                                CLR     E.DDT
                                                MOV     #400,MC
                                                MOV     #1000,MC1
                                                MOV     #777,MASK
                                                MOV     #160017,MASK1
                                                BR      TST5            ;;GOTO NEXT TEST

                                        7$:     MOV     #1455,LCM1      ;LOAD RK07 PARAMETERS
                                                MOV     #1456,LC
                                                MOV     #1457,LCP1
                                                MOV     #D.DDT,E.DDT
                                                MOV     #1000,MC
                                                MOV     #2000,MC1
                                                MOV     #1777,MASK
                                                MOV     #140017,MASK1
                                                BR      TST5            ;;GOTO NEXT TEST

                                        LCM1:   0                       ;LC-1
                                        LC:     0                       ;LAST CYL
                                        LCP1:   0                       ;LC+1
```

```
 4272  012104  000000                  MC:     0                       ;MAJ CYL
 4273  012106  000000                  MC1:    0                       ;MAJ CYL + 1 SHIFT
 4274  012110  000000                  MASK:   0
 4275  012112  000000                  MASK1:  0
 4276  012114  000000                  E.DDT:  0                       ;EXPECTED DRIVE TYPE TO E.A0
 4277  012116  000000                  FC:     0                       ;FIND CYL
 4278  012120  000001                  FCP1:   1                       ;FC+1
 4279                                  ;;*****************************************************************
 4280                                  ;*TEST 5        UNLOAD DRIVE TO BE TESTED
 4281                                  ;*
 4282                                  ;*      THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
 4283                                  ;*      WAITS FOR ATTN & VERIFIES IT CAME FROM  THE CORRECT DRIVE.
 4284                                  ;*      IT THEN WAITS FOR SPEED OK TO GO LOW BEFORE
 4285                                  ;*      PROCEEDING TO THE NEXT TEST
 4286                                  ;*
 4287                                  ;;*****************************************************************
 4288  012122  000004                  TST5:   SCOPE
 4289  012124  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4290  012132  012706  001100                  MOV     #STACK,SP
 4291
 4292  012136  005237  003304                  INC     UNLD            ;USED TO CHECK VALID HALT
 4293
 4294  012142  004737  045462                  JSR     PC,SUBCLR
 4295  012146  104024                          ERROR   24              ;CERR AFTER SCLR
 4296
 4297  012150  012737  000007  003322          MOV     #UNLOAD,HCS1
 4298  012156  004737  043472                  JSR     PC,DOCMD        ;DO UNLOAD CMD & GET CONTR READY
 4299  012162  104011                          ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
 4300  012164  004737  044050                  JSR     PC,TSTATN
 4301  012170  104012                          ERROR   12              ;NO ATTN AFTER UNLOAD CMD
 4302
 4303  012172  004737  045462                  JSR     PC,SUBCLR
 4304  012176  104024                          ERROR   24              ;CERR AFTER SCLR
 4305
 4306  012200  013737  001414  003362          MOV     T10,TEMP2
 4307  012206  004737  046516                  JSR     PC,FSPOK
 4308  012212  104315                          ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
 4309
 4310
 4311
 4312
 4313
 4314  012214                          PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
 4315                                  .SBTTL STATIC & CYCLE UP TESTS
 4316
 4317                                  ;;*****************************************************************
 4318                                  ;*TEST 6        REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3
 4319                                  ;*
 4320                                  ;*       CHECKS THE ABILITY TO REFERENCE ALL
 4321                                  ;*      DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.
 4322                                  ;*
 4323                                  ;;*****************************************************************
 4324  012214  000004                  TST6:   SCOPE
 4325  012216  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4326  012224  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4327
```

```
 4328  012230  004737  045462                    JSR     PC,SUBCLR
 4329  012234  104024                            ERROR   24              ;CERR AFTER SCLR
 4330
 4331  012236  004737  045132                    JSR     PC,GSTAT
 4332  012242  032737  000100  003350            BIT     #D.VV,HMR2
 4333  012250  001004                            BNE     4$              ;BR IF VV SET
 4334  012252  012737  000040  003412            MOV     #D.DRA,E.A0     ;LOAD EXPECTED VALUE FOR A0
 4335  012260  000403                            BR      5$
 4336
 4337  012262  012737  000140  003412  4$:       MOV     #<D.DRA!D.VV>,E.A0
 4338  012270  005037  003414        5$:       CLR     E.B0            ;EXPECTED MSG B0
 4339  012274  012737  000740  003416            MOV     #<D.HDHM!D.BRHM!D.DOOR!D.CART>,E.A1      ;EXPECTED MSG A1
 4340  012302  012737  000001  003420            MOV     #1,E.B1         ;EXPECTED MSG B1
 4341  012310  005037  003422                    CLR     E.A2            ;EXPECTED MSG A2
 4342  012314  012737  000002  003424            MOV     #2,E.B2         ;EXPECTED MSG B2
 4343  012322  012737  000003  003430            MOV     #3,E.B3         ;EXPECTED MSG B3
 4344
 4345  012330  004737  044310                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4346  012334  000007                            .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
 4347  012336  104016                            ERROR   16              ;MSG A0 ERROR FOR DRIVE UNLOADED
 4348  012340  104017                            ERROR   17              ;MSG B0 ERROR
 4349  012342  104020                            ERROR   20              ;MSG A1 ERROR
 4350  012344  104021                            ERROR   21              ;MSG B1 ERROR
 4351
 4352  012346  012765  000002  000026            MOV     #2,RKMR1(R5)    ;SELECT WORD 2
 4353  012354  004737  045132                    JSR     PC,GSTAT
 4354  012360  005737  001362                    TST     CYLDIF          ;SEE IF MSG A2=0
 4355  012364  001401                            BEQ     64$             ;BR IF YES
 4356  012366  104022                            ERROR   22              ;MSG A2 NOT CLEARED FOR DRIVE UNLOADED
 4357  012370  005737  001364        64$:      TST     CYLADD          ;SEE IF MSG B2=0
 4358  012374  001401                            BEQ     65$             ;BR IF YES
 4359  012376  104023                            ERROR   23              ;MSG B2 NOT CLEARED FOR DRIVE UNLOADED
 4360  012400                        65$:
 4361  012400  023727  001432  000001            CMP     HEADA,#1        ;FOR HEAD 0, B3=1
 4362  012406  001401                            BEQ     TST7            ;;GO TO NXT TST IF YES
 4363  012410  104056                            ERROR   56              ;HEAD REG IN B3 NOT 0 IN UNLOAD
 4364
 4365
 4366
 4367                                    ;;*************************************************************************
 4368                                    ;*TEST 7        PRINT DRIVE SERIAL NUMBER
 4369                                    ;*
 4370                                    ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
 4371                                    ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
 4372                                    ;*
 4373                                    ;;*************************************************************************
 4374  012412  000004                    TST7:   SCOPE
 4375  012414  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
 4376  012422  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
 4377
 4378  012426  005737  001216                    TST     $PASS
 4379  012432  001046                            BNE     TST10           ;;GO TO NEXT IF NOT FIRST PASS
 4380  012434  004737  045462                    JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
 4381  012440  104024                            ERROR   24              ;CERR AFTER SCLR
 4382
 4383  012442  104401  056366                    TYPE    ,MSG16          ;DRIVE SERIAL NO.
```

```
 4384  012446  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
 4385  012454  004737  045132                  JSR     PC,GSTAT        ;GET STATUS
 4386  012460  013701  003350                  MOV     HMR2,R1         ;GET SERIAL #
 4387  012464  012704  054002                  MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
 4388  012470  010446                          MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
 4389  012472  012703  000003                  MOV     #3,R3           ;SETUP CHAR COOUNT
 4390  012476  006101                          ROL     R1              ;INITIALIZE BIT POSITIONS
 4391  012500  006101                          ROL     R1
 4392  012502  006101                  1$:     ROL     R1              ;GET NEXT 4 BITS
 4393  012504  006101                          ROL     R1
 4394  012506  006101                          ROL     R1
 4395  012510  006101                          ROL     R1
 4396  012512  010100                          MOV     R1,R0           ;GET WORKING COPY
 4397  012514  042700  177760                  BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
 4398  012520  052700  000060                  BIS     #60,R0          ;CONVERT TO ASCII DIGIT
 4399  012524  110024                          MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
 4400  012526  005303                          DEC     R3
 4401  012530  001364                          BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
 4402  012532  105014                          CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR
 4403  012534  004737  054250                  JSR     PC,$SUPRS       ;TYPE
 4404  012540  104401  001205                  TYPE    ,$CRLF
 4405  012544  104401  001205                  TYPE    ,$CRLF
 4406
 4407                                  ;;**************  ************************************************
 4408                                  ;*TEST 10        SET VV WITH PACK CMD
 4409                                  ;*
 4410                                  ;*      IF VV IS RESET, THE PACK CMD IS USED TO SET IT.
 4411                                  ;*
 4412                                  ;;*****************************************************************
 4413  012550  000004                  TST10:  SCOPE
 4414  012552  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4415  012560  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4416
 4417  012564  005065  000026                  CLR     RKMR1(R5)       ;SELECT BYTE 0
 4418  012570  004737  045132                  JSR     PC,GSTAT        ;GET STATUS
 4419  012574  032737  000100  003350          BIT     #D.VV,HMR2
 4420  012602  001021                          BNE     TST11           ;;GO TO NEXT TEST IF VV SET
 4421
 4422  012604  104415                          SCOP1
 4423  012606  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4424
 4425  012612  004737  045462                  JSR     PC,SUBCLR
 4426  012616  104024                          ERROR   24              ;CERR AFTER SCLR
 4427
 4428  012620  012737  000003  003322          MOV     #PACK,HCS1
 4429  012626  004737  043472                  JSR     PC,DOCMD        ;DO PACK CMD & GET CONTR READY
 4430  012632  104116                          ERROR   116             ;RDY NOT SET AFTER PACK CMD
 4431
 4432  012634  032737  000100  003350          BIT     #D.VV,HMR2
 4433  012642  001001                          BNE     TST11           ;;GO TO NEXT TEST IF VV NOW SET
 4434  012644  104027                          ERROR   27              ;PACK DID NOT SET V.V.
 4435
 4436                                  ;;*****************************************************************
 4437                                  ;*TEST 11        RELEASE DRIVE
 4438                                  ;*
 4439                                  ;*      TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK
```

```
4440                                        ;*
4441                                        ;;*****************************************************************
4442  012646  000004                        TST11:  SCOPE
4443  012650  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
4444  012656  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR
4445
4446  012662  004737  045462                        JSR     PC,SUBCLR       ;DO SUBSYS CLEAR & GET STATUS
4447  012666  104024                                ERROR   24              ;CONTR ERROR SET AFTER SCLR
4448  012670  032737  000400  003324                BIT     #UFE,HCS2
4449  012676  001401                                BEQ     1$
4450  012700  104003                                ERROR   3               ;UFE SET AFTER SCLR
4451
4452  012702                                1$:
4453  012702  104415                                SCOP1
4454  012704  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR
4455
4456  012710  004737  045462                        JSP     PC,SUBCLR
4457  012714  104024                                ERROR   24              ;CERR AFTER SCLR
4458
4459  012716  062765  000010  000010                ADD     #RLS,RKCS2(R5)  ;ADD RELEASE BIT TO $UNIT
4460  012724  004737  045132                        JSR     PC,GSTAT        ;GET STATUS
4461
4462  012730  032737  100000  003322                BIT     #CERR,HCS1      ;CHECK FOR CONTR ERROR
4463  012736  001401                                BEQ     2$
4464  012740  104025                                ERROR   25              ;RLS SET CERR
4465  012742  032737  000400  003324  2$:           BIT     #UFE,HCS2
4466  012750  001401                                BEQ     TST12           ;;GO TO NEXT TEST IF SET
4467  012752  104026                                ERROR   26              ;SACK SET AFTER RLS SENT
```

```
4468
```

```
                                        ;;*****************************************************************
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
012754  000004                TST12:  SCOPE
012756  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
012764  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

012770  004737  045462                JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
012774  104024                        ERROR   24              ;CONT ERROR SET AFT SUBSYS CLEAR
012776  005737  001170                TST     $TMP4
013002  001013                        BNE     1$
013004  032737  000400  003350        BIT     #D.DDT,HMR2
013012  001401                        BEQ     2$
013014  104030                        ERROR   30              ;DR TYPE SET IN MR2
013016  032737  000400  003334  2$:   BIT     #DDT,HDS
013024  001414                        BEQ     3$
013026  104031                        ERROR   31              ;DDT SET IN RKDS
013030  000412                        BR      3$

013032  032737  000400  003350  1$:   BIT     #D.DDT,HMR2
013040  001001                        BNE     8$
013042  104005                        ERROR   5               ;DR TYPE NOT SET IN MR2
013044  032737  000400  003334  8$:   BIT     #DDT,HDS
013052  001001                        BNE     3$
013054  104152                        ERROR   152
013056  032737  000040  003336  3$:   BIT     #DTYE,HER
013064  001401                        BEQ     4$
013066  104032                        ERROR   32              ;DTYE SET IN RKER

013070                          4$:
013070  104415                        SCOP1
013072  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

013076  004737  045462                JSR     PC,SUBCLR
013102  104024                        ERROR   24              ;CERR AFTER SCLR

013104  005737  001170                TST     $TMP4           ;SEE IF RK07
013110  001404                        BEQ     9$              ;BR IF NO
013112  012765  000001  000000        MOV     #SELDRV,RKCS1(R5)
013120  000403                        BR      10$
013122  012765  002001  000000  9$:   MOV     #<CDT!SELDRV>,RKCS1(R5)  ;GET STATUS WITH CDT SET
013130  013737  001414  003360  10$:  MOV     T10,TEMP1
013136  004737  043566                JSR     PC,FRDY         ;FIND RDY
013142  104117                        ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
013144  005737  001170                TST     $TMP4
013150  001013                        BNE     11$
013152  032737  000400  003350        BIT     #D.DDT,HMR2
013160  001401                        BEQ     5$
013162  104030                        ERROR   30              ;DR TYPE SET IN MR2
013164  032737  000400  003334  5$:   BIT     #DDT,HDS
013172  001414                        BEQ     6$
```

```
 4525  013174  104031                          ERROR   31              ;DDT SET IN RKDS
 4526  013176  000412                          BR      6$
 4527
 4528  013200  032737  000400  003350  11$:    BIT     #D.DDT,HMR2
 4529  013206  001001                          BNE     12$
 4530  013210  104005                          ERROR   5               ;DR TYPE NOT SET IN MR2
 4531  013212  032737  000400  003334  12$:    BIT     #DDT,HDS
 4532  013220  001001                          BNE     6$
 4533  013222  104152                          ERROR   152             ;DDT NOT SET IN RKDS
 4534  013224  032737  000040  003336  6$:     BIT     #DTYE,HER       ;DTYE=DDT(NOT)*CDT
 4535  013232  001001                          BNE     7$
 4536  013234  104033                          ERROR   33              ;DTYE NOT SET AFT WRITING WRONG CDT
 4537  013236  032737  100000  003322  7$:     BIT     #CERR,HCS1
 4538  013244  001001                          BNE     TST13           ;;GO TO NEXT TEST
 4539  013246  104034                          ERROR   34              ;CERR NOT SET AFT WRITING WRONG CDT
 4540
 4541                                  ;;*******************************************************************
 4542                                  ;*TEST 13       C-D PARITY ERROR DETECTION
 4543                                  ;*
 4544                                  ;*      TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
 4545                                  ;*      THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
 4546                                  ;*      THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
 4547                                  ;*      THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
 4548                                  ;*      CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
 4549                                  ;*      AN ERROR AND SET DCPAR.
 4550                                  ;*      THE ERROR CONDITION IS RESET WITH THE CLEAR CMD
 4551                                  ;*
 4552                                  ;;*******************************************************************
 4553  013250  000004                  TST13:  SCOPE
 4554  013252  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4555  013260  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4556
 4557  013264  004737  045462                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
 4558  013270  104024                          ERROR   24              ;CONT ERROR AFTER SUBSYS CLR
 4559  013272  032737  001000  003352          BIT     #D.PAR,HMR3
 4560  013300  001401                          BEQ     2$
 4561  013302  104035                          ERROR   35              ;C-D PARITY ERROR SET IN MR3
 4562  013304  032737  020000  003322  2$:     BIT     #DCPAR,HCS1
```

```
 4563  013312  001401                        BEQ     3$
 4564  013314  104036                        ERROR   36              ;DCPAR SET IN CS1
 4565
 4566  013316                          3$:
 4567  013316  104415                        SCOP1
 4568  013320  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 4569
 4570  013324  004737  045462                JSR     PC,SUBCLR
 4571  013330  104024                        ERROR   24              ;CERR AFTER SCLR
 4572
 4573  013332  012765  000020  000026        MOV     #PAT,RKMR1(R5)  ;SELECT BYTE 0 & EVEN PARITY
 4574  013340  004737  045132                JSR     PC,GSTAT        ;GET STATUS
```

```
 4575  013344  032737  000200  003352          BIT     #D.FLT,HMR3
 4576  013352  001001                          BNE     4$
```

```
                                        ERROR   37              ;FAULT NOT SET IN MR3
013354  104037
013356  032737  001000  003352  4$:     BIT     #D.PAR,HMR3
013364  001001                          BNE     5$
013366  104040                          ERROR   40              ;C-D PARITY ERROR NOT SET IN MR3
013370  032737  020000  003322  5$:     BIT     #DCPAR,HCS1
013376  001001                          BNE     6$
013400  104041                          ERROR   41              ;DCPAR NOT SET AFT WRITING PAT IN MR1
013402  032737  100000  003322  6$:     BIT     #CERR,HCS1
013410  001001                          BNE     TST14           ;;GO TO NEXT TEST
013412  104042                          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.
                                ;*
                                ;;******************************************************************
013414  000004                  TST14:  SCOPE
013416  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
013424  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

013430  004737  045462                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
013434  104024                          ERROR   24              ;CERR AFTER SCLR

013436  012737  000011  003322          MOV     #SRTSPL,HCS1
013444  004737  043472                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
013450  104121                          ERROR   121             ;RDY NOT SET AFTER START SPIN CMD

013452  004737  045132                  JSR     PC,GSTAT        ;WORD 0
013456  032737  010000  003350          BIT     #D.SPIN,HMR2
013464  001001                          BNE     13$
013466  104306                          ERROR   306             ;SPIN NOT SET AFTER START SPIN CMD

013470  012737  014666  001176  13$:    MOV     #25$,$ESCAPE
013476  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
013502  000401                          BR      15$
013504  104316                          ERROR   316
013506  012737  010140  003412  15$:    MOV     #<D.SPIN!D.VV!D.DRA>,E.A0       ;LOAD IN EXPECTED VALUES
013514  005037  003414                  CLR     E.B0
013520  012737  000740  003416          MOV     #<D.CART!D.DOOR!D.HDHM!D.BRHM>,E.A1
013526  012737  000001  003420          MOV     #1,E.B1

013534  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
013540  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
013542  104057                          ERROR   57              ;MSG A0 ERROR AFTER START SPIN CMD REC'D BY DRIVE
013544  104060                          ERROR   60              ;MSG B0 ERROR
013546  104061                          ERROR   61              ;MSG A1 ERROR
013550  104062                          ERROR   62              ;MSG B1 ERROR
013552  005737  003470                  TST     DOTIM
013556  001126                          BNE     3$              ;BRANCH IF P OR L CLOCK PRESENT
013560  012737  014712  001176  1$:     MOV     #30$,$ESCAPE
013566  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
 4633  013574  013737  001416  003362          MOV     T100,TEMP2      ;SETUP TIMEOUT
 4634  013602  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
 4635  013606  104067                          ERROR   67              ;NO ATTN AFTER HEAD LOADING
 4636  013610                          2$:
 4637
 4638  013610  012737  050340  003412          MOV     #<D.LSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4639  013616  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 4640  013622  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4641  013630  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4642  013636  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 4643  013642  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4644  013650  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4645
 4646  013656  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4647  013662  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4648  013664  104063                          ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOAD
 4649  013666  104064                          ERROR   64              ;MSG B0 ERROR
 4650  013670  104065                          ERROR   65              ;MSG A1 ERROR
 4651  013672  104066                          ERROR   66              ;MSG B1 ERROR
 4652  013674  005737  001364                  TST     CYLADD          ;SEE IF MSG B2 = 0
 4653  013700  001401                          BEQ     16$             ;BR IF YES
 4654  013702  104176                          ERROR   176             ;CYL ADDR NOT 0 AT END OF HEAD LOAD
 4655  013704                          16$:
 4656
 4657  013704  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4658  013712  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4659  013720  012737  000005  003322          MOV     #CLEAR,HCS1
 4660  013726  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 4661  013732  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4662  013734  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4663  013740  000401                          BR      64$
 4664  013742  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4665  013744                          64$:
 4666
 4667  013744  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 4668  013752  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 4669  013756  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4670  013764  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4671  013772  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 4672  013776  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4673  014004  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4674
 4675  014012  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4676  014016  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4677  014020  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 4678  014022  104265                          ERROR   265             ;MSG B0 ERROR
 4679  014024  104274                          ERROR   274             ;MSG A1 ERROR
 4680  014026  104266                          ERROR   266             ;MSG B1 ERROR
 4681
 4682  014030  000137  014506                  JMP     12$
 4683
 4684  014034  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
 4685  014042  013737  001370  001372          MOV     HZ,COUNT
 4686  014050  012737  000074  001374          MOV     #60.,SEC
 4687  014056  004737  047244                  JSR     PC,CLKON        ;TURN CLK INTR ON FOR 60 SEC MAX
 4688  014062  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
```

```
 4689  014070  004737  045132          4$:   JSR     PC,GSTAT
 4690  014074  032737  002000  003350        BIT     #D.FWD,HMR2
 4691  014102  001004                        BNE     5$
 4692  014104  005737  001376                TST     TIMUP           ;IS 60 SEC DELAY UP?
 4693  014110  001767                        BEQ     4$              ;BRANCH IF NO & REPEAT
 4694  014112  104070                        ERROR   70              ;FWD NOT SET WITHIN 60 SEC FROM
 4695                                                                ;START SPINDLE CMD.
 4696  014114  004737  047340          5$:   JSR     PC,CLKOF        ;TURN OFF CLOCK INTERRUPT
 4697  014120  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 4698
 4699  014126  013737  001370  001372        MOV     HZ,COUNT
 4700  014134  012737  000005  001374        MOV     #5,SEC
 4701  014142  004737  047244                JSR     PC,CLKON        ;TURN CLK INTR ON FOR 5 SEC MAX
 4702  014146  012765  000001  000026  6$:   MOV     #1,RKMR1(R5)    ;WORD 1
 4703  014154  004737  045132                JSR     PC,GSTAT
 4704  014160  032737  002000  003350        BIT     #D.FWD,HMR2
 4705  014166  001404                        BEQ     7$
 4706  014170  005737  001376                TST     TIMUP
 4707  014174  001764                        BEQ     6$
 4708  014176  104075                        ERROR   75              ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
 4709                                                                ;FROM START SPINDLE CMD.
 4710  014200  004737  047340          7$:   JSR     PC,CLKOF        ;TURN OFF CLK INTERRUPT
 4711  014204  004737  044050                JSR     PC,TSTATN       ;TEST FOR ATTN
 4712  014210  000401                        BR      17$
 4713  014212  104320                        ERROR   320             ;UNEXP ATTN AFTER INNER LIM DETECT
 4714  014214  012737  030140  003412  17$:   MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 4715  014222  005037  003414                CLR     E.B0
 4716  014226  012737  025720  003416        MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 4717  014234  012737  000001  003420        MOV     #1,E.B1
 4718
 4719  014242  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4720  014246  000000                        .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 4721  014250  104076                        ERROR   76              ;MSG A0 ERROR AT INNER LIMIT DETECT
 4722  014252  104077                        ERROR   77              ;MSG B0 ERROR
 4723  014254  104100                        ERROR   100             ;MSG A1 ERROR
 4724  014256  104101                        ERROR   101             ;MSG B1 ERROR
 4725
 4726  014260  013737  001370  001372        MOV     HZ,COUNT
 4727  014266  012737  000004  001374        MOV     #4,SEC
 4728  014274  004737  047244                JSR     PC,CLKON        ;TURN CLK INTR ON FOR 4 SEC MAX
 4729  014300  012765  000001  000026  8$:   MOV     #1,RKMR1(R5)    ;WORD 1
 4730  014306  004737  045132                JSR     PC,GSTAT
 4731  014312  032737  002000  003350        BIT     #D.FWD,HMR2
 4732  014320  001004                        BNE     9$
 4733  014322  005737  001376                TST     TIMUP
 4734  014326  001764                        BEQ     8$
 4735  014330  104102                        ERROR   102             ;FWD NOT DETECTED WITHIN 4 SEC IN RTZ PORTION OF
 4736                                                                ;START SPINDLE CMD.
 4737  014332  004737  047340          9$:   JSR     PC,CLKOF        ;TURN CLOCK INTR OFF.
 4738  014336  004737  044050                JSR     PC,TSTATN       ;TEST FOR ATTN
 4739  014342  000401                        BR      18$
 4740  014344  104321                        ERROR   321             ;UNEXP ATTN AFTER OUTER LIM TO CYL 0
 4741  014346  012737  030140  003412  18$:   MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 4742  014354  005037  003414                CLR     E.B0
 4743  014360  012737  023720  003416        MOV     #<D.RTZ!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 4744  014366  012737  000001  003420        MOV     #1,E.B1
```

```
 4745
 4746  014374  004737  044310            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4747  014400  000000                    .WORD   0!0'0           ;& MSGS SPECIFIED HERE
 4748  014402  104103                    ERROR   103             ;MSG A0 ERROR FROM OUT LIM TO CYL 0 DURING LOAD
 4749  014404  104104                    ERROR   104             ;MSG B0 ERROR
 4750  014406  104105                    ERROR   105             ;MSG A1 ERROR
 4751  014410  104106                    ERROR   106             ;MSG B1 ERROR
 4752  014412  013737  001370  001372    MOV     HZ,COUNT
 4753  014420  012737  000001  001374    MOV     #1,SEC
 4754  014426  004737  047244            JSR     PC,CLKON        ;TURN CLK INTR ON FOR 1 SEC MAX
 4755  014432  005065  000026      10$:  CLR     RKMR1(R5)       ;WORD 0
 4756  014436  004737  045132            JSR     PC,GSTAT
 4757  014442  032737  000200  003350    BIT     #D.DRDY,HMR2    ;SEE IF DRIVE READY
 4758  014450  001004                    BNE     11$
 4759  014452  005737  001376            TST     TIMUP
 4760  014456  001765                    BEQ     10$
 4761  014460  104107                    ERROR   107             ;DRIVE READY NOT SET WITHIN 1 SEC FROM
 4762                                                            ;FWD IN RTZ PORTION OF START SPIN CMD
 4763  014462  004737  047340      11$:  JSR     PC,CLKOF        ;TURN CLOCKS OFF
 4764  014466  004737  044050            JSR     PC,TSTATN       ;TEST ATTN
 4765  014472  104067                    ERROR   67              ;NO ATTN AFTER START SPIN CMD
 4766  014474  012737  014712  001176    MOV     #30$,$ESCAPE
 4767  014502  000137  013610            JMP     2$              ;CHECK RKMR 2 & 3 WORDS 0 & 1
 4768
 4769  014506  005037  001176      12$:  CLR     $ESCAPE
 4770  014512  012765  100000  000000    MOV     #CCLR,RKCS1(R5)
 4771  014520  005065  000026            CLR     RKMR1(R5)
 4772  014524  004737  045132            JSR     PC,GSTAT
 4773  014530  032737  010000  003350    BIT     #D.SPIN,HMR2
 4774  014536  001003                    BNE     14$
 4775  014540  104306                    ERROR   306             ;SPIN NOT SET AFTER ST. SPIN CMD
 4776  014542  000137  042706            JMP     $EOP            ;ABORT DRIVE
 4777  014546  005037  003304      14$:  CLR     UNLD            ;USED FOR VALID HALT
 4778  014552  004737  047466            JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 4779  014556  000467                    BR      TST15           ;;GO TO NEXT TEST
 4780                                                            ;RETURN HERE IF SW 14 IS SET OR
 4781                                                            ;SW 8 WITH SWR <7:0> APPLY
 4782  014560  005237  003304      20$:  INC     UNLD
 4783
 4784  014564  004737  045462            JSR     PC,SUBCLR
 4785  014570  104024                    ERROR   24              ;CERR AFTER SCLR
 4786
 4787  014572  012737  000007  003322    MOV     #UNLOAD,HCS1
 4788  014600  004737  043472            JSR     PC,DOCMD        ;DO UNLOAD CMD & GET CONTR READY
 4789  014604  104011                    ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
 4790  014606  004737  044050            JSR     PC,TSTATN
 4791  014612  104012                    ERROR   12              ;NO ATTN AFTER UNLOAD CMD
 4792
 4793  014614  004737  045462            JSR     PC,SUBCLR
 4794  014620  104024                    ERROR   24              ;CERR AFTER SCLR
 4795
 4796  014622  013737  001414  003362    MOV     T10,TEMP2
 4797  014630  004737  046516            JSR     PC,FSPOK
 4798  014634  104315                    ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
 4799
 4800  014636  004737  045462            JSR     PC,SUBCLR
```

```
                                ERROR   24              ;CERR AFTER SCLR

                                CLR     $ESCAPE
                                TST     LPFLG
                                BEQ     65$
                                JMP     @$LPERR         ;SW 9 WAS SET.
                        65$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET


                        25$:    JSR     PC,CLKOF
                                INC     LPFLG
                                BIT     #SW9,@SWR       ;LOOP ON ERROR?
                                BNE     20$             ;YES, RECONDITION DRIVE
                                JMP     1$              ;RETURN TO MAINLINE
                        30$:    JSR     PC,CLKOF
                                INC     LPFLG
                                BIT     #SW9,@SWR       ;LOOP ON ERROR?
                                BNE     20$             ;YES, RECONDITION DRIVE
                                JMP     12$             ;RETURN TO MAINLINE
                        .SBTTL 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) (512 FOR RK07) WITH EVEN PARITY SET.
                        ;*      THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKING.
                        ;*      THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
                        ;*      NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.
                        ;*
                        ;;*******************************************************************
                        TST15:  SCOPE
                                MOV     #1,$TIMES       ;;DO 1 ITERATION
                                MOV     #STACK,SP       ;RESTORE STK PTR

                                CLR     R0              ;CYL # REGISTER
                                MOV     #BIT15,TEMP5

                        1$:
                                SCOP1
                                MOV     #STACK,SP       ;RESTORE STK PTR

                                JSR     PC,SUBCLR
                                ERROR   24              ;CERR AFTER SCLR


                                MOV     #CCLR,RKCS1(R5)
                                MOV     $UNIT,RKCS2(R5)
                                MOV     #RECAL,HCS1
                                JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
                                ERROR   124             ;RDY NOT SET AFTER RECAL CMD

                                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                                JSR     PC,GSTAT
```

```
 4857  015040  032737  020000  003350          BIT     #D.RTZ,HMR2
 4858  015046  001001                          BNE     64$
 4859  015050  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
 4860  015052  013737  001414  003362  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
 4861  015060  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
 4862  015064  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
 4863
 4864  015066  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4865  015074  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 4866  015100  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4867  015106  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4868  015114  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 4869  015120  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4870  015126  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4871
 4872  015134  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4873  015140  000007                          .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
 4874  015142  104221                          ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
 4875  015144  104275                          ERROR   275             ;MSG B0 ERROR
 4876  015146  104222                          ERROR   222             ;MSG A1 ERROR
 4877  015150  104276                          ERROR   276             ;MSG B1 ERROR
 4878                          ;                 LWD2    47,50,<AFTER RECAL CMD>
 4879  015152  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT THE MESSAGE
 4880  015160  004737  045132                  JSR     PC,GSTAT        ;GET THE STATUS
 4881  015164  005737  001364                  TST     CYLADD          ;RECAL SUCCESSFUL ?
 4882  015170  001401                          BEQ     65$             ;BRANCH IF SO
 4883  015172  104050                          ERROR   50              ;REPORT THE ERROR
 4884  015174                          65$:
 4885
 4886  015174  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4887  015202  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4888  015210  012737  000005  003322          MOV     #CLEAR,HCS1
 4889  015216  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 4890  015222  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4891  015224  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4892  015230  000401                          BR      66$
 4893  015232  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4894  015234                          66$:
 4895
 4896  015234  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 4897  015242  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 4898  015246  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4899  015254  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4900  015262  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 4901  015266  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4902  015274  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4903
 4904  015302  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4905  015306  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4906  015310  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 4907  015312  104265                          ERROR   265             ;MSG B0 ERROR
 4908  015314  104274                          ERROR   274             ;MSG A1 ERROR
 4909  015316  104266                          ERROR   266             ;MSG B1 ERROR
 4910
 4911
 4912  015320  104415                          SCOP1
```

```
 4913  015322  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 4914
 4915  015326  004737  045462                JSR     PC,SUBCLR
 4916  015332  104024                        ERROR   24              ;CERR AFTER SCLR
 4917
 4918  015334  005237  001462                INC     BYPCERR         ;DO NOT TEST CERR IN GSTAT1
 4919  015340  012765  000020  000026        MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
 4920  015346  010065  000020                MOV     R0,RKDC(R5)     ;CYL ADDR
 4921  015352  012737  000017  003322        MOV     #SEEK,HCS1
 4922  015360  004737  043472                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR RDY
 4923  015364  104122                        ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
 4924  015366  004737  044050                JSR     PC,TSTATN       ;TEST FOR ATTN
 4925  015372  104125                        ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
 4926  015374  012737  050340  003412        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
 4927  015402  012737  001200  003414        MOV     #<D.FLT!D.PAR>,E.B0
 4928  015410  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 4929  015416  012737  000001  003420        MOV     #1,E.B1
 4930  015424  010037  003422                MOV     R0,E.A2
 4931  015430  006137  003422                ROL     E.A2
 4932  015434  006137  003422                ROL     E.A2
 4933  015440  006137  003422                ROL     E.A2
 4934  015444  006137  003422                ROL     E.A2
 4935  015450  013737  003422  003424        MOV     E.A2,E.B2
 4936  015456  052737  000002  003424        BIS     #2,E.B2         ;ADD MSG ID
 4937
 4938  015464  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4939  015470  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4940  015472  104110                        ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
 4941  015474  104111                        ERROR   111             ;MSG B0 ERROR
 4942  015476  104146                        ERROR   146             ;MSG A1 ERROR
 4943  015500  104147                        ERROR   147             ;MSG B1 ERROR
 4944
 4945  015502  020037  001364                CMP     R0,CYLADD
 4946  015506  001401                        BEQ     2$
 4947  015510  104043                        ERROR   43              ;CYL ADDR IN B2 NOT=RKDC
 4948
 4949  015512  020037  001362        2$:     CMP     R0,CYLDIF
 4950  015516  001401                        BEQ     3$
 4951  015520  104044                        ERROR   44              ;CYL DIFF IN A2 NOT=RKDC
 4952
 4953  015522  005037  001462        3$:     CLR     BYPCERR         ;ALLOW CHECKING FOR ANY CERR IN GSTAT1
 4954  015526  006137  003370                ROL     TEMP5           ;SET CARRY ONLY ONCE
 4955  015532  006100                        ROL     R0              ;SELECT NEXT MAJOR CYL
 4956  015534  020037  012106                CMP     R0,MC1          ;ALL MAJOR CYL DONE?
 4957  015540  001001                        BNE     4$              ;BRANCH IF NO
 4958  015542  000402                        BR      TST16           ;;GO TO  NEXT TST
 4959  015544  000137  014762        4$:     JMP     1$
 4960
 4961                                ;;*************************************************************
 4962                                ;*TEST 16       STATIC CYL DIFF & CYL ADDR REG TEST-PART 2
 4963                                ;*
 4964                                ;*      THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
 4965                                ;*      DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
 4966                                ;*      ALL CYLS FROM EVERY OTHER CYL. (N SQUARE SEEKS).
 4967                                ;*      IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.
 4968                                ;*
```

```
                                ;;*****************************************************************
015550  000004                  TST16:  SCOPE
015552  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
015560  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
015564  005737  001340                  TST     BYPT16
015570  001404                          BEQ     13$
015572  104401  056105                  TYPE    ,MSG9           ;BYPASSING TEST 16
015576  000137  017012                  JMP     12$
015602  005737  001170          13$:    TST     $TMP4           ;SEE IF RK07
015606  001403                          BEQ     15$             ;BR OF NO
015610  104401  057001                  TYPE    ,MSG23          ;10-12 MIN TEST
015614  000402                          BR      16$
015616  104401  056047          15$:    TYPE    ,MSG8           ;2-4 MIN TEST
015622  005037  001350          16$:    CLR     FRCYL           ;FROM CYL
015626  005037  001352                  CLR     TOCYL           ;TO CYL
015632  005037  001354                  CLR     CCYL            ;CURRENT CYL
015636  005037  001356                  CLR     PCYL            ;PREV CYL
015642  104415                          SCOP1
015644  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
015650  004737  045462                  JSR     PC,SUBCLR
015654  104024                          ERROR   24              ;CERR AFTER SCLR
015656  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
015664  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
015672  012737  000013  003322          MOV     #RECAL,HCS1
015700  004737  043472                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
015704  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
015706  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
015714  004737  045132                  JSR     PC,GSTAT
015720  032737  020000  003350          BIT     #D.RTZ,HMR2
015726  001001                          BNE     64$
015730  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
015732  013737  001414  003362  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
015740  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
015744  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
015746  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
015754  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
015760  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
015766  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
015774  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
016000  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
016006  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
016014  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
016020  000007                          .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
016022  104221                          ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
016024  104275                          ERROR   275             ;MSG B0 ERROR
016026  104222                          ERROR   222             ;MSG A1 ERROR
```

```
5025  016030  104276                          ERROR   276             ;MSG B1 ERROR
5026                              ;           CWD2    47,50,<AFTER RECAL CMD>
5027  016032  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT THE MESSAGE
5028  016040  004737  045132                  JSR     PC,GSTAT        ;GET THE STATUS
5029  016044  005737  001364                  TST     CYLADD          ;RECAL SUCCESSFUL ?
5030  016050  001401                          BEQ     65$             ;BRANCH IF SO
5031  016052  104050                          ERROR   50              ;REPORT THE ERROR
5032  016054                            65$:
5033
5034  016054  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5035  016062  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5036  016070  012737  000005  003322          MOV     #CLEAR,HCS1
5037  016076  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5038  016102  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5039  016104  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
5040  016110  000401                          BR      66$
5041  016112  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5042  016114                            66$:
5043
5044  016114  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5045  016122  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
5046  016126  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
5047  016134  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5048  016142  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5049  016146  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5050  016154  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5051
5052  016162  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5053  016166  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5054  016170  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5055  016172  104265                          ERROR   265             ;MSG B0 ERROR
5056  016174  104274                          ERROR   274             ;MSG A1 ERROR
5057  016176  104266                          ERROR   266             ;MSG B1 ERROR
5058
5059
5060  016200  104415                          SCOP1
5061  016202  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5062
5063  016206  004737  045462                  JSR     PC,SUBCLR
5064  016212  104024                          ERROR   24              ;CERR AFTER SCLR
5065
5066
5067  016214  012765  000020  000026  1$:     MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
5068  016222  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;SET TO CYL ADDR
5069  016230  013737  001352  001354          MOV     TOCYL,CCYL      ;CURRENT CYL
5070  016236  013737  001354  003364          MOV     CCYL,TEMP3
5071  016244  013737  001356  003366          MOV     PCYL,TEMP4      ;PREV CYL
5072  016252  163737  003364  003366          SUB     TEMP3,TEMP4
5073  016260  100002                          BPL     2$              ;BR IF TEMP4 IS POS
5074  016262  005437  003366                  NEG     TEMP4
5075  016266  013737  003366  001360  2$:     MOV     TEMP4,CALDIF
5076  016274  013737  001354  001356          MOV     CCYL,PCYL
5077  016302  012737  000017  003322          MOV     #SEEK,HCS1
5078  016310  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR RDY
5079  016314  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
5080  016316  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
```

```
 5081  016322  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
 5082  016324  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
 5083  016332  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
 5084  016340  004737  046052                  JSR     PC,RDCYLA       ;READ CYL ADDR
 5085  016344  023737  001352  001364          CMP     TOCYL,CYLADD    ;SEE IF TO CYL ECHOED OK
 5086  016352  001401                          BEQ     3$
 5087  016354  104045                          ERROR   45              ;MR3 NOT=RKDC
 5088
 5089  016356  004737  045766          3$:     JSR     PC,RDCYLD       ;READ CYL DIFF
 5090  016362  023737  001360  C01362          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF CORRECT
 5091  016370  001401                          BEQ     4$
 5092  016372  104046                          ERROR   46              ;CYL DIFF IN RKMR2 INCORRECT
 5093
 5094  016374                          4$:
 5095
 5096  016374  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 5097  016402  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 5098  016410  012737  000005  003322          MOV     #CLEAR,HCS1
 5099  016416  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 5100  016422  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 5101  016424  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 5102  016430  000401                          BR      67$
 5103  016432  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 5104  016434                          67$:
 5105
 5106
 5107  016434  104415                          SCOP1
 5108  016436  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 5109
 5110  016442  004737  045462                  JSR     PC,SUBCLR
 5111  016446  104024                          ERROR   24              ;CERR AFTER SCLR
 5112
 5113  016450  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
 5114  016456  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;SET RETURN CYL ADDR
 5115  016464  013737  001350  001354          MOV     FRCYL,CCYL
 5116  016472  013737  001354  003364          MOV     CCYL,TEMP3
 5117  016500  013737  001356  003366          MOV     PCYL,TEMP4
 5118  016506  163737  003364  003366          SUB     TEMP3,TEMP4
 5119  016514  100002                          BPL     5$              ;BR IF TEMP4 IS POS
 5120  016516  005437  003366                  NEG     TEMP4
 5121  016522  013737  003366  001360  5$:     MOV     TEMP4,CALDIF
 5122  016530  013737  001354  001356          MOV     CCYL,PCYL
 5123  016536  012737  000017  003322          MOV     #SEEK,HCS1
 5124  016544  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR RDY
 5125  016550  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
 5126  016552  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 5127  016556  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
 5128  016560  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
 5129  016566  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
 5130  016574  004737  046052                  JSR     PC,RDCYLA       ;READ CYL ADDR
 5131  016600  023737  001350  001364          CMP     FRCYL,CYLADD    ;SEE IF RETURN CYL ECHOED OK
 5132  016606  001401                          BEQ     6$
 5133  016610  104241                          ERROR   241             ;MR3 NOT=RKDC
 5134
 5135  016612  023737  001352  001350  6$:     CMP     TOCYL,FRCYL     ;SEE IF TO=FROM
 5136  016620  001022                          BNE     10$             ;DO NORMAL TEST IF NO
```

```
5137  016622  005737  001352              TST     TOCYL           ;SEE IF=0
5138  016626  001007                      BNE     9$
5139  016630  004737  045766              JSR     PC,RDCYLD       ;CYL DIFF S/B 0 ON CYL 0
5140  016634  005737  001362              TST     CYLDIF
5141  016640  001421                      BEQ     7$
5142  016642  104242                      ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
5143  016644  000417                      BR      7$
5144
5145  016646  004737  045766      9$:     JSR     PC,RDCYLD       ;CYL DIFF/OFFSET SHOULD NOT
5146  016652  023727  001362  000001      CMP     CYLDIF,#1       ;CHANGE IN SEEK TO SELF
5147  016660  001411                      BEQ     7$              ;SHOULD = 1 IN THIS TEST
5148  016662  104263                      ERROR   263             ;CYL DIFF IN RKMR2 DID NOT REMAIN = 0
5149  016664  000407                      BR      7$
5150  016666  004737  045766      10$:    JSR     PC,RDCYLD       ;READ CYL DIFF
5151  016672  023737  001360  001362      CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF OK
5152  016700  001401                      BEQ     7$
5153  016702  104242                      ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
5154
5155  016704              7$:
5156
5157  016704  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
5158  016712  013765  001222  000010      MOV     $UNIT,RKCS2(R5) ;DRIVE#
5159  016720  012737  000005  003322      MOV     #CLEAR,HCS1
5160  016726  004737  043472              JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5161  016732  104151                      ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5162  016734  004737  044050              JSR     PC,TSTATN       ;TEST FOR ATTN
5163  016740  000401                      BR      68$
5164  016742  104154                      ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5165  016744              68$:
5166
5167
5168  016744  005237  001352              INC     TOCYL
5169  016750  023737  001352  012102      CMP     TOCYL,LCP1      ;SEE IF SCANNED ALL CYLS
5170  016756  001402                      BEQ     8$              ;BR IF YES
5171  016760  000137  016214              JMP     1$              ;ELSE REPEAT
5172
5173  016764  005237  001350      8$:     INC     FRCYL
5174  016770  023737  001350  012102      CMP     FRCYL,LCP1      ;SEE IF ALL DONE
5175  016776  001405                      BEQ     TST17           ;;GO TO NEXT TST
5176  017000  013737  001350  001352      MOV     FRCYL,TOCYL     ;FRCYL ALWAYS = OR > TOCYL
5177  017006  000137  016214              JMP     1$              ;ELSE REPEAT
5178  017012              12$:
5179
5180
5181                                      ;;*****************************************************************
5182                                      ;*TEST 17       HEAD REGISTER TEST
5183                                      ;*
5184                                      ;*      THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
5185                                      ;*      VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
5186                                      ;*      HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.
5187                                      ;*
5188                                      ;*      SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK CMDS,
5189                                      ;*      SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
5190                                      ;*      ILLEGAL ADDRESS. IF NOT,THIS MEANS THAT CHANGING HEAD ADDRESSES
5191                                      ;*      ARE NOT TIED TO SEEK CMDS
5192                                      ;*
```

```
                                ;;*****************************************************************
017012  000004                TST17:  SCOPE
017014  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
017022  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

017026  005000                        CLR     R0              ;HEAD #
017030                          1$:
017030  104415                        SCOP1
017032  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

017036  004737  045462                JSR     PC,SUBCLR
017042  104024                        ERROR   24              ;CERR AFTER SCLR


017044  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
017052  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
017060  012737  000013  003322        MOV     #RECAL,HCS1
017066  004737  043472                JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
017072  104124                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD

017074  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
017102  004737  045132                JSR     PC,GSTAT
017106  032737  020000  003350        BIT     #D.RTZ,HMR2
017114  001001                        BNE     64$
017116  104244                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
017120  013737  001414  003362  64$:  MOV     T10,TEMP2       ;SETUP TIMEOUT
017126  004737  044102                JSR     PC,FATT1        ;FIND ATTN
017132  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD

017134  012737  050340  003412        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
017142  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
017146  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
017154  012737  000001  003420        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
017162  005037  003422                CLR     E.A2            ;EXPECTED MSG A2
017166  012737  000002  003424        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
017174  012737  000003  003430        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

017202  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
017206  000007                        .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
017210  104221                        ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
017212  104275                        ERROR   275             ;MSG B0 ERROR
017214  104222                        ERROR   222             ;MSG A1 ERROR
017216  104276                        ERROR   276             ;MSG B1 ERROR
                                ;       CWD2    47,50,<AFTER RECAL CMD>
017220  012765  000002  000026        MOV     #2,RKMR1(R5)    ;SELECT THE MESSAGE
017226  004737  045132                JSR     PC,GSTAT        ;GET THE STATUS
017232  005737  001364                TST     CYLADD          ;RECAL SUCCESSFUL ?
017236  001401                        BEQ     65$             ;BRANCH IF SO
017240  104050                        ERROR   50              ;REPORT THE ERROR
017242                          65$:

017242  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
017250  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
017256  012737  000005  003322        MOV     #CLEAR,HCS1
017264  004737  043472                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
```

```
5249  017270  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5250  017272  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
5251  017276  000401                          BR      66$
5252  017300  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5253  017302                          66$:
5254
5255  017302  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5256  017310  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
5257  017314  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
5258  017322  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5259  017330  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5260  017334  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5261  017342  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5262
5263  017350  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5264  017354  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5265  017356  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5266  017360  104265                          ERROR   265             ;MSG B0 ERROR
5267  017362  104274                          ERROR   274             ;MSG A1 ERROR
5268  017364  104266                          ERROR   266             ;MSG B1 ERROR
5269
5270
5271  017366  023727  001432  000001          CMP     HEADA,#1        ;FOR HEAD 0, B3=1
5272  017374  001401                          BEQ     3$
5273  017376  104053                          ERROR   53              ;RECAL DID NOT RESET HEAD REG IN B3.
5274
5275  017400                          3$:
5276  017400  104415                          SCOP1
5277  017402  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5278
5279  017406  004737  045462                  JSR     PC,SUBCLR
5280  017412  104024                          ERROR   24              ;CERR AFTER SCLR
5281
5282  017414  000300                          SWAB    R0
5283  017416  010065  000006                  MOV     R0,RKDA(R5)     ;HEAD #
5284  017422  000300                          SWAB    R0
5285
5286  017424  012737  000017  003322          MOV     #SEEK,HCS1
5287  017432  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR RDY
5288  017436  104156                          ERROR   156             ;NO RDY AFTER SEEK TO SELF
5289  017440  004737  044050                  JSR     PC,TSTATN
5290  017444  104157                          ERROR   157             ;NO ATTN AFTER SEEK TO SELF
5291  017446  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
5292  017454  020027  000003                  CMP     R0,#3
5293  017460  001403                          BEQ     4$              ;BR FOR HEAD 3
5294  017462  005037  003414                  CLR     E.B0            ;FOR HEADS 0,1,2
5295  017466  000403                          BR      5$
5296  017470  012737  002240  003414  4$:     MOV     #<D.SKI!D.FLT!D.IDAE>,E.B0       ;FOR HEAD 3
5297  017476  012737  001720  003416  5$:     MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5298  017504  012737  000001  003420          MOV     #1,E.B1
5299  017512  005037  003422                  CLR     E.A2
5300  017516  012737  000002  003424          MOV     #2,E.B2
5301  017524  005700                          TST     R0              ;SEE IF HEAD 0
5302  017526  001004                          BNE     6$              ;BR IF NO
5303  017530  012737  001003  003430          MOV     #<BIT9!3>,E.B3  ;LOAD EXPECTED B3 FOR HEAD 0 & MSG ID
5304  017536  000412                          BR      8$
```

```
5305  017540  020027  000001          6$:     CMP     R0,#1           ;SEE IF HEAD 1
5306  017544  001004                  BNE     7$              ;BR IF NO
5307  017546  012737  002003  003430          MOV     #<BIT10!3>,E.B3 ;B3 FOR HEAD 1
5308  017554  000403                  BR      8$
5309  017556  012737  004003  003430  7$:     MOV     #<BIT11!3>,E.B3 ;B3 FOR HEAD 2
5310  017564                          8$:
5311
5312  017564  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5313  017570  000007                  .WORD   T.A2!T.B2!T.B3                  ;& MSGS SPECIFIED HERE
5314  017572  104114                  ERROR   114             ;MSG A0 ERROR AFTER LOAD HEAD REG & SEEK CMD
5315  017574  104115                  ERROR   115             ;MSG B0 ERROR
5316  017576  104322                  ERROR   322             ;MSG A1 ERROR
5317  017600  104323                  ERROR   323             ;MSG B1 ERROR
5318
5319  017602  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
5320  017610  004737  045132                  JSR     PC,GSTAT
5321  017614  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5322  017620  001401                  BEQ     67$             ;BR IF YES
5323  017622  104324                  ERROR   324             ;MSG A2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
5324  017624  005737  001364          67$:    TST     CYLADD          ;SEE IF MSG B2=0
5325  017630  001401                  BEQ     68$             ;BR IF YES
5326  017632  104325                  ERROR   325             ;MSG B2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
5327  017634                          68$:
5328
5329  017634  020027  000003                  CMP     R0,#3
5330  017640  001412                  BEQ     9$              ;BR IF HEAD 3
5331
5332  017642  005037  003360                  CLR     TEMP1
5333  017646  116037  003312  003360          MOVB    ATTN(R0),TEMP1
5334  017654  023737  003360  001432          CMP     TEMP1,HEADA     ;FOR RKDA=HEAD 0, HEAD=1 IN B3
5335                                                                  ;FOR RKDA=HEAD 1, HEAD=2 IN B3
5336                                                                  ;FOR RKDA=HEAD 2, HEAD=4 IN B3
5337  017662  001401                  BEQ     9$
5338  017664  104054                  ERROR   54              ;HEAD DECODE IN B3 INCORRECT
5339
5340
5341  017666  005200                  9$:     INC     R0
5342  017670  020027  000004                  CMP     R0,#4           ;0 THRU 3 DONE?
5343  017674  001402                  BEQ     10$             ;BR IF YES
5344  017676  000137  017030                  JMP     1$              ;ELSE REPEAT
5345
5346  017702                          10$:
5347
5348  017702  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5349  017710  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
5350  017716  012737  000013  003322          MOV     #RECAL,HCS1
5351  017724  004737  043472                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
5352  017730  104124                  ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
5353
5354  017732  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5355  017740  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5356  017746  012737  000005  003322          MOV     #CLEAR,HCS1
5357  017754  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5358  017760  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5359  017762  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
5360  017766  000401                  BR      70$
```

```
5361  017770  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5362  017772                        70$:
5363
5364
5365  017772  004737  045132                JSR     PC,GSTAT
5366  017776  032737  000040  003352        BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
5367  020004  001401                        BEQ     69$             ;BR IF YES
5368  020006  104155                        ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
5369
5370  020010  012765  100000  000000  69$:  MOV     #CCLR,RKCS1(R5)
5371  020016  013737  001412  003362        MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
5372  020024  004737  044102                JSR     PC,FATT1
5373  020030  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD
5374
5375
5376                                ;;*****************************************************************
5377                                ;*TEST 20       SEEK TO CYL 0
5378                                ;*
5379                                ;*      TESTS THE ABILITY TO DO A SEEK CMD.
5380                                ;*      VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
5381                                ;*      STATUS BITS.  VERIFIES CMD COMPLETION BETWEEN 10-15USEC.
5382                                ;*      READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.
5383                                ;*
5384                                ;;*****************************************************************
5385  020032  000004                TST20:  SCOPE
5386  020034  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
5387  020042  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5388
5389  020046  004737  045462                JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
5390  020052  104024                        ERROR   24              ;CERR AFTER SCLR
5391  020054  004737  046052                JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
5392  020060  005737  001364                TST     CYLADD
5393  020064  001401                        BEQ     1$
5394  020066  104130                        ERROR   130             ;CYL ADDR NOT CLEARED AFTER SCLR
5395  020070                        1$:
5396  020070  104415                        SCOP1
5397  020072  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5398
5399  020076  004737  045462                JSR     PC,SUBCLR
5400  020102  104024                        ERROR   24              ;CERR AFTER SCLR
5401
5402  020104  012737  000017  003322        MOV     #SEEK,HCS1
5403  020112  004737  043472                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR RDY
5404  020116  104131                        ERROR   131             ;RDY NOT SET AFTER SEEK CMD
5405  020120  012737  000005  003360        MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
5406
5407  020126  004737  044176                JSR     PC,FATT2        ;FIND ATTN
5408  020132  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
5409  020134  032737  100000  003322        BIT     #CERR,HCS1
5410  020142  001401                        BEQ     64$
5411  020144  104210                        ERROR   210             ;CERR AFTER SEEK CMD
5412  020146                        64$:
5413
5414  020146  012737  050340  003412        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5415  020154  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
5416  020160  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
```

```
5417  020166  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5418  020174  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5419  020200  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5420  020206  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5421
5422  020214  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5423  020220  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5424  020222  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
5425  020224  104134                          ERROR   134             ;MSG B0 ERROR
5426  020226  104135                          ERROR   135             ;MSG A1 ERROR
5427  020230  104136                          ERROR   136             ;MSG B1 ERROR
5428  020232  005737  001362                  TST     CYLDIF
5429  020236  001401                          BEQ     65$
5430  020240  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
5431
5432  020242                          65$:
5433
5434  020242  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5435  020250  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5436  020256  012737  000005  003322          MOV     #CLEAR,HCS1
5437  020264  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5438  020270  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5439  020272  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
5440  020276  000401                          BR      66$
5441  020300  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5442  020302                          66$:
5443
5444  020302  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5445  020310  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
5446  020314  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
5447  020322  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5448  020330  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5449  020334  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5450  020342  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5451
5452  020350  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5453  020354  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5454  020356  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5455  020360  104265                          ERROR   265             ;MSG B0 ERROR
5456  020362  104274                          ERROR   274             ;MSG A1 ERROR
5457  020364  104266                          ERROR   266             ;MSG B1 ERROR
5458
5459  020366  005737  001364                  TST     CYLADD
5460  020372  001401                          BEQ     TST21           ;;GO TO NEXT TEST
5461  020374  104140                          ERROR   140             ;CYL ADDR IN B2 NOT CLEARED AFT SEEK CMD.
5462
5463                          ;;*************************************************************
5464                          ;*TEST 21        TEST SECTOR COUNT REG. IN MSG B3
5465                          ;;*************************************************************
5466  020376  000004          TST21:  SCOPE
5467  020400  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5468  020406  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5469
5470  020412  004737  045462                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
5471  020416  104024                          ERROR   24              ;CERR AFTER SCLR
5472  020420  012737  020516  001176          MOV     #2$,$ESCAPE     ;GO TO NEXT TEST IF ANY ERROR DETECTED
```

```
5473
5474  020426  012737  000025  001400          MOV     #21.,SECNT      ;22 SECTOR FORMAT TEST
5475
5476  020434  004737  045612                  JSR     PC,FS022                ;FIND SECTOR 0
5477  020440  104142                          ERROR   142             ;SECTOR 0 NOT FOUND BY TIMEOUT
5478
5479  020442  005037  001402                  CLR PSEC                ;PREVIOUS SECTOR
5480  020446  004737  045676          64$:    JSR     PC,FNS22                ;FIND NEXT SECTOR
5481  020452  104143                          ERROR   143             ;DIFFERENT SECTOR  NOT FOUND BY TIMEOUT
5482  020454  013737  001402  001404          MOV     PSEC,ESEC
5483  020462  062737  000001  001404          ADD     #1,ESEC         ;SETUP EXPECTED SECTOR
5484  020470  013737  001406  001402          MOV     SECTOR,PSEC     ;UPDATE PREV SECTOR
5485                                  ;       JSR     PC,RDSEC                ;READ SECTOR
5486                                  ;       CMP     SECTOR,PSEC
5487                                  ;       BEQ     65$             ;BR IF READ SAME TWICE
5488                                  ;       JSR     PC,RDSEC
5489                                  ;       CMP     SECTOR,PSEC
5490                                  ;       BEQ     65$             ;TRY 1 MORE TIME
5491                                  ;       ERROR   144             ;MSG B3 ERROR, SECTOR REG UNSTABLE
5492                                  ;                               ;MAY BE DURING SECTOR PULSE TIME
5493  020476  023737  001406  001404  65$:    CMP     SECTOR,ESEC
5494  020504  001401                          BEQ     66$
5495  020506  104145                          ERROR   145             ;MSG B3 ERROR BETWEEN SECTOR COUNTS
5496  020510  005337  001400          66$:    DEC     SECNT
5497  020514  001354                          BNE     64$             ;BR IF SECTOR COUNT NOT DONE
5498
5499
5500  020516  005037  001176          2$:     CLR     $ESCAPE
5501
5502                                  ;;**********************************************************************
5503                                  ;*TEST 22       DETECT OUTER LIMIT
5504                                  ;*
5505                                  ;*      THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYL 0
5506                                  ;*      BY DETECTING OUTER LIMIT AS THE ADJACENT CYL.
5507                                  ;*      AN ERROR IN THIS TEST INDICATES:
5508                                  ;*
5509                                  ;*              A.  HEADS WERE NOT ON CYL 0
5510                                  ;*      AND/OR  B.  COULD NOT SEEK IN REVERSE DIRECTION.
5511                                  ;*
5512                                  ;;**********************************************************************
5513  020522  000004                  TST22:  SCOPE
5514  020524  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5515  020532  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5516
5517  020536  004737  045462                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
5518  020542  104024                          ERROR   24              ;CERR AFTER SCLR
5519
5520  020544  005037  001410                  CLR     LPFLG
5521  020550  005237  001462                  INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
5522  020554  005237  003304                  INC     UNLD            ;USED FOR VALID HALT
5523
5524  020560  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
5525  020566  013765  012120  000020          MOV     FCP1,RKDC(R5)
5526  020574  012737  000017  003322          MOV     #SEEK,HCS1
5527  020602  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
5528  020606  104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
```

```
 5529  020610  004737  044050                    JSR     PC,TSTATN       ;TEST FOR ATTN
 5530  020614  104125                            ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
 5531  020616  012737  050340  003412            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
 5532  020624  012737  001200  003414            MOV     #<D.FLT!D.PAR>,E.B0
 5533  020632  012737  001720  003416            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 5534  020640  012737  000001  003420            MOV     #1,E.B1
 5535
 5536  020646  004737  044310                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 5537  020652  000000                            .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 5538  020654  104110                            ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
 5539  020656  104111                            ERROR   111             ;MSG B0 ERROR
 5540  020660  104146                            ERROR   146             ;MSG A1 ERROR
 5541  020662  104147                            ERROR   147             ;MSG B1 ERROR
 5542
 5543  020664  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
 5544  020672  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
 5545  020700  012737  000005  003322            MOV     #CLEAR,HCS1
 5546  020706  004737  043472                    JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 5547  020712  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 5548  020714  004737  044050                    JSR     PC,TSTATN       ;TEST FOR ATTN
 5549  020720  000401                            BR      65$
 5550  020722  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 5551  020724                          65$:
 5552
 5553  020724  012737  010340  003412            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
 5554  020732  005037  003414                    CLR     E.B0            ;EXPECTED MSG B0
 5555  020736  012737  001720  003416            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 5556  020744  012737  000001  003420            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 5557  020752  005037  003422                    CLR     E.A2            ;EXPECTED MSG A2
 5558  020756  012737  000002  003424            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 5559  020764  012737  000003  003430            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5560
 5561  020772  004737  044310                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 5562  020776  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5563  021000  104273                            ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 5564  021002  104265                            ERROR   265             ;MSG B0 ERROR
 5565  021004  104274                            ERROR   274             ;MSG A1 ERROR
 5566  021006  104266                            ERROR   266             ;MSG B1 ERROR
 5567
 5568
 5569  021010  013765  012116  000020            MOV     FC,RKDC(R5)
 5570  021016  012737  000017  003322            MOV     #SEEK,HCS1
 5571  021024  004737  043472                    JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 5572  021030  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD
 5573  021032  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
 5574  021040  004737  045132                    JSR     PC,GSTAT
 5575  021044  004737  046316                    JSR     PC,FLIM         ;FIND LIMIT DETECT
 5576  021050  104160                            ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
 5577
 5578  021052  032737  040000  003350            BIT     #D.UNLD,HMR2
 5579  021060  001003                            BNE     1$
 5580  021062  104305                            ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
 5581  021064  000137  021602                    JMP     30$             ;BYPASS REST OF TEST
 5582
 5583  021070  012737  021520  001176  1$:       MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
 5584  021076  012737  070140  003412            MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
```

```
5585  021104  012737  002200  003414          MOV     #<D.SKI!D.FLT>,E.B0
5586  021112  012737  045720  003416          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5587  021120  012737  030001  003420          MOV     #<D.LIMD!D.NMOV!1>,E.B1
5588                          ;               CHECK   161,162,163,164,<AFTER OUTER LIMIT DETECT>,0,0,0
5589  021126  004737  044310                  JSR     PC,CHKMSG       ;CHECK MESSAGE A0,B0,A1,B1
5590  021132  000000                          .WORD   0!0!0
5591  021134  104161                          ERROR   161             ;MAY BE A0 ERROR
5592  021136  104162                          ERROR   162             ;MAY BE B0 ERROR
5593  021140  104163                          ERROR   163             ;MAY BE B1 ERROR
5594  021142  000240                          NOP                     ;NEED FOR THE CALLIN SEQ
5595  021144  032737  020000  003400          BIT     #D.LIMD,H.B1    ;SEE IF LIMIT DETECT BIT SET IN B1
5596  021152  001003                          BNE     3$              ;BRANCH IF SO
5597  021154  104401  061436                  TYPE    ,EM42           ;OTHERWISE REPORT ERROR
5598  021160  104164                          ERROR   164
5599
5600  021162  004737  044050          3$:     JSR     PC,TSTATN
5601  021166  104165                          ERROR   165             ;NO ATTN AFTER OUTER LIMIT DETECT
5602  021170  005037  001462                  CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1
5603
5604  021174  004737  045462                  JSR     PC,SUBCLR       ;SUBSYS CLR
5605  021200  104024                          ERROR   24              ;CERR AFTER SCLR
5606  021202  013737  001414  003362          MOV     T10,TEMP2       ;SET UP TIMEOUT
5607  021210  004737  046374                  JSR     PC,FHDHM        ;FIND HEAD HOME
5608  021214  104166                          ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
5609  021216  004737  046450                  JSR     PC,FLOAD        ;FIND LOAD HEADS
5610  021222  104167                          ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
5611  021224  013737  001416  003362          MOV     T100,TEMP2      ;SETUP TIMEOUT
5612  021232  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
5613  021236  104067                          ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
5614  021240  005037  001176          2$:     CLR     $ESCAPE
5615  021244  005037  003304                  CLR     UNLD            ;CLEAR FLAG
5616
5617  021250  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5618  021256  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
5619  021262  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5620  021270  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5621  021276  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5622  021302  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5623  021310  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5624
5625  021316  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5626  021322  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5627  021324  104063                          ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOADING
5628  021326  104064                          ERROR   64              ;MSG B0 ERROR
5629  021330  104065                          ERROR   65              ;MSG A1 ERROR
5630  021332  104066                          ERROR   66              ;MSG B1 ERROR
5631                          ;               CWD2    175,176,<AT END OF HEAD LOADING>
5632  021334  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT MESSAGE
5633  021342  004737  045132                  JSR     PC,GSTAT        ;GET STATUS AND MR2,MR3
5634  021346  005737  001364                  TST     CYLADD          ;RECAL SUCCFUL ?
5635  021352  001401                          BEQ     64$             ;BRANCH IF SO
5636  021354  104050                          ERROR   50              ;REPORT ERROR
5637  021356                          64$:
5638
5639  021356  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5640  021364  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
```

```
5641  021372  012737  000005  003322          MOV     #CLEAR,HCS1
5642  021400  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5643  021404  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5644  021406  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
5645  021412  000401                          BR      66$
5646  021414  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5647  021416                          66$:
5648
5649  021416  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5650  021424  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
5651  021430  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
5652  021436  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5653  021444  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5654  021450  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5655  021456  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5656
5657  021464  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5658  021470  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5659  021472  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5660  021474  104265                          ERROR   265             ;MSG B0 ERROR
5661  021476  104274                          ERROR   274             ;MSG A1 ERROR
5662  021500  104266                          ERROR   266             ;MSG B1 ERROR
5663
5664  021502  004737  047466                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5665  021506  000435                          BR      TST23           ;;GO TO NEXT TEST
5666                                                                  ;RETURN HERE IF SW 14 IS SET OR
5667                                                                  ;SW 8 WITH SWR <7:0> APPLY
5668
5669
5670  021510  005037  001176          10$:    CLR     $ESCAPE
5671  021514  000177  157366                  JMP     @$LPADR
5672  021520                          20$:
5673
5674  021520  004737  045462                  JSR     PC,SUBCLR
5675  021524  104024                          ERROR   24              ;CERR AFTER SCLR
5676
5677  021526  012737  000011  003322          MOV     #SRTSPL,HCS1
5678  021534  004737  043472                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
5679  021540  104121                          ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
5680
5681  021542  013737  001420  003362          MOV     T500,TEMP2      ;SETUP  TIMEOUT
5682  021550  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
5683  021554  104067                          ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
5684
5685  021556  005037  003304                  CLR     UNLD
5686  021562  005237  001410                  INC     LPFLG
5687  021566  032777  001000  157344          BIT     #SW9,@SWR       ;LOOP ON ERROR?
5688  021574  001345                          BNE     10$             ;YES, RECONDITION DRIVE
5689  021576  000137  021240                  JMP     2$              ;RETURN TO MAINLINE
5690  021602                          30$:
5691
5692
5693
5694
5695
5696                                    ;;*****************************************************************
```

```
                                        ;*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
                                        ;*
                                        ;;*****************************************************************
021602  000004                  TST23:  SCOPE
021604  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
021612  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
021616  005737  001460                  TST     LIMERR          ;CHK FOR LIMIT ERROR
021622  001403                          BEQ     5$              ;BR IF NO
021624  104170                          ERROR   170             ;FATAL ERROR
021626  000137  042706                  JMP     $EOP            ;ABORT BAL OF TESTS
021632                          5$:
021632  005237  003306                  INC     BADHDR          ;USED FOR VALID HALT
021636  005037  001430                  CLR     HEAD            ;HEAD CTR
021642  104415                          SCOP1
021644  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
021650  004737  045462                  JSR     PC,SUBCLR
021654  104024                          ERROR   24              ;CERR AFTER SCLR
021656  052765  000020  000010  1$:     BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
021664  012765  001470  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
021672  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT.
021700  000337  001430                  SWAB    HEAD
021704  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
021712  000337  001430                  SWAB    HEAD
021716  013700  001430                  MOV     HEAD,R0
021722  006300                          ASL     R0              ;DOUBLE R0
021724  016037  001444  001470          MOV     DATA0(R0),HDTAB ;SETUP HEADER WORD FOR RKBA
021732  012737  000027  003322          MOV     #<WRHEAD>,HCS1
021740  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
021744  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
021746  004737  045132                  JSR     PC,GSTAT        ;GET FRESH STATUS
021752  032737  100000  003322          BIT     #CERR,HCS1
021760  001405                          BEQ     64$
021762  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
021764  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
021770  000137  042706                  JMP     $EOP            ;ABORT DRIVE
021774                          64$:
022002  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
022006  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
022014  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
022022  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
022030  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
```

```
 5753  022026  012737  000002  003424          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 5754  022034  012737  000003  003430          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 5755
 5756  022042  004737  044310                  JSR     PC,CHKMSG        ;CHECK MSGS A0,B0,A1,B1
 5757  022046  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
 5758  022050  104277                          ERROR   277              ;MSG A0 ERROR AFTER WRITE HEADER CMD
 5759  022052  104267                          ERROR   267              ;MSG B0 ERROR
 5760  022054  104300                          ERROR   300              ;MSG A1 ERROR
 5761  022056  104270                          ERROR   270              ;MSG B1 ERROR
 5762
 5763
 5764  022060  012765  000002  000026          MOV     #2,RKMR1(R5)     ;SELECT WORD 2
 5765  022066  004737  045132                  JSR     PC,GSTAT
 5766  022072  005737  001362                  TST     CYLDIF           ;SEE IF MSG A2=0
 5767  022076  001401                          BEQ     65$              ;BR IF YES
 5768  022100  104303                          ERROR   303              ;MSG A2 NOT CLEARED AFTER WRITE HEADER CMD
 5769  022102  005737  001364          65$:    TST     CYLADD           ;SEE IF MSG B2=0
 5770  022106  001401                          BEQ     66$              ;BR IF YES
 5771  022110  104304                          ERROR   304              ;MSG B2 NOT CLEARED AFTER WRITE HEADER CMD
 5772  022112                          66$:
 5773
 5774  022112  005237  001430                  INC     HEAD
 5775  022116  023727  001430  000003          CMP     HEAD,#3
 5776  022124  001254                          BNE     1$
 5777
 5778  022126  005037  001430                  CLR     HEAD             ;HEAD CTR
 5779  022132  104415                          SCOP1
 5780  022134  012706  001100                  MOV     #STACK,SP        ;RESTORE STK PTR
 5781
 5782  022140  004737  045462                  JSR     PC,SUBCLR
 5783  022144  104024                          ERROR   24               ;CERR AFTER SCLR
 5784
 5785
 5786
 5787  022146  000337  001430          2$:     SWAB    HEAD
 5788  022152  013765  001430  000006          MOV     HEAD,RKDA(R5)    ;SETUP HEAD ADDR
 5789  022160  000337  001430                  SWAB    HEAD
 5790
 5791  022164  012700  001674                  MOV     #RHTAB,R0
 5792  022170  012737  000025  003322          MOV     #<RDHEAD>,HCS1
 5793  022176  004737  043530                  JSR     PC,DATCMD        ;DO DATA XFER CMD & GET CONTR RDY
 5794  022202  104171                          ERROR   171              ;NO RDY AFTER READ HEADER CMD
 5795  022204  032737  100000  003322          BIT     #CERR,HCS1
 5796  022212  001405                          BEQ     67$
 5797  022214  104174                          ERROR   174              ;CERR AFTER READ HEADER CMD
 5798  022216  104401  056436                  TYPE    ,MSG18           ;ABORT BALANCE OF TESTS
 5799  022222  000137  042706                  JMP     $EOP             ;ABORT DRIVE
 5800
 5801  022226  016520  000024          67$:    MOV     RKDB(R5),(R0)+   ;1'ST WORD FROM SILO TO RHTAB
 5802  022232  016520  000024                  MOV     RKDB(R5),(R0)+   ;2'ND WORD
 5803  022236  016520  000024                  MOV     RKDB(R5),(R0)+   ;3'RD WORD
 5804
 5805
 5806  022242  032765  100000  000010          BIT     #DLT,RKCS2(R5)
 5807  022250  001407                          BEQ     68$
 5808  022252  004737  045132                  JSR     PC,GSTAT
```

```
5809  022256  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
5810  022260  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5811  022264  000137  042706                  JMP     $EOP            ;ABORT DRIVE
5812  022270                          68$:
5813
5814  022270  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
5815  022276  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
5816  022302  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
5817  022310  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5818  022316  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
5819  022322  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5820  022330  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5821
5822  022336  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5823  022342  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5824  022344  104301                          ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
5825  022346  104271                          ERROR   271             ;MSG B0 ERROR
5826  022350  104302                          ERRCR   302             ;MSG A1 ERROR
5827  022352  104272                          ERROR   272             ;MSG B1 ERROR
5828
5829
5830  022354  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
5831  022362  004737  045132                  JSR     PC,GSTAT
5832  022366  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5833  022372  001401                          BEQ     69$             ;BR IF YES
5834  022374  104172                          ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5835  022376  005737  001364          69$:    TST     CYLADD          ;SEE IF MSG B2=0
5836  022402  001401                          BEQ     70$             ;BR IF YES
5837  022404  104264                          ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5838  022406                          70$:
5839  022406  000337  001430                  SWAB    HEAD
5840  022412  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;RESTORE RKDA
5841  022420  000337  001430                  SWAB    HEAD
5842
5843  022424  012701  001674                  MOV     #RHTAB,R1
5844
5845  022430  005037  001442                  CLR     WDCNT           ;HEADER WORD COUNT
5846  022434  013700  001430                  MOV     HEAD,R0
5847  022440  006300                          ASL     R0              ;DOUBLE R0
5848  022442  016037  001444  003360          MOV     DATA0(R0),TEMP1 ;GET THE 'SHOULD BE' DATA
5849  022450  012137  001454          3$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
5850  022454  023737  001454  003360          CMP     HDWD,TEMP1
5851  022462  001401                          BEQ     4$
5852  022464  104202                          ERROR   202             ;READ HEADER MISMATCH
5853  022466  005237  001442          4$:     INC     WDCNT
5854  022472  023727  001442  000003          CMP     WDCNT,#3        ;DO ONLY 1 SECTOR
5855  022500  001363                          BNE     3$
5856
5857  022502  005237  001430                  INC     HEAD
5858  022506  023727  001430  000003          CMP     HEAD,#3         ;ALL 3 HEADS DONE?
5859  022514  001402                          BEQ     TST24           ;;GO TO NXT TST IF YES
5860  022516  000137  022146                  JMP     2$              ;ELSE REPEAT
5861
5862                          ;;*****************************************************************
5863                          ;*TEST 24       BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS
5864                          ;*
```

```
                        ;*      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.
                        ;*
                        ;;*****************************************************************
022522  000004         TST24:  SCOPE
022524  012737  000001  001174         MOV     #1,$TIMES       ;;DO 1 ITERATION
022532  012706  001100                 MOV     #STACK,SP       ;RESTORE STK PTR

022536  004737  045462                 JSR     PC,SUBCLR
022542  104024                         ERROR   24              ;CERR AFTER SCLR
022544  052765  000020  000010         BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
022552  012765  001450  000004         MOV     #DATA1,RKBA(R5) ;XFER 1'S ONLY
022560  012765  177704  000002         MOV     #-50.,RKWC(R5)  ;WORD COUNT

022566  012737  010027  003322         MOV     #<CFMT!WRHEAD>,HCS1
022574  004737  043530                 JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
022600  104200                         ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
022602  004737  045132                 JSR     PC,GSTAT        ;GET FRESH STATUS
022606  032737  100000  003322         BIT     #CERR,HCS1
022614  001405                         BEQ     64$
022616  104201                         ERROR   201             ;CERR AFTER WRITE HEADER CMD
022620  104401  056436                 TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
022624  000137  042706                 JMP     $EOP            ;ABORT DRIVE
022630                         64$:

022630  012737  010001  003322         MOV     #<CFMT!SELDRV>,HCS1     ;GET 20 SECTOR STATUS
022636  004737  043472                 JSR     PC,DOCMD        ;DO COMMAND
022642  104117                         ERROR   117             ;NO RDY AFTER SELDRV CMD
022644  032737  001000  003350         BIT     #D.FORM,HMR2
022652  001001                         BNE     1$
022654  104312                         ERROR   312             ;FORMAT NOT SET AFTER WRITE HDR CMD

022656                         1$:

022656  012737  010340  003412         MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
022664  005037  003414                 CLR     E.B0            ;EXPECTED MSG B0
022670  012737  001720  003416         MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
022676  012737  000001  003420         MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
022704  005037  003422                 CLR     E.A2            ;EXPECTED MSG A2
022710  012737  000002  003424         MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
022716  012737  000003  003430         MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

022724  004737  044310                 JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
022730  000000                         .WORD   0!0!0           ;& MSGS SPECIFIED HERE
022732  104277                         ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
022734  104267                         ERROR   267             ;MSG B0 ERROR
022736  104300                         ERROR   300             ;MSG A1 ERROR
022740  104270                         ERROR   270             ;MSG B1 ERROR

022742  005037  001400                 CLR     SECNT           ;SECTOR COUNT
022746                         2$:
022746  104415                         SCOP1
022750  012706  001100                 MOV     #STACK,SP       ;RESTORE STK PTR
```

```
5921
5922  022754  004737  045462                  JSR     PC,SUBCLR
5923  022760  104024                          ERROR   24                  ;CERR AFTER SCLR
5924
5925
5926  022762  012700  001674                  MOV     #RHTAB,R0
5927  022766  012737  010025  003322          MOV     #<CFMT!RDHEAD>,HCS1
5928  022774  004737  043530                  JSR     PC,DATCMD           ;DO DATA XFER CMD & GET CONTR RDY
5929  023000  104171                          ERROR   171                 ;NO RDY AFTER READ HEADER CMD
5930  023002  032737  100000  003322          BIT     #CERR,HCS1
5931  023010  001405                          BEQ     65$
5932  023012  104174                          ERROR   174                 ;CERR AFTER READ HEADER CMD
5933  023014  104401  056436                  TYPE    ,MSG18              ;ABORT BALANCE OF TESTS
5934  023020  000137  042706                  JMP     $EOP                ;ABORT DRIVE
5935
5936  023024  016520  000024          65$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
5937  023030  016520  000024                  MOV     RKDB(R5),(R0)+      ;2'ND WORD
5938  023034  016520  000024                  MOV     RKDB(R5),(R0)+      ;3'RD WORD
5939
5940
5941  023040  032765  100000  000010          BIT     #DLT,RKCS2(R5)
5942  023046  001407                          BEQ     66$
5943  023050  004737  045132                  JSR     PC,GSTAT
5944  023054  104173                          ERROR   173                 ;DLT AFTER READ HEADER CMD
5945  023056  104401  056436                  TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
5946  023062  000137  042706                  JMP     $EOP                ;ABORT DRIVE
5947  023066                          66$:
5948
5949  023066  012737  010001  003322          MOV     #<CFMT!SELDRV>,HCS1
5950  023074  004737  043472                  JSR     PC,DOCMD            ;DO CMD.
5951  023100  104117                          ERROR   117                 ;NO RDY AFTER SELDRV CMD
5952  023102  032737  001000  003350          BIT     #D.FORM,HMR2
5953  023110  001001                          BNE     6$
5954  023112  104313                          ERROR   313                 ;FORMAT NOT SET AFTER READ HDR CMD
5955
5956  023114                          6$:
5957
5958  023114  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5959  023122  005037  003414                  CLR     E.B0                ;EXPECTED MSG B0
5960  023126  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5961  023134  012737  000001  003420          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5962  023142  005037  003422                  CLR     E.A2                ;EXPECTED MSG A2
5963  023146  012737  000002  003424          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5964  023154  012737  000003  003430          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5965
5966  023162  004737  044310                  JSR     PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
5967  023166  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
5968  023170  104301                          ERROR   301                 ;MSG A0 ERROR AFTER READ HEADER CMD
5969  023172  104271                          ERROR   271                 ;MSG B0 ERROR
5970  023174  104302                          ERROR   302                 ;MSG A1 ERROR
5971  023176  104272                          ERROR   272                 ;MSG B1 ERROR
5972
5973  023200  012765  000002  000026          MOV     #2,RKMR1(R5)        ;SELECT WORD 2
5974  023206  004737  045132                  JSR     PC,GSTAT
5975  023212  005737  001362                  TST     CYLDIF              ;SEE IF MSG A2=0
5976  023216  001401                          BEQ     67$                 ;BR IF YES
```

```
 5977  023220  104172                    ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
 5978  023222  005737  001364    67$:    TST     CYLADD          ;SEE IF MSG B2=0
 5979  023226  001401                    BEQ     68$             ;BR IF YES
 5980  023230  104264                    ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
 5981  023232                    68$:
 5982  023232  012701  001674            MOV     #RHTAB,R1
 5983
 5984  023236  005037  001442    3$:     CLR     WDCNT           ;HEADER WORD COUNT
 5985  023242  013737  001450  003360    MOV     DATA1,TEMP1     ;GET 'SHOULD BE' DATA
 5986  023250  012137  001454    4$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
 5987  023254  023737  001454  003360    CMP     HDWD,TEMP1      ;MATCH OK?
 5988  023262  001401                    BEQ     5$              ;BR IF YES
 5989  023264  104202                    ERROR   202             ;READ HEADER MISMATCH
 5990  023266  005237  001442    5$:     INC     WDCNT
 5991  023272  023727  001442  000003    CMP     WDCNT,#3        ;JUST 1 SECTOR AND 1 HEAD
 5992  023300  001363                    BNE     4$
 5993
 5994
 5995
 5996
 5997                            ;;*****************************************************************
 5998                            ;*TEST 25       WRITE & READ HEADERS CYL 0, HEAD 0
 5999                            ;;*****************************************************************
 6000  023302  000004    TST25:  SCOPE
 6001  023304  012737  000001  001174    MOV     #1,$TIMES       ;;DO 1 ITERATION
 6002  023312  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
 6003
 6004  023316  004737  045462            JSR     PC,SUBCLR
 6005  023322  104024                    ERROR   24              ;CERR AFTER SCLR
 6006
 6007  023324  005237  001464            INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
 6008
 6009  023330  012765  001470  000004    MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
 6010  023336  012765  177676  000002    MOV     #-66.,RKWC(R5)  ;WORD COUNT.
 6011  023344  012737  000000  001352    MOV     #0,TOCYL
 6012
 6013  023352  013737  001352  001366    MOV     TOCYL,CALADD    ;SETUP
 6014  023360  012737  000000  001430    MOV     #0,HEAD         ;TO FILL
 6015  023366  012737  000000  001436    MOV     #0,FORMAT       ;HEADER
 6016  023374  004737  046572            JSR     PC,FHDTAB       ;TABLE
 6017
 6018  023400  012765  000000  000020    MOV     #0,RKDC(R5)     ;CYL#
 6019
 6020  023406  012737  000027  003322    MOV     #<WRHEAD>,HCS1
 6021  023414  004737  043530            JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
 6022  023420  104200                    ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
 6023  023422  004737  045132            JSR     PC,GSTAT        ;GET FRESH STATUS
 6024  023426  032737  100000  003322    BIT     #CERR,HCS1
 6025  023434  001405                    BEQ     64$
 6026  023436  104201                    ERROR   201             ;CERR AFTER WRITE HEADER CMD
 6027  023440  104401  056436            TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 6028  023444  000137  042706            JMP     $EOP            ;ABORT DRIVE
 6029  023450                    64$:
 6030
 6031  023450  012737  010340  003412    MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
 6032  023456  005037  003414            CLR     E.B0            ;EXPECTED MSG B0
```

```
 6033  023462  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6034  023470  012737  000001  003420          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
 6035  023476  005037  003422                  CLR     E.A2                ;EXPECTED MSG A2
 6036  023502  012737  000002  003424          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
 6037  023510  012737  000003  003430          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
 6038
 6039  023516  004737  044310                  JSR     PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
 6040  023522  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
 6041  023524  104277                          ERROR   277                 ;MSG A0 ERROR AFTER WRITE HEADER CMD
 6042  023526  104267                          ERROR   267                 ;MSG B0 ERROR
 6043  023530  104300                          ERROR   300                 ;MSG A1 ERROR
 6044  023532  104270                          ERROR   270                 ;MSG B1 ERROR
 6045
 6046  023534  005037  001400                  CLR     SECNT               ;SECTOR COUNT
 6047  023540  104415                          SCOP1
 6048  023542  012706  001100                  MOV     #STACK,SP           ;RESTORE STK PTR
 6049
 6050  023546  004737  045462                  JSR     PC,SUBCLR
 6051  023552  104024                          ERROR   24                  ;CERR AFTER SCLR
 6052
 6053  023554  012765  000000  000020          MOV     #0,RKDC(R5)         ;CYL #
 6054
 6055  023562  012700  001674                  MOV     #RHTAB,R0
 6056
 6057  023566  012737  000025  003322  65$:     MOV     #RDHEAD,HCS1
 6058  023574  004737  043530                  JSR     PC,DATCMD           ;DO READ HEADER CMD & GET CONTR RDY
 6059  023600  104171                          ERROR   171                 ;NO RDY AFTER READ HEADER CMD
 6060  023602  032737  100000  003322          BIT     #CERR,HCS1
 6061  023610  001405                          BEQ     66$
 6062  023612  104174                          ERROR   174                 ;CERR AFTER READ HEADER  CMD
 6063  023614  104401  056436                  TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
 6064  023620  000137  042706                  JMP     $EOP                ;ABORT DRIVE
 6065
 6066  023624  016520  000024          66$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
 6067  023630  016520  000024                  MOV     RKDB(R5),(R0)+      ;2'ND WORD
 6068  023634  016520  000024                  MOV     RKDB(R5),(R0)+      ;3'RD WORD
 6069
 6070  023640  032765  100000  000010          BIT     #DLT,RKCS2(R5)      ;SEE IF DATA LATE
 6071  023646  001407                          BEQ     67$
 6072  023650  004737  045132                  JSR     PC,GSTAT
 6073  023654  104173                          ERROR   173                 ;DATA LATE ON READ HEADER
 6074  023656  104401  056436                  TYPE    ,MSG18              ;ABORT BALANCE OF TESTS
 6075  023662  000137  042706                  JMP     $EOP                ;ABORT DRIVE
 6076
 6077  023666  020027  002100          67$:    CMP     R0,#RHTAB+132.      ;ALL 66 WORDS DONE?
 6078  023672  001335                          BNE     65$                 ;BR IF NO
 6079
 6080  023674  004737  047114                  JSR     PC,SORT             ;SORT RHTAB INTO SRTTAB SO THAT IT
 6081                                                                      ;BEGINS WITH SECTOR 0
 6082  023700  005037  001442                  CLR     WDCNT               ;WORD COUNT
 6083  023704  012700  002100                  MOV     #SRTTAB,R0          ;ACTUAL HEADER TABLE
 6084  023710  012701  001470                  MOV     #HDTAB,R1           ;CALC HEADER TABLE
 6085
 6086  023714  012037  001454          68$:    MOV     (R0)+,HDWD
 6087  023720  012137  003360                  MOV     (R1)+,TEMP1
 6088  023724  023737  001454  003360          CMP     HDWD,TEMP1          ;COMPARE ACTUAL WITH CALCULATED WORD
```

```
 6089  023732  001401                        BEQ     69$             ;BR IF COMPARE
 6090  023734  104202                        ERROR   202             ;READ HEADER MISMATCH
 6091
 6092  023736  005237  001442          69$:  INC     WDCNT
 6093  023742  023727  001442  000102      CMP     WDCNT,#66.      ;ALL WORDS DONE?
 6094  023750  001361                        BNE     68$             ;BR IF NO
 6095
 6096
 6097  023752  005037  001464              CLR BYPFMT              ;ALLOW CORRECT FORMATTING
 6098
 6099
 6100
 6101
 6102                                  ;;*****************************************************************
 6103                                  ;*TEST 26       SEEK FROM CYL 0 TO 1 & READ HEADERS
 6104                                  ;*
 6105                                  ;*      THIS TEST CHECKS MSG A & B WORDS 0,1,2 FOF CORRECT STATUS AFTER RDY
 6106                                  ;*      IS RECEIVED FROM A SEEK CMD TO DETERMINE
 6107                                  ;*      THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1.
 6108                                  ;*      AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS.
 6109                                  ;*      CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR
 6110                                  ;*      IN MSG B2 IS VERIFIED TO BE 1.
 6111                                  ;*
 6112                                  ;*      HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE
 6113                                  ;*      DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.
 6114                                  ;*
 6115                                  ;;*****************************************************************
 6116  023756  000004                  TST26:  SCOPE
 6117  023760  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 6118  023766  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 6119  023772  004737  045462                  JSR     PC,SUBCLR
 6120  023776  104024                          ERROR   24              ;CERR AFTER SCLR
 6121  024000  005037  001350                  CLR     FRCYL
 6122  024004  012737  000001  001352          MOV     #1,TOCYL
 6123  024012  012737  000001  001360          MOV     #1,CALDIF
 6124  024020  012765  000001  000020          MOV     #1,RKDC(R5)     ;SET FOR CYL 1
 6125  024026  012737  025012  001176          MOV     #10$,$ESCAPE
 6126
 6127  024034  012737  000017  003322          MOV     #SEEK,HCS1
 6128  024042  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 6129  024046  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 6130  024050  012737  030140  003412          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 6131  024056  005037  003414                  CLR     E.B0
 6132  024062  012737  003720  003416          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 6133  024070  012737  000001  003420          MOV     #1,E.B1
 6134
 6135  024076  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6136  024102  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6137  024104  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 6138  024106  104204                          ERROR   204             ;MSG B0 ERROR
 6139  024110  104205                          ERROR   205             ;MSG A1 ERROR
 6140  024112  104206                          ERROR   206             ;MSG B1 ERROR
 6141
 6142  024114  023727  001362  000001          CMP     CYLDIF,#1
 6143  024122  001401                          BEQ     1$
 6144  024124  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK CMD.
```

```
6145
6146
6147  024126  012737  025032  001176  1$:     MOV     #12$,$ESCAPE
6148  024134  013737  001422  003360          MOV     T2500,TEMP1     ;SETUP TIMEOUT
6149
6150
6151  024142  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
6152  024146  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6153  024150  032737  100000  003322          BIT     #CERR,HCS1
6154  024156  001401                          BEQ     64$
6155  024160  104210                          ERROR   210             ;CERR AFTER SEEK CMD
6156  024162                          64$:
6157
6158  024162  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
6159  024170  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
6160  024174  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6161  024202  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6162  024210  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
6163  024214  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6164  024222  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6165
6166  024230  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6167  024234  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6168  024236  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6169  024240  104134                          ERROR   134             ;MSG B0 ERROR
6170  024242  104135                          ERROR   135             ;MSG A1 ERROR
6171  024244  104136                          ERROR   136             ;MSG B1 ERROR
6172  024246  005737  001362                  TST     CYLDIF
6173  024252  001401                          BEQ     65$
6174  024254  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6175
6176  024256                          65$:
6177
6178  024256  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6179  024264  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6180  024272  012737  000005  003322          MOV     #CLEAR,HCS1
6181  024300  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
6182  024304  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6183  024306  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
6184  024312  000401                          BR      66$
6185  024314  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6186  024316                          66$:
6187
6188  024316  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
6189  024324  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
6190  024330  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6191  024336  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6192  024344  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
6193  024350  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6194  024356  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6195
6196  024364  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6197  024370  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6198  024372  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6199  024374  104265                          ERROR   265             ;MSG B0 ERROR
6200  024376  104274                          ERROR   274             ;MSG A1 ERROR
```

```
6201  024400  104266                          ERROR   266             ;MSG B1 ERROR
6202
6203  024402  005737  001364                  TST     CYLADD
6204  024406  023727  001364  000001          CMP     CYLADD,#1
6205  024414  001401                          BEQ     2$
6206  024416  104207                          ERROR   207             ;CYL ADDR INCORRECT AFTER SEEK CMD
6207
6208
6209  024420                          2$:
6210  024420  104415                          SCOP1
6211  024422  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6212
6213  024426  004737  045462                  JSR     PC,SUBCLR
6214  024432  104024                          ERROR   24              ;CERR AFTER SCLR
6215
6216  024434  005037  001176                  CLR     $ESCAPE
6217  024440  012765  000001  000020          MOV     #1,RKDC(R5)     ;CYL #
6218
6219  024446  012700  001674                  MOV     #RHTAB,R0
6220  024452  012737  000025  003322          MOV     #<RDHEAD>,HCS1
6221  024460  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
6222  024464  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
6223  024466  032737  100000  003322          BIT     #CERR,HCS1
6224  024474  001405                          BEQ     67$
6225  024476  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
6226  024500  104401  056436                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
6227  024504  000137  042706                  JMP     $EOP            ;ABORT DRIVE
6228
6229  024510  016520  000024          67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6230  024514  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
6231  024520  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
6232
6233
6234  024524  032765  100000  000010          BIT     #DLT,RKCS2(R5)
6235  024532  001407                          BEQ     68$
6236  024534  004737  045132                  JSR     PC,GSTAT
6237  024540  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
6238  024542  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6239  024546  000137  042706                  JMP     $EOP            ;ABORT DRIVE
6240  024552                          68$:
6241
6242
6243  024552  005737  001674                  TST     RHTAB           ;CHECK 1'ST WORD ONLY:CYL#
6244  024556  001001                          BNE     3$
6245  024560  104211                          ERROR   211             ;CYL 0 HEADER ON CYL 1
6246
6247  024562  013737  001674  001454  3$:     MOV     RHTAB,HDWD
6248  024570  012737  000001  003360          MOV     #1,TEMP1
6249  024576  023737  001454  003360          CMP     HDWD,TEMP1
6250  024604  001401                          BEQ     4$
6251  024606  104202                          ERROR   202             ;READ CYL WORD HEADER ERROR
6252  024610                          4$:
6253  024610  004737  047466                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6254  024614  000516                          BR      TST27           ;;GO TO NEXT TEST
6255                                                                  ;RETURN HERE IF SW 14 IS SET OR
6256                                                                  ;SW 8 WITH SWR <7:0> APPLY
```

```
 6257  024616  004737  045462          6$:     JSR     PC,SUBCLR
 6258  024622  104024                          ERROR   24              ;CERR AFTER SCLR
 6259
 6260  024624  012737  000017  003322          MOV     #SEEK,HCS1
 6261  024632  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 6262  024636  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
 6263
 6264  024640  013737  001426  003360          MOV     T50000,TEMP1
 6265  024646  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 6266  024652  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 6267  024654  032737  100000  003322          BIT     #CERR,HCS1
 6268  024662  001401                          BEQ     69$
 6269  024664  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
 6270
 6271  024666  004737  045462          69$:    JSR     PC,SUBCLR
 6272  024672  104024                          ERROR   24              ;CERR AFTER SCLR
 6273
 6274
 6275  024674  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 6276  024702  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 6277  024706  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6278  024714  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6279  024722  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 6280  024726  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6281  024734  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6282
 6283  024742  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6284  024746  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6285  024750  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 6286  024752  104134                          ERROR   134             ;MSG B0 ERROR
 6287  024754  104135                          ERROR   135             ;MSG A1 ERROR
 6288  024756  104136                          ERROR   136             ;MSG B1 ERROR
 6289  024760  005737  001364                  TST     CYLADD
 6290  024764  001401                          BEQ     7$
 6291  024766  104043                          ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
 6292
 6293  024770                          7$:
 6294  024770  005037  001176                  CLR     $ESCAPE
 6295  024774  005737  001410                  TST     LPFLG
 6296  025000  001402                          BEQ     70$
 6297  025002  000177  154102                  JMP     @$LPERR         ;SW 9 WAS SET.
 6298  025006  000177  154074          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 6299
 6300  025012                          10$:
 6301  025012  005237  001410                  INC     LPFLG
 6302  025016  032777  001000  154114          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 6303  025024  001274                          BNE     6$              ;YES, RECONDITION DRIVE
 6304  025026  000137  024126                  JMP     1$              ;RETURN TO MAINLINE
 6305  025032                          12$:
 6306  025032  005237  001410                  INC     LPFLG
 6307  025036  032777  001000  154074          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 6308  025044  001264                          BNE     6$              ;YES, RECONDITION DRIVE
 6309  025046  000137  024420                  JMP     2$              ;RETURN TO MAINLINE
 6310
 6311
 6312
```

```
                                        ;;*****************************************************************
                                        ;*TEST 27       WRITE & READ HEADERS CYL 1, HEAD 0
                                        ;;*****************************************************************
                                        TST27:  SCOPE
025054  012737  000001  001174                  MOV     #1,$TIMES       ;;DO 1 ITERATION
025062  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR

025066  004737  045462                          JSR     PC,SUBCLR
025072  104024                                 ERROR   24              ;CERR AFTER SCLR

025074  005237  001464                          INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER

025100  012765  001470  000004                  MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
025106  012765  177676  000002                  MOV     #-66.,RKWC(R5)  ;WORD COUNT.
025114  012737  000001  001352                  MOV     #1,TOCYL

025122  013737  001352  001366                  MOV     TOCYL,CALADD    ;SETUP
025130  012737  000000  001430                  MOV     #0,HEAD         ;TO FILL
025136  012737  000000  001436                  MOV     #0,FORMAT       ;HEADER
025144  004737  046572                          JSR     PC,FHDTAB       ;TABLE

025150  012765  000001  000020                  MOV     #1,RKDC(R5)     ;CYL#

025156  012737  000027  003322                  MOV     #<WRHEAD>,HCS1
025164  004737  043530                          JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
025170  104200                                 ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
025172  004737  045132                          JSR     PC,GSTAT        ;GET FRESH STATUS
025176  032737  100000  003322                  BIT     #CERR,HCS1
025204  001405                                  BEQ     64$
025206  104201                                 ERROR   201             ;CERR AFTER WRITE HEADER CMD
025210  104401  056436                         TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
025214  000137  042706                         JMP     $EOP            ;ABORT DRIVE
025220                                  64$:

025220  012737  010340  003412                  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
025226  005037  003414                          CLR     E.B0            ;EXPECTED MSG B0
025232  012737  001720  003416                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
025240  012737  000001  003420                  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
025246  005037  003422                          CLR     E.A2            ;EXPECTED MSG A2
025252  012737  000002  003424                  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
025260  012737  000003  003430                  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

025266  004737  044310                          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
025272  000003                                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
025274  104277                                 ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
025276  104267                                 ERROR   267             ;MSG B0 ERROR
025300  104300                                 ERROR   300             ;MSG A1 ERROR
025302  104270                                 ERROR   270             ;MSG B1 ERROR

025304  005037  001400                          CLR     SECNT           ;SECTOR COUNT
025310  104415                                  SCOP1
025312  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR

025316  004737  045462                          JSR     PC,SUBCLR
025322  104024                                 ERROR   24              ;CERR AFTER SCLR
```

```
 6369
 6370  025324  012765  000001  000020          MOV     #1,RKDC(R5)     ;CYL #
 6371
 6372  025332  012700  001674                  MOV     #RHTAB,R0
 6373
 6374  025336  012737  000025  003322  65$:    MOV     #RDHEAD,HCS1
 6375  025344  004737  043530                  JSR     PC,DATCMD       ;DO READ HEADER CMD & GET CONTR RDY
 6376  025350  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
 6377  025352  032737  100000  003322          BIT     #CERR,HCS1
 6378  025360  001405                          BEQ     66$
 6379  025362  104174                          ERROR   174             ;CERR AFTER READ HEADER  CMD
 6380  025364  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 6381  025370  000137  042706                  JMP     $EOP            ;ABORT DRIVE
 6382
 6383  025374  016520  000024          66$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 6384  025400  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
 6385  025404  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
 6386
 6387  025410  032765  100000  000010          BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
 6388  025416  001407                          BEQ     67$
 6389  025420  004737  045132                  JSR     PC,GSTAT
 6390  025424  104173                          ERROR   173             ;DATA LATE ON READ HEADER
 6391  025426  104401  056436                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 6392  025432  000137  042706                  JMP     $EOP            ;ABORT DRIVE
 6393
 6394  025436  020027  002100          67$:    CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
 6395  025442  001335                          BNE     65$             ;BR IF NO
 6396
 6397  025444  004737  047114                  JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
 6398                                                                  ;BEGINS WITH SECTOR 0
 6399  025450  005037  001442                  CLR     WDCNT           ;WORD COUNT
 6400  025454  012700  002100                  MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
 6401  025460  012701  001470                  MOV     #HDTAB,R1       ;CALC HEADER TABLE
 6402
 6403  025464  012037  001454          68$:    MOV     (R0)+,HDWD
 6404  025470  012137  003360                  MOV     (R1)+,TEMP1
 6405  025474  023737  001454  003360          CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
 6406  025502  001401                          BEQ     69$             ;BR IF COMPARE
 6407  025504  104202                          ERROR   202             ;READ HEADER MISMATCH
 6408
 6409  025506  005237  001442          69$:    INC     WDCNT
 6410  025512  023727  001442  000102          CMP     WDCNT,#66.      ;ALL WORDS DONE?
 6411  025520  001361                          BNE     68$             ;BR IF NO
 6412
 6413
 6414  025522  005037  001464                  CLR BYPFMT              ;ALLOW CORRECT FORMATTING
 6415
 6416
 6417
 6418                          ;;*****************************************************************
 6419                          ;*TEST 30       TEST RECALIBRATE CMD & READ HEADERS
 6420                          ;*
 6421                          ;*      THIS TEST DOES A RECALIBRATE & READS HEADERS.
 6422                          ;*      IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS
 6423                          ;*      TEST DID NOT OVERWRITE CYL 0 HEADERS.
 6424                          ;*      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
                                        ;*
                                        ;;*****************************************************************
025526  000004                  TST30:  SCOPE
025530  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
025536  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

025542  004737  045462                  JSR     PC,SUBCLR
025546  104024                          ERROR   24              ;CERR AFTER SCLR
025550  012737  000001  001350          MOV     #1,FRCYL        ;PARAMETERS
025556  005037  001352                  CLR     TOCYL           ;FOR
025562  012737  000001  001360          MOV     #1,CALDIF       ;ERROR TYPEOUTS
025570  012737  026566  001176          MOV     #10$,$ESCAPE
025576  012737  000013  003322          MOV     #RECAL,HCS1
025604  004737  043472                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
025610  104124                          ERROR   124             ;NO RDY AFTER RECAL CMD
025612  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
025620  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
025626  004737  045132                  JSR     PC,GSTAT
025632  032737  020000  003350          BIT     #D.RTZ,HMR2
025640  001001                          BNE     1$
025642  104307                          ERROR   307             ;RTZ NOT SET DURING RECAL CMD

025644  012737  030140  003412  1$:     MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
025652  005037  003414                  CLR     E.B0
025656  012737  025720  003416          MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
025664  012737  000001  003420          MOV     #1,E.B1

025672  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
025676  000001                          .WORD   T.A2!0!0                ;& MSGS SPECIFIED HERE
025700  104213                          ERROR   213             ;MSG A0 ERROR DURING RECAL CMD
025702  104214                          ERROR   214             ;MSG B0 ERROR
025704  104215                          ERROR   215             ;MSG A1 ERROR
025706  104216                          ERROR   216             ;MSG B1 ERROR
                                ;       TST     CYLDIF
                                ;       BEQ     2$
                                ;       ERROR   217             ;CYL DIFF INCORRECT DURING RECAL CMD.

025710  012737  026606  001176  2$:     MOV     #12$,$ESCAPE
025716  012737  177777  003360          MOV     #-1,TEMP1       ;SETUP TIMEOUT
025724  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
025730  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
025732  032737  100000  003322          BIT     #CERR,HCS1
025740  001401                          BEQ     3$
025742  104220                          ERROR   220             ;CERR AFTER RECAL CMD
025744                          3$:

025744  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
025752  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
025756  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
025764  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
025772  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
025776  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
```

```
 6481  026004  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6482
 6483  026012  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6484  026016  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6485  026020  104221                          ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
 6486  026022  104275                          ERROR   275             ;MSG B0 ERROR
 6487  026024  104222                          ERROR   222             ;MSG A1 ERROR
 6488  026026  104276                          ERROR   276             ;MSG B1 ERROR
 6489
 6490  026030  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
 6491  026036  004737  045132                  JSR     PC,GSTAT
 6492  026042  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
 6493  026046  001401                          BEQ     64$             ;BR IF YES
 6494  026050  104047                          ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
 6495  026052  005737  001364          64$:    TST     CYLADD          ;SEE IF MSG B2=0
 6496  026056  001401                          BEQ     65$             ;BR IF YES
 6497  026060  104050                          ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
 6498  026062                          65$:
 6499
 6500  026062  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6501  026070  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 6502  026076  012737  000005  003322          MOV     #CLEAR,HCS1
 6503  026104  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 6504  026110  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 6505  026112  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 6506  026116  000401                          BR      66$
 6507  026120  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 6508  026122                          66$:
 6509
 6510  026122  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 6511  026130  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 6512  026134  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
 6513  026142  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6514  026150  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 6515  026154  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6516  026162  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6517
 6518  026170  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6519  026174  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6520  026176  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 6521  026200  104265                          ERROR   265             ;MSG B0 ERROR
 6522  026202  104274                          ERROR   274             ;MSG A1 ERROR
 6523  026204  104266                          ERROR   266             ;MSG B1 ERROR
 6524
 6525  026206  004737  045462          4$:     JSR     PC,SUBCLR
 6526  026212  104024                          ERROR   24              ;CERR AFTER SCLR
 6527
 6528  026214  005037  001176                  CLR     $ESCAPE
 6529
 6530  026220  012700  001674                  MOV     #RHTAB,R0
 6531  026224  012737  000025  003322          MOV     #<RDHEAD>,HCS1
 6532  026232  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
 6533  026236  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
 6534  026240  032737  100000  003322          BIT     #CERR,HCS1
 6535  026246  001405                          BEQ     67$
 6536  026250  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
```

```
 6537  026252  104401  056436                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 6538  026256  000137  042706                JMP     $EOP            ;ABORT DRIVE
 6539
 6540  026262  016520  000024        67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 6541  026266  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
 6542  026272  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
 6543
 6544
 6545  026276  032765  100000  000010        BIT     #DLT,RKCS2(R5)
 6546  026304  001407                        BEQ     68$
 6547  026306  004737  045132                JSR     PC,GSTAT
 6548  026312  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
 6549  026314  104401  056436                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 6550  026320  000137  042706                JMP     $EOP            ;ABORT DRIVE
 6551  026324                        68$:
 6552
 6553  026324  023727  001674  000001        CMP     RHTAB,#1        ;CHECK WORD 0, CYL # ONLY
 6554  026332  001001                        BNE     5$
 6555  026334  104240                        ERROR   240             ;CYL 1 HEADERS ON CYL 0
 6556
 6557  026336  005737  001674          5$:   TST     RHTAB
 6558  026342  001401                        BEQ     6$
 6559  026344  104202                        ERROR   202             ;READ CYL WORD HEADER ERROR
 6560  026346                        6$:
 6561  026346  004737  047466                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 6562  026352  000575                        BR      TST31           ;;GO TO NEXT TEST
 6563                                                                ;RETURN HERE IF SW 14 IS SET OR
 6564                                                                ;SW 8 WITH SWR <7:0> APPLY
 6565  026354  004737  045462          8$:   JSR     PC,SUBCLR
 6566  026360  104024                        ERROR   24              ;CERR AFTER SCLR
 6567  026362  012765  000001  000020        MOV     #1,RKDC(R5)     ;RECONDITION BACK TO CYL 1
 6568
 6569  026370  012737  000017  003322        MOV     #SEEK,HCS1
 6570  026376  004737  043472                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 6571  026402  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
 6572
 6573  026404  013737  001426  003360        MOV     T50000,TEMP1
 6574  026412  004737  044176                JSR     PC,FATT2        ;FIND ATTN
 6575  026416  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
 6576  026420  032737  100000  003322        BIT     #CERR,HCS1
 6577  026426  001401                        BEQ     69$
 6578  026430  104210                        ERROR   210             ;CERR AFTER SEEK CMD.
 6579
 6580  026432  004737  045462          69$:  JSR     PC,SUBCLR
 6581  026436  104024                        ERROR   24              ;CERR AFTER SCLR
 6582
 6583
 6584
 6585  026440  012737  010340  003412        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 6586  026446  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
 6587  026452  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
 6588  026460  012737  000001  003420        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6589  026466  005037  003422                CLR     E.A2            ;EXPECTED MSG A2
 6590  026472  012737  000002  003424        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6591  026500  012737  000003  003430        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6592  026506  012737  000022  003424        MOV     #<BIT4!2>,E.B2  ;EXPECTED MSG B2 & ID FOR CYL 1
```

```
6593
6594  026514  004737  044310          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6595  026520  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6596  026522  104133                  ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6597  026524  104134                  ERROR   134             ;MSG B0 ERROR
6598  026526  104135                  ERROR   135             ;MSG A1 ERROR
6599  026530  104136                  ERROR   136             ;MSG B1 ERROR
6600  026532  023727  001364  000001  CMP     CYLADD,#1
6601  026540  001401                  BEQ     9$
6602  026542  104043                  ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
6603  026544                  9$:
6604  026544  005037  001176          CLR     $ESCAPE
6605  026550  005737  001410          TST     LPFLG
6606  026554  001402                  BEQ     70$
6607  026556  000177  152326          JMP     @$LPERR         ;SW 9 WAS SET.
6608  026562  000177  152320  70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6609  026566                  10$:
6610  026566  005237  001410          INC     LPFLG
6611  026572  032777  001000  152340  BIT     #SW9,@SWR       ;LOOP ON ERROR?
6612  026600  001265                  BNE     8$              ;YES, RECONDITION DRIVE
6613  026602  000137  025710          JMP     2$              ;RETURN TO MAINLINE
6614  026606                  12$:
6615  026606  005237  001410          INC     LPFLG
6616  026612  032777  001000  152320  BIT     #SW9,@SWR       ;LOOP ON ERROR?
6617  026620  001255                  BNE     8$              ;YES, RECONDITION DRIVE
6618  026622  000137  026206          JMP     4$              ;RETURN TO MAINLINE
6619                          ;;*****************************************************************
6620                          ;*TEST 31       SINGLE INCREMENT SEEKS TO LAST CYL
6621                          ;*
6622                          ;*      THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO THE LAST CYL
6623                          ;*      WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY
6624                          ;*      DESTROY DATA.
6625                          ;*
6626                          ;;*****************************************************************
6627  026626  000004          TST31:  SCOPE
6628  026630  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
6629  026636  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
6630
6631  026642  004737  045462          JSR     PC,SUBCLR
6632  026646  104024                  ERROR   24              ;CERR AFTER SCLR
6633  026650  005037  001350          CLR     FRCYL           ;FROM CYL
6634  026654  012737  000001  001352  MOV     #1,TOCYL        ;TO CYL
6635  026662  012737  000001  001360  MOV     #1,CALDIF       ;CALCULATED DIFF.
6636
6637  026670                  1$:
6638  026670  104415                  SCOP1
6639  026672  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
6640
6641  026676  004737  045462          JSR     PC,SUBCLR
6642  026702  104024                  ERROR   24              ;CERR AFTER SCLR
6643
6644  026704  012737  027466  001176  MOV     #10$,$ESCAPE
6645  026712  013765  001352  000020  MOV     TOCYL,RKDC(R5)  ;CYL TO SEEK TO
6646
6647  026720  012737  000017  003322  MOV     #SEEK,HCS1
6648  026726  004737  043472          JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
```

```
 6649  026732  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 6650  026734  012737  030140  003412          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 6651  026742  005037  003414                  CLR     E.B0
 6652  026746  012737  003720  003416          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 6653  026754  012737  000001  003420          MOV     #1,E.B1
 6654
 6655  026762  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6656  026766  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6657  026770  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 6658  026772  104204                          ERROR   204             ;MSG B0 ERROR
 6659  026774  104205                          ERROR   205             ;MSG A1 ERROR
 6660  026776  104206                          ERROR   206             ;MSG B1 ERROR
 6661
 6662  027000  023727  001362  000001          CMP     CYLDIF,#1
 6663  027006  001401                          BEQ     2$
 6664  027010  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
 6665
 6666  027012  012737  027506  001176  2$:     MOV     #12$,$ESCAPE
 6667  027020  013737  001422  003360          MOV     T2500,TEMP1     ;SETUP TIMEOUT
 6668
 6669  027026  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 6670  027032  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 6671  027034  032737  100000  003322          BIT     #CERR,HCS1
 6672  027042  001401                          BEQ     64$
 6673  027044  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 6674  027046                          64$:
 6675
 6676  027046  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 6677  027054  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 6678  027060  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6679  027066  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6680  027074  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 6681  027100  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6682  027106  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6683
 6684  027114  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6685  027120  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6686  027122  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 6687  027124  104134                          ERROR   134             ;MSG B0 ERROR
 6688  027126  104135                          ERROR   135             ;MSG A1 ERROR
 6689  027130  104136                          ERROR   136             ;MSG B1 ERROR
 6690  027132  005737  001362                  TST     CYLDIF
 6691  027136  001401                          BEQ     65$
 6692  027140  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 6693
 6694  027142                          65$:
 6695
 6696  027142  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6697  027150  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 6698  027156  012737  000005  003322          MOV     #CLEAR,HCS1
 6699  027164  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 6700  027170  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 6701  027172  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 6702  027176  000401                          BR      66$
 6703  027200  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 6704  027202                          66$:
```

```
6705
6706  027202  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
6707  027210  005037  003414                  CLR     E.B0                    ;EXPECTED MSG B0
6708  027214  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6709  027222  012737  000001  003420          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
6710  027230  005037  003422                  CLR     E.A2                    ;EXPECTED MSG A2
6711  027234  012737  000002  003424          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
6712  027242  012737  000003  003430          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
6713
6714  027250  004737  044310                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
6715  027254  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
6716  027256  104273                          ERROR   273                     ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6717  027260  104265                          ERROR   265                     ;MSG B0 ERROR
6718  027262  104274                          ERROR   274                     ;MSG A1 ERROR
6719  027264  104266                          ERROR   266                     ;MSG B1 ERROR
6720
6721  027266  023737  001364  001352          CMP     CYLADD,TOCYL
6722  027274  001401                          BEQ     3$
o723  027276  104207                          ERROR   207                     ;CYL ADDR IN RKMR2 NOT=RKDC
6724
6725  027300  023737  001352  012100  3$:     CMP     TOCYL,LC                ;LAST CYL DONE?
6726  027306  001407                          BEQ     4$                      ;BR IF YES
6727  027310  005237  001350                  INC     FRCYL                   ;ELSE DO ANOTHER
6728  027314  005237  001352                  INC     TOCYL
6729  027320  001402                          BEQ     4$                      ;BR IF YES
6730  027322  000137  026670                  JMP     1$
6731
6732  027326                          4$:
6733  027326  004737  047466                  JSR     PC,SWTST                ;SEE IF SW 14 OR 8 IS SET
6734  027332  000475                          BR      TST32                   ;;GO TO NEXT TEST
6735                                                                          ;RETURN HERE IF SW 14 IS SET OR
6736                                                                          ;SW 8 WITH SWR <7:0> APPLY
6737
6738
6739
6740  027334                          6$:
6741
6742  027334  004737  045462                  JSR     PC,SUBCLR
6743  027340  104024                          ERROR   24                      ;CERR AFTER SCRL
6744
6745  027342  013765  001352  000020  67$:     MOV     TOCYL,RKDC(R5)          ;CYL#
6746
6747  027350  012737  000017  003322          MOV     #SEEK,HCS1
6748  027356  004737  043472                  JSR     PC,DOCMD                ;DO SEEK CMD & GET CONTR READY
6749  027362  104131                          ERROR   131                     ;NO RDY AFTER SEEK CMD.
6750
6751  027364  013737  001426  003360          MOV     T50000,TEMP1
6752  027372  004737  044176                  JSR     PC,FATT2                ;FIND ATTN
6753  027376  104132                          ERROR   132                     ;NO ATTN AFTER SEEK CMD
6754  027400  032737  100000  003322          BIT     #CERR,HCS1
6755  027406  001401                          BEQ     69$
6756  027410  104210                          ERROR   210                     ;CERR AFTER SEEK CMD.
6757
6758  027412  004737  045462          69$:    JSR     PC,SUBCLR
6759  027416  104024                          ERROR   24                      ;CERR AFTER SCLR
6760
```

```
027420  023737  001352  000000          CMP     TOCYL,0 ;LAST CYL DONE?
027426  001403                          BEQ     68$             ;BR IF YES
027430  005337  001352                  DEC     TOCYL           ;ELSE DO ANOTHER
027434  000742                          BR      67$

027436  004737  045462          68$:    JSR     PC,SUBCLR
027442  104024                          ERROR   24              ;CERR AFTER SCLR

027444  005037  001176                  CLR     $ESCAPE
027450  005737  001410                  TST     LPFLG
027454  001402                          BEQ     70$
027456  000177  151426                  JMP     @$LPERR         ;SW 9 WAS SET.
027462  000177  151420          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET



027466                          10$:
027466  005237  001410                  INC     LPFLG
027472  032777  001000  151440          BIT     #SW9,@SWR       ;LOOP ON ERROR?
027500  001315                          BNE     6$              ;YES, RECONDITION DRIVE
027502  000137  027012                  JMP     2$              ;RETURN TO MAINLINE

027506                          12$:
027506  005237  001410                  INC     LPFLG
027512  032777  001000  151420          BIT     #SW9,@SWR       ;LOOP ON ERROR?
027520  001305                          BNE     6$              ;YES, RECONDITION DRIVE
027522  000137  027326                  JMP     4$              ;RETURN TO MAINLINE

                                ;;*****************************************************************
                                ;*TEST 32        READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
                                ;*
                                ;*      THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), 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
                                ;*
                                ;;*****************************************************************
027526  000004                  TST32:  SCOPE
027530  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
027536  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

027542  004737  045462                  JSR     PC,SUBCLR
027546  104024                          ERROR   24              ;CERR AFTER SCLR
```

```
6817  027550  005037  003362                  CLR     TEMP2           ;SECTOR CTR
6818  027554  005037  003364                  CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
6819                                                                  ;1=22 SECTOR SOFTWARE DETECTED TABLE
6820                                                                  ;2=DONE
6821  027560  012737  002304  003366          MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE INFO
6822  027566  013765  003366  000004          MOV     TEMP4,RKBA(R5)
6823  027574  012737  001000  003370          MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
6824  027602  013765  003370  000006          MOV     TEMP5,RKDA(R5)
6825
6826  027610  013765  012100  000020  1$:     MOV     LC,RKDC(R5)     ;LAST CYL
6827  027616  012765  177400  000002          MOV     #-256.,RKWC(R5) ;LOAD WORD CT
6828  027624  012737  000021  003322          MOV     #RDDATA,HCS1
6829  027632  004737  043530                  JSR     PC,DATCMD       ;DO COMMAND
6830  027636  104226                          ERROR   226             ;NO RDY AFTER READ DATA CMD
6831  027640  004737  045132                  JSR     PC,GSTAT        ;GET FRESH STATUS
6832  027644  032737  100000  003322          BIT     #CERR,HCS1
6833  027652  001470                          BEQ     8$
6834  027654  104227                          ERROR   227             ;CERR AFTER READ DATA CMD
6835
6836  027656  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
6837  027664  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
6838  027670  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
6839  027676  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6840  027704  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
6841  027710  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6842  027716  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6843
6844  027724  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6845  027730  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6846  027732  104051                          ERROR   51              ;MSG A0 ERROR AFTER READ DATA CMD
6847  027734  104052                          ERROR   52              ;MSG B0 ERROR
6848  027736  104112                          ERROR   112             ;MSG A1 ERROR
6849  027740  104113                          ERROR   113             ;MSG B1 ERROR
6850
6851  027742  004737  045462                  JSR     PC,SUBCLR
6852  027746  104024                          ERROR   24              ;CERR AFTER SUBCLR
6853
6854  027750  005237  003362                  INC     TEMP2
6855  027754  023727  003362  000005          CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
6856  027762  001007                          BNE     5$
6857  027764  005737  003364                  TST     TEMP3
6858  027770  001002                          BNE     2$
6859  027772  104233                          ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
6860  027774  000414                          BR      3$
6861  027776  104230                  2$:     ERROR   230             ;CANT READ SECTORS 10,12,14,16,18,20
6862  030000  000412                          BR      3$
6863
6864  030002  013765  003366  000004  5$:     MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
6865  030010  062737  000002  003370          ADD     #2,TEMP5        ;SETUP TO READ 2 SECTORS FROM LAST
6866  030016  013765  003370  000006          MOV     TEMP5,RKDA(R5)
6867  030024  000671                          BR      1$
6868
6869  030026  005237  001456          3$:     INC     BSERR           ;SET BSE FLAG
6870  030032  000454                          BR      TST33           ;;GO TO NEXT TEST
6871
6872  030034  005737  002312          8$:     TST     BSE22H+6        ;TEST CARTRIDGE TYPE
```

```
6873  030040  001404                        BEQ     9$              ;BRANCH IF DATA CARTRIDGE
6874  030042  104235                        ERROR   235             ;ALIGNMENT CARTRIDGE USED
6875  030044  005237  001456                INC     BSERR           ;SET BSE ERROR FLAG
6876  030050  000426                        BR      10$
6877
6878  030052  005237  003364          9$:   INC     TEMP3
6879  030056  023727  003364  000001        CMP     TEMP3,#1
6880  030064  001020                        BNE     10$
6881  030066  005037  003362                CLR     TEMP2
6882  030072  012737  054474  003366        MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
6883  030100  013765  003366  000004        MOV     TEMP4,RKBA(R5)
6884  030106  012737  001012  003370        MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12
6885  030114  013765  003370  000006        MOV     TEMP5,RKDA(R5)
6886  030122  000137  027610                JMP     1$              ;REPEAT
6887
6888  030126  005737  001216          10$:  TST     $PASS
6889  030132  001014                        BNE     TST33           ;;GO TO NEXT TST IF NOT 1'ST PASS
6890  030134  104401  056407                TYPE    ,MSG17          ;CART SERIAL #
6891  030140  012746  002304                MOV     #BSE22H,-(SP)
6892  030144  004737  053700                JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
6893  030150  004737  054250                JSR     PC,$SUPRS       ;TYPE SERIAL #
6894  030154  104401  001205                TYPE    ,$CRLF
6895  030160  104401  001205                TYPE    ,$CRLF
6896
6897                                  ;;*****************************************************************
6898                                  ;*TEST 33        DETECT INNER LIMIT
6899                                  ;*
6900                                  ;*      THIS TEST VERIFIES THAT THE LAST CYL IN THE ABOVE
6901                                  ;*      TEST WAS 632 (1456) BY DETECTING INNER LIMIT AS THE ADJACENT CYL.
6902                                  ;*      IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON THE LAST CYL
6903                                  ;*      & THAT BSE INFO IS NOT VALID.  THE FORMAT PACK TEST
6904                                  ;*      & ALL READ-WRITE TESTS ARE BYPASSED
6905                                  ;*      TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
6906                                  ;*      SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.
6907                                  ;*
6908                                  ;;*****************************************************************
6909  030164  000004                  TST33:  SCOPE
6910  030166  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6911  030174  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6912
6913  030200  004737  045462                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
6914  030204  104024                          ERROR   24              ;CERR AFTER SCLR
6915
6916  030206  005037  001410                  CLR     LPFLG
6917  030212  005237  001462                  INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
6918  030216  005237  003304                  INC     UNLD            ;USED FOR VALID HALT
6919
6920  030222  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
6921  030230  013765  012076  000020          MOV     LCM1,RKDC(R5)
6922  030236  012737  000017  003322          MOV     #SEEK,HCS1
6923  030244  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
6924  030250  104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
6925  030252  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
6926  030256  104125                          ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
6927  030260  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
6928  030266  012737  001200  003414          MOV     #<D.FLT!D.PAR>,E.B0
```

```
 6929  030274  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 6930  030302  012737  000001  003420          MOV     #1,E.B1
 6931
 6932  030310  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6933  030314  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 6934  030316  104110                          ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
 6935  030320  104111                          ERROR   111             ;MSG B0 ERROR
 6936  030322  104146                          ERROR   146             ;MSG A1 ERROR
 6937  030324  104147                          ERROR   147             ;MSG B1 ERROR
 6938
 6939  030326  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6940  030334  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 6941  030342  012737  000005  003322          MOV     #CLEAR,HCS1
 6942  030350  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 6943  030354  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 6944  030356  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 6945  030362  000401                          BR      65$
 6946  030364  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 6947  030366                          65$:
 6948
 6949  030366  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 6950  030374  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 6951  030400  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6952  030406  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6953  030414  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 6954  030420  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6955  030426  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6956
 6957  030434  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6958  030440  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6959  030442  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 6960  030444  104265                          ERROR   265             ;MSG B0 ERROR
 6961  030446  104274                          ERROR   274             ;MSG A1 ERROR
 6962  030450  104266                          ERROR   266             ;MSG B1 ERROR
 6963
 6964
 6965  030452  013765  012100  000020          MOV     LC,RKDC(R5)
 6966  030460  012737  000017  003322          MOV     #SEEK,HCS1
 6967  030466  004737  043472                  JSR     PC,DOCM         ;DO SEEK CMD & GET CONTR READY
 6968  030472  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 6969  030474  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6970  030502  004737  045132                  JSR     PC,GSTAT
 6971  030506  004737  046316                  JSR     PC,FLIM         ;FIND LIMIT DETECT
 6972  030512  104160                          ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
 6973
 6974  030514  032737  040000  003350          BIT     #D.UNLD,HMR2
 6975  030522  001003                          BNE     1$
 6976  030524  104305                          ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
 6977  030526  000137  031366                  JMP     30$             ;BYPASS REST OF TEST
 6978
 6979  030532  012737  031304  001176  1$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
 6980  030540  012737  070140  003412          MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
 6981  030546  012737  002200  003414          MOV     #<D.SKI!D.FLT>,E.B0
 6982  030554  012737  045720  003416          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 6983  030562  012737  030001  003420          MOV     #<D.LIMD!D.NMOV!1>,E.B1
 6984                                  ;       CHECK   161,162,163,164,<AFTER INNER LIMIT DETECT>,0,0,0
```

```
 6985  030570  004737  044310                    JSR     PC,CHKMSG       ;CHECK MESSAGE A0,B0,A1,B1
 6986  030574  000000                            .WORD   0!0!0
 6987  030576  104161                            ERROR   161             ;MAY BE A0 ERROR
 6988  030600  104162                            ERROR   162             ;MAY BE B0 ERROR
 6989  030602  104163                            ERROR   163             ;MAY BE B1 ERROR
 6990  030604  000240                            NOP                     ;NEED FOR THE CALLIN SEQ
 6991  030606  032737  020000  003400            BIT     #D.LIMD,H.B1    ;SEE IF LIMIT DETECT BIT SET IN B1
 6992  030614  001003                            BNE     3$              ;BRANCH IF SO
 6993  030616  104401  061436                    TYPE    ,EM42           ;OTHERWISE REPORT ERROR
 6994  030622  104164                            ERROR   164
 6995
 6996  030624  004737  044050            3$:     JSR     PC,TSTATN
 6997  030630  104165                            ERROR   165             ;NO ATTN AFTER INNER LIMIT DETECT
 6998  030632  005037  001462                    CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1
 6999
 7000  030636  004737  045462                    JSR     PC,SUBCLR       ;SUBSYS CLR
 7001  030642  104024                            ERROR   24              ;CERR AFTER SCLR
 7002  030644  013737  001414  003362            MOV     T10,TEMP2       ;SET UP TIMEOUT
 7003  030652  004737  046374                    JSR     PC,FHDHM        ;FIND HEAD HOME
 7004  030656  104166                            ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
 7005  030660  004737  046450                    JSR     PC,FLOAD        ;FIND LOAD HEADS
 7006  030664  104167                            ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
 7007  030666  013737  001416  003362            MOV     T100,TEMP2      ;SETUP TIMEOUT
 7008  030674  004737  044102                    JSR     PC,FATT1        ;FIND ATTN
 7009  030700  104067                            ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
 7010  030702  005037  001176            2$:     CLR     $ESCAPE
 7011  030706  005037  003304                    CLR     UNLD            ;CLEAR FLAG
 7012
 7013  030712  012737  050340  003412            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 7014  030720  005037  003414                    CLR     E.B0            ;EXPECTED MSG B0
 7015  030724  012737  001720  003416            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 7016  030732  012737  000001  003420            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 7017  030740  005037  003422                    CLR     E.A2            ;EXPECTED MSG A2
 7018  030744  012737  000002  003424            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 7019  030752  012737  000003  003430            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 7020
 7021  030760  004737  044310                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7022  030764  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7023  030766  104063                            ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOADING
 7024  030770  104064                            ERROR   64              ;MSG B0 ERROR
 7025  030772  104065                            ERROR   65              ;MSG A1 ERROR
 7026  030774  104066                            ERROR   66              ;MSG B1 ERROR
 7027                                      ;     CWD2    175,176,<AT END OF HEAD LOADING>
 7028  030776  012765  000002  000026            MOV     #2,RKMR1(R5)    ;SELECT MESSAGE
 7029  031004  004737  045132                    JSR     PC,GSTAT        ;GET STATUS AND MR2,MR3
 7030  031010  005737  001364                    TST     CYLADD          ;RECAL SUCCFUL ?
 7031  031014  001401                            BEQ     64$             ;BRANCH IF SO
 7032  031016  104050                            ERROR   50              ;REPORT ERROR
 7033  031020                            64$:
 7034
 7035  031020  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
 7036  031026  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
 7037  031034  012737  000005  003322            MOV     #CLEAR,HCS1
 7038  031042  004737  043472                    JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 7039  031046  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 7040  031050  004737  044050                    JSR     PC,TSTATN       ;TEST FOR ATTN
```

```
 7041  031054  000401                          BR      66$
 7042  031056  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 7043  031060                          66$:
 7044
 7045  031060  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
 7046  031066  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 7047  031072  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 7048  031100  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 7049  031106  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 7050  031112  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 7051  031120  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 7052
 7053  031126  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7054  031132  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7055  031134  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 7056  031136  104265                          ERROR   265             ;MSG B0 ERROR
 7057  031140  104274                          ERROR   274             ;MSG A1 ERROR
 7058  031142  104266                          ERROR   266             ;MSG B1 ERROR
 7059
 7060  031144  004737  047466                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 7061  031150  000506                          BR      TST34           ;;GO TO NEXT TEST
 7062                                                                  ;RETURN HERE IF SW 14 IS SET OR
 7063                                                                  ;SW 8 WITH SWR <7:0> APPLY
 7064
 7065
 7066  031152                          10$:
 7067
 7068  031152  004737  045462                  JSR     PC,SUBCLR
 7069  031156  104024                          ERROR   24              ;CERR AFTER SCRL
 7070
 7071  031160  013765  001352  000020  67$:    MOV     TOCYL,RKDC(R5)  ;CYL#
 7072
 7073  031166  012737  000017  003322          MOV     #SEEK,HCS1
 7074  031174  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 7075  031200  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
 7076
 7077  031202  013737  001426  003360          MOV     T50000,TEMP1
 7078  031210  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 7079  031214  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 7080  031216  032737  100000  003322          BIT     #CERR,HCS1
 7081  031224  001401                          BEQ     69$
 7082  031226  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
 7083
 7084  031230  004737  045462          69$:    JSR     PC,SUBCLR
 7085  031234  104024                          ERROR   24              ;CERR AFTER SCLR
 7086
 7087  031236  023737  001352  012100          CMP     TOCYL,LC        ;LAST CYL DONE?
 7088  031244  001403                          BEQ     68$             ;BR IF YES
 7089  031246  005237  001352                  INC     TOCYL           ;ELSE DO ANOTHER
 7090  031252  000742                          BR      67$
 7091
 7092  031254  004737  045462          68$:    JSR     PC,SUBCLR
 7093  031260  104024                          ERROR   24              ;CERR AFTER SCLR
 7094
 7095  031262  005037  001176                  CLR     $ESCAPE
 7096  031266  005737  001410                  TST     LPFLG
```

```
 7097  031272  001402                        BEQ     70$
 7098  031274  000177  147610                JMP     @$LPERR         ;SW 9 WAS SET.
 7099  031300  000177  147602        70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 7100
 7101  031304                        20$:
 7102
 7103  031304  004737  045462                JSR     PC,SUBCLR
 7104  031310  104024                        ERROR   24              ;CERR AFTER SCLR
 7105
 7106  031312  012737  000011  003322        MOV     #SRTSPL,HCS1
 7107  031320  004737  043472                JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
 7108  031324  104121                        ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
 7109
 7110  031326  013737  001420  003362        MOV     T500,TEMP2      ;SETUP  TIMEOUT
 7111  031334  004737  044102                JSR     PC,FATT1        ;FIND ATTN
 7112  031340  104067                        ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
 7113
 7114  031342  005037  003304                CLR     UNLD
 7115  031346  005237  001410                INC     LPFLG
 7116  031352  032777  001000  147560        BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7117  031360  001274                        BNE     10$             ;YES, RECONDITION DRIVE
 7118  031362  000137  030702                JMP     2$              ;RETURN TO MAINLINE
 7119  031366                        30$:
 7120
 7121  031366                        FORM:
 7122                                ;;*****************************************************************
 7123                                ;*TEST 34       FORMAT PACK
 7124                                ;*
 7125                                ;*      THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
 7126                                ;*      DOING 1 CYL INCREMENTAL SEEKS
 7127                                ;*      FROM 0 TO 632 (1456 FOR RK07) WITH WRITE HEADER CMDS (ALL TRACKS).
 7128                                ;*      HEADERS WILL BE READ IN THE NEXT TEST
 7129                                ;*
 7130                                ;;*****************************************************************
 7131  031366  000004                TST34:  SCOPE
 7132  031370  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
 7133
 7134
 7135  031376  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 7136  031402  005737  001342                TST     MODTST          ;SEE IF MODULE TESTING
 7137  031406  001402                        BEQ     22$             ;BR IF NO
 7138  031410  104401  056522                TYPE    ,MSG20          ;RUNNING MODIFIED VERSION OF TEST
 7139
 7140  031414  005737  001460        22$:    TST     LIMERR          ;CHECK IF FOUND LIMIT DETECT ERROR
 7141  031420  001403                        BEQ     1$
 7142  031422  104170                        ERROR   170             ;FATAL ERROR
 7143  031424  000137  042706                JMP     $EOP            ;ABORT BAL OF TESTS
 7144  031430  005737  001456        1$:     TST     BSERR           ;CHECK IF FOUND BSE INFO OK
 7145  031434  001403                        BEQ     2$
 7146  031436  104177                        ERROR   177             ;FORMAT TEST BYPASSED-BSE ERROR
 7147  031440  000137  042706                JMP     $EOP
 7148
 7149  031444  004737  045462        2$:     JSR     PC,SUBCLR
 7150  031450  104024                        ERROR   24              ;CERR AFTER SCLR
 7151
 7152  031452  104401  056222                TYPE    ,MSG12          ;FORMATTING PACK, PLEASE WAIT
```

```
7153
7154  031456  005037  001352                  CLR     TOCYL
7155
7156  031462  013737  001352  001366          MOV     TOCYL,CALADD    ;SETUP
7157  031470  012737  000000  001430          MOV     #0,HEAD         ;TO FILL
7158  031476  012737  000000  001436          MOV     #0,FORMAT       ;HEADER
7159  031504  004737  046572                  JSR     PC,FHDTAB       ;TABLE
7160
7161
7162  031510  012765  001470  000004  9$:     MOV     #HDTAB,RKBA(R5) ;THIS SECTION
7163  031516  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;OF CODE
7164  031524  000337  001430                  SWAB    HEAD            ;IS TO RESTORE STANDARD FORMAT
7165  031530  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;TO CYL 0
7166  031536  000337  001430                  SWAB    HEAD            ;HEAD 0,1 & 2
7167
7168  031542  012737  000027  003322          MOV     #<WRHEAD>,HCS1
7169  031550  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
7170  031554  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
7171  031556  004737  045132                  JSR     PC,GSTAT        ;GET FRESH STATUS
7172  031562  032737  100000  003322          BIT     #CERR,HCS1
7173  031570  001405                          BEQ     64$
7174  031572  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
7175  031574  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7176  031600  000137  042706                  JMP     $EOP            ;ABORT DRIVE
7177  031604                          64$:
7178
7179  031604  005237  001430                  INC     HEAD
7180  031610  023727  001430  000003          CMP     HEAD,#3
7181  031616  001403                          BEQ     11$             ;BR IF ALL HEADS DONE
7182
7183  031620  004737  046572                  JSR     PC,FHDTAB
7184  031624  000731                          BR      9$
7185
7186  031626  012737  000001  001366  11$:     MOV     #1,CALADD       ;SETUP
7187  031634  005037  001430                  CLR     HEAD            ;FOR
7188  031640  005037  001436                  CLR     FORMAT          ;FHDTAB ROUTINE
7189
7190  031644  012737  000001  001360          MOV     #1,CALDIF       ;SETUP
7191  031652  005037  001350                  CLR     FRCYL           ;FOR
7192  031656  012737  000001  001352          MOV     #1,TOCYL        ;ERROR REPORT
7193                                                                  ;START FORMATTING CYL 1 TO 410 HERE
7194
7195
7196  031664                          3$:
7197  031664  104415                          SCOP1
7198  031666  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7199
7200  031672  004737  045462                  JSR     PC,SUBCLR
7201  031676  104024                          ERROR   24              ;CERR AFTER SCLR
7202
7203  031700  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
7204  031704  001404                          BEQ     18$             ;BR IF NO
7205  031706  012737  033376  001176          MOV     #16$,$ESCAPE
7206  031714  000403                          BR      19$
7207  031716  012737  033102  001176  18$:     MOV     #10$,$ESCAPE
7208  031724  013765  001366  000020  19$:     MOV     CALADD,RKDC(R5) ;CYL #
```

```
7209  031732  000337  001430                  SWAB    HEAD
7210  031736  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;HEAD #
7211  031744  000337  001430                  SWAB    HEAD
7212
7213  031750  012737  000017  003322          MOV     #SEEK,HCS1
7214  031756  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
7215  031762  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7216  031764  012737  030140  003412          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7217  031772  005037  003414                  CLR     E.B0
7218  031776  012737  003720  003416          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7219  032004  012737  000001  003420          MOV     #1,E.B1
7220
7221  032012  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7222  032016  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7223  032020  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7224  032022  104204                          ERROR   204             ;MSG B0 ERROR
7225  032024  104205                          ERROR   205             ;MSG A1 ERROR
7226  032026  104206                          ERROR   206             ;MSG B1 ERROR
7227
7228  032030  023727  001362  000001          CMP     CYLDIF,#1
7229  032036  001401                          BEQ     4$
7230  032040  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
7231
7232  032042  005737  001342          4$:     TST     MODTST          ;SEE IF MODULE TESTING
7233  032046  001404                          BEQ     20$             ;BR IF NO
7234  032050  012737  033416  001176          MOV     #17$,$ESCAPE
7235  032056  000403                          BR      21$
7236
7237  032060  012737  033122  001176  20$:     MOV     #12$,$ESCAPE
7238  032066  012737  004704  003360  21$:     MOV     #2500.,TEMP1    ;SETUP TIMEOUT
7239
7240  032074  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
7241  032100  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7242  032102  032737  100000  003322          BIT     #CERR,HCS1
7243  032110  001401                          BEQ     65$
7244  032112  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7245  032114                          65$:
7246
7247  032114  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7248  032122  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
7249  032126  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7250  032134  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7251  032142  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
7252  032146  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7253  032154  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7254
7255  032162  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7256  032166  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7257  032170  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7258  032172  104134                          ERROR   134             ;MSG B0 ERROR
7259  032174  104135                          ERROR   135             ;MSG A1 ERROR
7260  032176  104136                          ERROR   136             ;MSG B1 ERROR
7261  032200  005737  001362                  TST     CYLDIF
7262  032204  001401                          BEQ     66$
7263  032206  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7264
```

```
7265  032210                          66$:
7266
7267  032210  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7268  032216  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7269  032224  012737  000005  003322          MOV     #CLEAR,HCS1
7270  032232  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
7271  032236  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7272  032240  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
7273  032244  000401                          BR      67$
7274  032246  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7275  032250                          67$:
7276
7277  032250  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
7278  032256  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
7279  032262  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
7280  032270  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7281  032276  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
7282  032302  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7283  032310  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7284
7285  032316  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7286  032322  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7287  032324  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7288  032326  104265                          ERROR   265             ;MSG B0 ERROR
7289  032330  104274                          ERROR   274             ;MSG A1 ERROR
7290  032332  104266                          ERROR   266             ;MSG B1 ERROR
7291
7292  032334  023737  001364  001366          CMP     CYLADD,CALADD
7293  032342  001401                          BEQ     5$
7294  032344  104232                          ERROR   232             ;CYL ADDR IN RKMR2 NOT=RKDC
7295
7296  032346                          5$:
7297  032346  104415                          SCOP1
7298  032350  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7299
7300  032354  004737  045462                  JSR     PC,SUBCLR
7301  032360  104024                          ERROR   24              ;CERR AFTER SCLR
7302
7303  032362  005037  001176                  CLR     $ESCAPE
7304  032366  004737  046572                  JSR     PC,FHDTAB       ;FILL HEADER TABLE
7305  032372  000337  001430                  SWAB    HEAD
7306  032376  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;SET TRACK #
7307  032404  000337  001430                  SWAB    HEAD
7308  032410  012765  001470  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
7309  032416  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD CT
7310  032424  013765  001366  000020          MOV     CALADD,RKDC(R5) ;CYL #
7311
7312
7313  032432  012737  000027  003322          MOV     #<WRHEAD>,HCS1
7314  032440  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
7315  032444  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
7316  032446  004737  045132                  JSR     PC,GSTAT        ;GET FRESH STATUS
7317  032452  032737  100000  003322          BIT     #CERR,HCS1
7318  032460  001405                          BEQ     68$
7319  032462  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
7320  032464  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
```

```
 7321  032470  000137  042706                  JMP     $EOP            ;ABORT DRIVE
 7322  032474                          68$:
 7323
 7324  032474  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 7325  032502  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 7326  032506  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
 7327  032514  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 7328  032522  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 7329  032526  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 7330  032534  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 7331
 7332  032542  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7333  032546  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7334  032550  104277                          ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
 7335  032552  104267                          ERROR   267             ;MSG B0 ERROR
 7336  032554  104300                          ERROR   300             ;MSG A1 ERROR
 7337  032556  104270                          ERROR   270             ;MSG B1 ERROR
 7338
 7339
 7340  032560  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
 7341  032564  001402                          BEQ     23$             ;BR IF NO
 7342  032566  000137  033142                  JMP     14$             ;ELSE RESTORE HEADERS ONLY
```

```
7343
7344  032572  005237  001430          23$:    INC     HEAD
7345  032576  023727  001430  000002          CMP     HEAD,#2
7346  032604  001006                          BNE     6$
7347  032606  023737  001366  012100          CMP     CALADD,LC       ;HEAD 2, SEE IF ON LAST CYL
7348  032614  001002                          BNE     6$              ;DO NOT WRITE ON LAST CYL HEAD 2
7349  032616  000137  033042                  JMP     7$
7350
7351  032622  023727  001430  000003  6$:     CMP     HEAD,#3         ;ALL HEADS DONE?
7352  032630  001246                          BNE     5$              ;BR IF NO
7353  032632  005037  001430                  CLR     HEAD            ;ALL HEADS ON CYL DONE
7354  032636  005237  001366                  INC     CALADD          ;GO TO NEXT CYL
7355  032642  005237  001350                  INC     FRCYL           ;FOR ERROR REPORT
7356  032646  005237  001352                  INC     TOCYL           ;FOR ERROR REPORT
7357  032652  005737  003310                  TST     HPEND           ;SEE IF HALT PENDING
7358  032656  001002                          BNE     24$             ;BR IF YES
7359  032660  000137  031664                  JMP     3$              ;ELSE KEEP FORMATTING
7360
7361  032664  005037  003310          24$:    CLR     HPEND           ;CLEAR FOR FUTURE FORMATTING
7362  032670  005037  003306                  CLR     BADHDR          ;HEADERS NOW OK
7363  032674  000137  047536                  JMP     STOP            ;GO & HALT THE CPU
7364
7365  032700  005037  001366          8$:     CLR     CALADD
7366  032704  005037  001350                  CLR     FRCYL
7367
7368  032710  004737  045462                  JSR     PC,SUBCLR
7369  032714  104024                          ERROR   24              ;CERR AFTER SCRL
7370
7371  032716  013765  001352  000020  69$:    MOV     TOCYL,RKDC(R5)  ;CYL#
7372
7373  032724  012737  000017  003322          MOV     #SEEK,HCS1
7374  032732  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
7375  032736  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
7376
7377  032740  013737  001426  003360          MOV     T50000,TEMP1
7378  032746  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
7379  032752  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7380  032754  032737  100000  003322          BIT     #CERR,HCS1
7381  032762  001401                          BEQ     71$
7382  032764  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
7383
7384  032766  004737  045462          71$:    JSR     PC,SUBCLR
7385  032772  104024                          ERROR   24              ;CERR AFTER SCLR
7386
7387  032774  023737  001352  000000          CMP     TOCYL,0 ;LAST CYL DONE?
7388  033002  001403                          BEQ     70$             ;BR IF YES
7389  033004  005337  001352                  DEC     TOCYL           ;ELSE DO ANOTHER
7390  033010  000742                          BR      69$
7391
7392  033012  004737  045462          70$:    JSR     PC,SUBCLR
7393  033016  104024                          ERROR   24              ;CERR AFTER SCLR
7394
7395  033020  005037  001176                  CLR     $ESCAPE
7396  033024  005737  001410                  TST     LPFLG
7397  033030  001402                          BEQ     72$
7398  033032  000177  146052                  JMP     @$LPERR         ;SW 9 WAS SET.
```

```
 7399  033036  000177  146044          72$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 7400
 7401
 7402  033042  004737  050044          7$:     JSR     PC,HPEN         ;SEE IF HALT PENDING
 7403  033046  000137  032664                  JMP     24$             ;RET HERE IF YES & EXIT
 7404                                          ;ELSE RET HERE
 7405  033052  004737  047466                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 7406  033056  000567                          BR      TST35           ;;GO TO NEXT TEST
 7407                                          ;RETURN HERE IF SW 14 IS SET OR
 7408                                          ;SW 8 WITH SWR <7:0> APPLY
 7409  033060  005037  001176                  CLR     $ESCAPE
 7410  033064  005737  001410                  TST     LPFLG
 7411  033070  001402                          BEQ     73$
 7412  033072  000177  146012                  JMP     @$LPERR         ;SW 9 WAS SET.
 7413  033076  000177  146004          73$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 7414
 7415  033102                          10$:
 7416  033102  005237  001410                  INC     LPFLG
 7417  033106  032777  001000  146024          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7418  033114  001271                          BNE     8$              ;YES, RECONDITION DRIVE
 7419  033116  000137  032042                  JMP     4$              ;RETURN TO MAINLINE
 7420
 7421  033122                          12$:
 7422  033122  005237  001410                  INC     LPFLG
 7423  033126  032777  001000  146004          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7424  033134  001261                          BNE     8$              ;YES, RECONDITION DRIVE
 7425  033136  000137  032346                  JMP     5$              ;RETURN TO MAINLINE
 7426
 7427
 7428  033142  005237  001430          14$:    INC     HEAD
 7429  033146  023727  001430  000003          CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
 7430  033154  001402                          BEQ     15$             ;BR IF YES TO GO BACK TO CYL 0
 7431  033156  000137  032346                  JMP     5$              ;ELSE REPEAT FOR NEXT HEAD
 7432
 7433  033162  005065  000006          15$:    CLR     RKDA(R5)        ;SEEK TO CYL 0 & READ HEADERS
 7434  033166  005037  001352                  CLR     TOCYL           ;TO RECONDITION DRIVE
 7435
 7436  033172  012737  000017  003322          MOV     #SEEK,HCS1
 7437  033200  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 7438  033204  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
 7439
 7440  033206  013737  001426  003360          MOV     T50000,TEMP1
 7441  033214  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 7442  033220  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 7443  033222  032737  100000  003322          BIT     #CERR,HCS1
 7444  033230  001401                          BEQ     74$
 7445  033232  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
 7446
 7447  033234  004737  045462          74$:    JSR     PC,SUBCLR
 7448  033240  104024                          ERROR   24              ;CERR AFTER SCLR
 7449
 7450
 7451
 7452  033242  012700  001674                  MOV     #RHTAB,R0
 7453  033246  012737  000025  003322          MOV     #<RDHEAD>,HCS1
 7454  033254  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
```

```
 7455  033260  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
 7456  033262  032737  100000  003322        BIT     #CERR,HCS1
 7457  033270  001405                        BEQ     76$
 7458  033272  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
 7459  033274  104401  056436                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 7460  033300  000137  042706                JMP     $EOP            ;ABORT DRIVE
 7461
 7462  033304  016520  000024        76$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 7463  033310  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
 7464  033314  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
 7465
 7466
 7467  033320  032765  100000  000010        BIT     #DLT,RKCS2(R5)
 7468  033326  001407                        BEQ     77$
 7469  033330  004737  045132                JSR     PC,GSTAT
 7470  033334  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
 7471  033336  104401  056436                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 7472  033342  000137  042706                JMP     $EOP            ;ABORT DRIVE
 7473  033346                        77$:
 7474
 7475  033346  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
 7476  033354  001401                        BEQ     75$             ;BR IF SAME
 7477  033356  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
 7478  033360                        75$:
 7479
 7480  033360  004737  050044                JSR     PC,HPEN         ;SEE IF HALT PENDING
 7481  033364  000137  032664                JMP     24$             ;RET HERE IF YES
 7482                                                                ;ELSE RET HERE & EXIT
 7483  033370  004737  047466                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 7484  033374  000420                        BR      TST35           ;;GO TO NEXT TEST
 7485                                                                ;RETURN HERE IF SW 14 IS SET OR
 7486                                                                ;SW 8 WITH SWR <7:0> APPLY
 7487  033376                        16$:
 7488  033376  005237  001410                INC     LPFLG
 7489  033402  032777  001000  145530        BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7490  033410  001264                        BNE     15$             ;YES, RECONDITION DRIVE
 7491  033412  000137  032042                JMP     4$              ;RETURN TO MAINLINE
 7492  033416                        17$:
 7493  033416  005237  001410                INC     LPFLG
 7494  033422  032777  001000  145510        BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7495  033430  001254                        BNE     15$             ;YES, RECONDITION DRIVE
 7496  033432  000137  032346                JMP     5$              ;RETURN TO MAINLINE
 7497
 7498
 7499  033436                        13$:
 7500
 7501
 7502
 7503                                        ;;**********************************************************************
 7504                                        ;*TEST 35       DECREMENT FROM LAST CYL TO 0 & READ HEADERS
 7505                                        ;*
 7506                                        ;*      THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
 7507                                        ;*      SINGLE CYL INCREMENTAL SEEKS.
 7508                                        ;*
 7509                                        ;;**********************************************************************
 7510  033436  000004                TST35:  SCOPE
```

```
7511  033440  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
7512  033446  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7513
7514  033452  104401  056751                  TYPE    ,MSG22          ;FORMATTING FINISHED
7515  033456  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
7516  033462  001404                          BEQ     5$              ;BR IF NO
7517  033464  104401  056671                  TYPE    ,MSG21          ;BYP TESTS 36,40,41
7518  033470  000137  034514                  JMP     13$
7519  033474  013737  012100  001350  5$:     MOV     LC,FRCYL        ;FROM CYL
7520  033502  013737  012076  001352          MOV     LCM1,TOCYL      ;TO CYL
7521
7522  033510                          1$:
7523  033510  104415                          SCOP1
7524  033512  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7525
7526  033516  004737  045462                  JSR     PC,SUBCLR
7527  033522  104024                          ERROR   24              ;CERR AFTER SCLR
7528
7529  033524  012737  034454  001176          MOV     #10$,$ESCAPE
7530  033532  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
7531
7532  033540  012737  000017  003322          MOV     #SEEK,HCS1
7533  033546  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
7534  033552  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7535  033554  012737  030140  003412          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7536  033562  005037  003414                  CLR     E.B0
7537  033566  012737  005720  003416          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7538  033574  012737  000001  003420          MOV     #1,E.B1
7539
7540  033602  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7541  033606  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7542  033610  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7543  033612  104204                          ERROR   204             ;MSG B0 ERROR
7544  033614  104205                          ERROR   205             ;MSG A1 ERROR
7545  033616  104206                          ERROR   206             ;MSG B1 ERROR
7546
7547  033620  023727  001362  000001          CMP     CYLDIF,#1
7548  033626  001406                          BEQ     2$
7549  033630  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WD 2
7550  033636  004737  045132                  JSR     PC,GSTAT
7551  033642  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
7552
7553  033644  012737  034474  001176  2$:     MOV     #12$,$ESCAPE
7554  033652  012737  004704  003360          MOV     #2500.,TEMP1    ;SETUP TIMEOUT
7555
7556  033660  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
7557  033664  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7558  033666  032737  100000  003322          BIT     #CERR,HCS1
7559  033674  001401                          BEQ     64$
7560  033676  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7561  033700                          64$:
7562
7563  033700  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7564  033706  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
7565  033712  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7566  033720  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
```

```
 7567  033726  005037  003422                  CLR     E.A2                ;EXPECTED MSG A2
 7568  033732  012737  000002  003424          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
 7569  033740  012737  000003  003430          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
 7570
 7571  033746  004737  044310                  JSR     PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
 7572  033752  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
 7573  033754  104133                          ERROR   133                 ;MSG A0 ERROR AFTER SEEK CMD
 7574  033756  104134                          ERROR   134                 ;MSG B0 ERROR
 7575  033760  104135                          ERROR   135                 ;MSG A1 ERROR
 7576  033762  104136                          ERROR   136                 ;MSG B1 ERROR
 7577  033764  005737  001362                  TST     CYLDIF
 7578  033770  001401                          BEQ     65$
 7579  033772  104137                          ERROR   137                 ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 7580
 7581  033774                          65$:
 7582
 7583  033774  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 7584  034002  013765  001222  000010          MOV     $UNIT,RKCS2(R5)     ;DRIVE#
 7585  034010  012737  000005  003322          MOV     #CLEAR,HCS1
 7586  034016  004737  043472                  JSR     PC,DOCMD            ;DO DRIVE CLEAR CMD & GET CONTR RDY
 7587  034022  104151                          ERROR   151                 ;NO RDY AFTER DRIVE CLEAR CMD
 7588  034024  004737  044050                  JSR     PC,TSTATN           ;TEST FOR ATTN
 7589  034030  000401                          BR      66$
 7590  034032  104154                          ERROR   154                 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 7591  034034                          66$:
 7592
 7593  034034  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 7594  034042  005037  003414                  CLR     E.B0                ;EXPECTED MSG B0
 7595  034046  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
 7596  034054  012737  000001  003420          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
 7597  034062  005037  003422                  CLR     E.A2                ;EXPECTED MSG A2
 7598  034066  012737  000002  003424          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
 7599  034074  012737  000003  003430          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
 7600
 7601  034102  004737  044310                  JSR     PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
 7602  034106  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
 7603  034110  104273                          ERROR   273                 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 7604  034112  104265                          ERROR   265                 ;MSG B0 ERROR
 7605  034114  104274                          ERROR   274                 ;MSG A1 ERROR
 7606  034116  104266                          ERROR   266                 ;MSG B1 ERROR
 7607
 7608  034120  023737  001364  001352          CMP     CYLADD,TOCYL
 7609  034126  001401                          BEQ     3$
 7610  034130  104207                          ERROR   207                 ;CYL ADDR IN RKMR3 NOT = RKDC
 7611
 7612  034132                            3$:
 7613  034132  104415                          SCOP1
 7614  034134  012706  001100                  MOV     #STACK,SP           ;RESTORE STK PTR
 7615
 7616  034140  004737  045462                  JSR     PC,SUBCLR
 7617  034144  104024                          ERROR   24                  ;CERR AFTER SCLR
 7618
 7619  034146  005037  001176                  CLR     $ESCAPE
 7620
 7621  034152  013765  001352  000020          MOV     TOCYL,RKDC(R5)      ;CYL #
 7622
```

```
7623
7624  034160  012700  001674                MOV     #RHTAB,R0
7625  034164  012737  000025  003322        MOV     #<RDHEAD>,HCS1
7626  034172  004737  043530                JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
7627  034176  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
7628  034200  032737  100000  003322        BIT     #CERR,HCS1
7629  034206  001405                        BEQ     68$
7630  034210  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
7631  034212  104401  056436                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7632  034216  000137  042706                JMP     $EOP            ;ABORT DRIVE
7633
7634  034222  016520  000024        68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7635  034226  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
7636  034232  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
7637
7638
7639  034236  032765  100000  000010        BIT     #DLT,RKCS2(R5)
7640  034244  001407                        BEQ     69$
7641  034246  004737  045132                JSR     PC,GSTAT
7642  034252  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
7643  034254  104401  056436                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7644  034260  000137  042706                JMP     $EOP            ;ABORT DRIVE
7645  034264                        69$:
7646
7647  034264  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7648  034272  001401                        BEQ     67$             ;BR IF SAME
7649  034274  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
7650  034276                        67$:
7651
7652  034276  005337  001350                DEC     FRCYL
7653  034302  001404                        BEQ     4$
7654  034304  005337  001352                DEC     TOCYL
7655  034310  000137  033510                JMP     1$
7656
7657  034314                        4$:
7658  034314  004737  047466                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7659  034320  000475                        BR      TST36           ;;GO TO NEXT TEST
7660                                                                ;RETURN HERE IF SW 14 IS SET OR
7661                                                                ;SW 8 WITH SWR <7:0> APPLY
7662
7663
7664  034322                        6$:
7665
7666  034322  004737  045462                JSR     PC,SUBCLR
7667  034326  104024                        ERROR   24              ;CERR AFTER SCRL
7668
7669  034330  013765  001352  000020  70$:  MOV     TOCYL,RKDC(R5)  ;CYL#
7670
7671  034336  012737  000017  003322        MOV     #SEEK,HCS1
7672  034344  004737  043472                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
7673  034350  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
7674
7675  034352  013737  001426  003360        MOV     T50000,TEMP1
7676  034360  004737  044176                JSR     PC,FATT2        ;FIND ATTN
7677  034364  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
7678  034366  032737  100000  003322        BIT     #CERR,HCS1
```

```
7679  034374  001401                    BEQ     72$
7680  034376  104210                    ERROR   210             ;CERR AFTER SEEK CMD.
7681
7682  034400  004737  045462    72$:    JSR     PC,SUBCLR
7683  034404  104024                    ERROR   24              ;CERR AFTER SCLR
7684
7685  034406  023737  001352  012100    CMP     TOCYL,LC        ;LAST CYL DONE?
7686  034414  001403                    BEQ     71$             ;BR IF YES
7687  034416  005237  001352            INC     TOCYL           ;ELSE DO ANOTHER
7688  034422  000742                    BR      70$
7689
7690  034424  004737  045462    71$:    JSR     PC,SUBCLR
7691  034430  104024                    ERROR   24              ;CERR AFTER SCLR
7692
7693  034432  005037  001176            CLR     $ESCAPE
7694  034436  005737  001410            TST     LPFLG
7695  034442  001402                    BEQ     73$
7696  034444  000177  144440            JMP     @$LPERR         ;SW 9 WAS SET.
7697  034450  000177  144432    73$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7698
7699
7700
7701  034454                    10$:
7702  034454  005237  001410            INC     LPFLG
7703  034460  032777  001000  144452    BIT     #SW9,@SWR       ;LOOP ON ERROR?
7704  034466  001315                    BNE     6$              ;YES, RECONDITION DRIVE
7705  034470  000137  033644            JMP     2$              ;RETURN TO MAINLINE
7706
7707  034474                    12$:
7708  034474  005237  001410            INC     LPFLG
7709  034500  032777  001000  144432    BIT     #SW9,@SWR       ;LOOP ON ERROR?
7710  034506  001305                    BNE     6$              ;YES, RECONDITION DRIVE
7711  034510  000137  034132            JMP     3$              ;RETURN TO MAINLINE
7712
7713  034514                    13$:
7714
7715
7716                            ;;*****************************************************************
7717                            ;*TEST 36       SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS
7718                            ;*
7719                            ;*      THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLS & READS HEADERS.
7720                            ;*      IT THEN SEEKS CYL 0 & READS HEADERS.
7721                            ;*
7722                            ;*      MAJOR CYLS ARE: 1  (DECIMAL)  =  1 (OCTAL)
7723                            ;*                      2                2
7724                            ;*                      4                4
7725                            ;*                      8                10
7726                            ;*                      16               20
7727                            ;*                      32               40
7728                            ;*                      64               100
7729                            ;*                      128              200
7730                            ;*                      256              400
7731                            ;*                      512              1000 (RK07)
7732                            ;*
7733                            ;;*****************************************************************
7734  034514  000004            TST36:  SCOPE
```

```
 7735  034516  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 7736  034524  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 7737
 7738  034530  013737  012116  001350          MOV     FC,FRCYL        ;SETUP FROM CYL
 7739  034536  013737  012120  001352          MOV     FCP1,TOCYL      ;SETUP TO CYL
 7740
 7741  034544                          1$:
 7742  034544  104415                          SCOP1
 7743  034546  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 7744
 7745  034552  004737  045462                  JSR     PC,SUBCLR
 7746  034556  104024                          ERROR   24              ;CERR AFTER SCLR
 7747
 7748  034560  012737  036266  001176          MOV     #10$,$ESCAPE
 7749  034566  013737  001350  003364          MOV     FRCYL,TEMP3     ;SETUP
 7750  034574  013737  001352  003366          MOV     TOCYL,TEMP4     ;CYL DIFF
 7751  034602  163737  003364  003366          SUB     TEMP3,TEMP4     ;FOR
 7752  034610  013737  003366  001360          MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
 7753
 7754  034616  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
 7755
 7756  034624  012737  000017  003322          MOV     #SEEK,HCS1
 7757  034632  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 7758  034636  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 7759  034640  012737  030140  003412          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 7760  034646  005037  003414                  CLR     E.B0
 7761  034652  012737  003720  003416          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 7762  034660  012737  000001  003420          MOV     #1,E.B1
 7763
 7764  034666  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7765  034672  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7766  034674  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 7767  034676  104204                          ERROR   204             ;MSG B0 ERROR
 7768  034700  104205                          ERROR   205             ;MSG A1 ERROR
 7769  034702  104206                          ERROR   206             ;MSG B1 ERROR
 7770
 7771  034704  012737  036306  001176  2$:     MOV     #12$,$ESCAPE
 7772  034712  013737  001426  003360          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 7773
 7774  034720  004737  044176                  JSR     PC,FATT2        ;FIND A.TN
 7775  034724  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 7776  034726  032737  100000  003322          BIT     #CERR,HCS1
 7777  034734  001401                          BEQ     64$
 7778  034736  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 7779  034740                          64$:
 7780
 7781  034740  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 7782  034746  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 7783  034752  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 7784  034760  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 7785  034766  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 7786  034772  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 7787  035000  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 7788
 7789  035006  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7790  035012  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
```

```
 7791  035014  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 7792  035016  104134                          ERROR   134             ;MSG B0 ERROR
 7793  035020  104135                          ERROR   135             ;MSG A1 ERROR
 7794  035022  104136                          ERROR   136             ;MSG B1 ERROR
 7795  035024  005737  001362                  TST     CYLDIF
 7796  035030  001401                          BEQ     65$
 7797  035032  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 7798
 7799  035034                          65$:
 7800
 7801  035034  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 7802  035042  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
 7803  035050  012737  000005  003322        MOV     #CLEAR,HCS1
 7804  035056  004737  043472                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 7805  035062  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 7806  035064  004737  044050                JSR     PC,TSTATN       ;TEST FOR ATTN
 7807  035070  000401                          BR      66$
 7808  035072  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 7809  035074                          66$:
 7810
 7811  035074  012737  010340  003412        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 7812  035102  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
 7813  035106  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
 7814  035114  012737  000001  003420        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 7815  035122  005037  003422                CLR     E.A2            ;EXPECTED MSG A2
 7816  035126  012737  000002  003424        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 7817  035134  012737  000003  003430        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 7818
 7819  035142  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7820  035146  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7821  035150  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 7822  035152  104265                          ERROR   265             ;MSG B0 ERROR
 7823  035154  104274                          ERROR   274             ;MSG A1 ERROR
 7824  035156  104266                          ERROR   266             ;MSG B1 ERROR
 7825
 7826  035160  023737  001364  001352        CMP     CYLADD,TOCYL
 7827  035166  001401                          BEQ     3$
 7828  035170  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
 7829
 7830  035172                          3$:
 7831  035172  104415                          SCOP1
 7832  035174  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 7833
 7834  035200  004737  045462                JSR     PC,SUBCLR
 7835  035204  104024                          ERROR   24              ;CERR AFTER SCLR
 7836
 7837  035206  005037  001176                CLR     $ESCAPE
 7838  035212  013765  001352  000020        MOV     TOCYL,RKDC(R5)  ;CYL #
 7839
 7840
 7841  035220  012700  001674                MOV     #RHTAB,R0
 7842  035224  012737  000025  003322        MOV     #<RDHEAD>,HCS1
 7843  035232  004737  043530                JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
 7844  035236  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
 7845  035240  032737  100000  003322        BIT     #CERR,HCS1
 7846  035246  001405                          BEQ     68$
```

```
 7847  035250  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
 7848  035252  104401  056436                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 7849  035256  000137  042706                JMP     $EOP            ;ABORT DRIVE
 7850
 7851  035262  016520  000024        68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 7852  035266  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
 7853  035272  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
 7854
 7855
 7856  035276  032765  100000  000010        BIT     #DLT,RKCS2(R5)
 7857  035304  001407                        BEQ     69$
 7858  035306  004737  045132                JSR     PC,GSTAT
 7859  035312  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
 7860  035314  104401  056436                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 7861  035320  000137  042706                JMP     $EOP            ;ABORT DRIVE
 7862  035324                        69$:
 7863
 7864  035324  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
 7865  035332  001401                        BEQ     67$             ;BR IF SAME
 7866  035334  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
 7867  035336                        67$:
 7868
 7869
 7870  035336  104415                        SCOP1
 7871  035340  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 7872
 7873  035344  004737  045462                JSR     PC,SUBCLR
 7874  035350  104024                        ERROR   24              ;CERR AFTER SCLR
 7875
 7876  035352  012737  036326  001176        MOV     #14$,$ESCAPE
 7877  035360  013765  001350  000020        MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
 7878  035366  013737  001350  001354        MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROUTINE
 7879
 7880  035374  012737  000017  003322        MOV     #SEEK,HCS1
 7881  035402  004737  043472                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 7882  035406  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
 7883  035410  012737  030140  003412        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 7884  035416  005037  003414                CLR     E.B0
 7885  035422  012737  005720  003416        MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 7886  035430  012737  000001  003420        MOV     #1,E.B1
 7887
 7888  035436  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 7889  035442  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7890  035444  104203                        ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 7891  035446  104204                        ERROR   204             ;MSG B0 ERROR
 7892  035450  104205                        ERROR   205             ;MSG A1 ERROR
 7893  035452  104206                        ERROR   206             ;MSG B1 ERROR
 7894
 7895
 7896  035454  012737  036346  001176  4$:   MOV     #16$,$ESCAPE
 7897  035462  013737  001426  003360        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 7898
 7899  035470  004737  044176                JSR     PC,FATT2        ;FIND ATTN
 7900  035474  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
 7901  035476  032737  100000  003322        BIT     #CERR,HCS1
 7902  035504  001401                        BEQ     70$
```

```
7903  035506  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7904  035510                          70$:
7905
7906  035510  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7907  035516  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
7908  035522  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7909  035530  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7910  035536  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
7911  035542  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7912  035550  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7913
7914  035556  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7915  035562  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7916  035564  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7917  035566  104134                          ERROR   134             ;MSG B0 ERROR
7918  035570  104135                          ERROR   135             ;MSG A1 ERROR
7919  035572  104136                          ERROR   136             ;MSG B1 ERROR
7920  035574  005737  001362                  TST     CYLDIF
7921  035600  001401                          BEQ     71$
7922  035602  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7923
7924  035604                          71$:
7925
7926  035604  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7927  035612  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7928  035620  012737  000005  003322          MOV     #CLEAR,HCS1
7929  035626  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
7930  035632  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7931  035634  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
7932  035640  000401                          BR      72$
7933  035642  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7934  035644                          72$:
7935
7936  035644  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
7937  035652  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
7938  035656  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7939  035664  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7940  035672  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
7941  035676  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7942  035704  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7943
7944  035712  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7945  035716  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7946  035720  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7947  035722  104265                          ERROR   265             ;MSG B0 ERROR
7948  035724  104274                          ERROR   274             ;MSG A1 ERROR
7949  035726  104266                          ERROR   266             ;MSG B1 ERROR
7950
7951  035730  023737  001364  001350          CMP     CYLADD,FRCYL
7952  035736  001401                          BEQ     5$
7953  035740  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7954
7955  035742                          5$:
7956  035742  104415                          SCOP1
7957  035744  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7958
```

```
7959  035750  004737  045462              JSR     PC,SUBCLR
7960  035754  104024                      ERROR   24                  ;CERR AFTER SCLR
7961
7962  035756  005037  001176              CLR     $ESCAPE
7963  035762  013765  001350  000020      MOV     FRCYL,RKDC(R5)      ;CYL #
7964
7965
7966  035770  012700  001674              MOV     #RHTAB,R0
7967  035774  012737  000025  003322      MOV     #<RDHEAD>,HCS1
7968  036002  004737  043530              JSR     PC,DATCMD           ;DO DATA XFER CMD & GET CONTR RDY
7969  036006  104171                      ERROR   171                 ;NO RDY AFTER READ HEADER CMD
7970  036010  032737  100000  003322      BIT     #CERR,HCS1
7971  036016  001405                      BEQ     74$
7972  036020  104174                      ERROR   174                 ;CERR AFTER READ HEADER CMD
7973  036022  104401  056436              TYPE    ,MSG18              ;ABORT BALANCE OF TESTS
7974  036026  000137  042706              JMP     $EOP                ;ABORT DRIVE
7975
7976  036032  016520  000024      74$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
7977  036036  016520  000024              MOV     RKDB(R5),(R0)+      ;2'ND WORD
7978  036042  016520  000024              MOV     RKDB(R5),(R0)+      ;3'RD WORD
7979
7980
7981  036046  032765  100000  000010      BIT     #DLT,RKCS2(R5)
7982  036054  001407                      BEQ     75$
7983  036056  004737  045132              JSR     PC,GSTAT
7984  036062  104173                      ERROR   173                 ;DLT AFTER READ HEADER CMD
7985  036064  104401  056436              TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
7986  036070  000137  042706              JMP     $EOP                ;ABORT DRIVE
7987  036074                      75$:
7988
7989  036074  023737  001674  001350      CMP     RHTAB,FRCYL         ;CHECK WORD 0 (CYL#) ONLY
7990  036102  001401                      BEQ     73$                 ;BR IF SAME
7991  036104  104311                      ERROR   311                 ;READ CYL WORD HEADER ERROR
7992  036106                      73$:
7993
7994
7995  036106  023737  001352  012104      CMP     TOCYL,MC            ;ALL CYL DONE?
7996  036114  001404                      BEQ     6$                  ;BR IF YES
7997  036116  006337  001352              ASL     TOCYL               ;ELSE DO ANOTHER
7998  036122  000137  034544              JMP     1$
7999  036126                      6$:
8000  036126  004737  047466              JSR     PC,SWTST            ;SEE IF SW 14 OR 8 IS SET
8001  036132  000515                      BR      TST37               ;;GO TO NEXT TEST
8002                                              ;RETURN HERE IF SW 14 IS SET OR
8003                                              ;SW 8 WITH SWR <7:0> APPLY
8004  036134                      8$:
8005
8006  036134  004737  045462              JSR     PC,SUBCLR
8007  036140  104024                      ERROR   24                  ;CERR AFTER SCRL
8008
8009  036142  013765  001352  000020  76$:MOV     TOCYL,RKDC(R5)      ;CYL#
8010
8011  036150  012737  000017  003322      MOV     #SEEK,HCS1
8012  036156  004737  043472              JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
8013  036162  104131                      ERROR   131                 ;NO RDY AFTER SEEK CMD.
8014
```

```
 8015  036164  013737  001426  003360          MOV     T50000,TEMP1
 8016  036172  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 8017  036176  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 8018  036200  032737  100000  003322          BIT     #CERR,HCS1
 8019  036206  001401                          BEQ     78$
 8020  036210  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
 8021
 8022  036212  004737  045462          78$:    JSR     PC,SUBCLR
 8023  036216  104024                          ERROR   24              ;CERR AFTER SCLR
 8024
 8025  036220  023737  001352  012116          CMP     TOCYL,FC        ;LAST CYL DONE?
 8026  036226  001403                          BEQ     77$             ;BR IF YES
 8027  036230  005337  001352                  DEC     TOCYL           ;ELSE DO ANOTHER
 8028  036234  000742                          BR      76$
 8029
 8030  036236  004737  045462          77$:    JSR     PC,SUBCLR
 8031  036242  104024                          ERROR   24              ;CERR AFTER SCLR
 8032
 8033  036244  005037  001176                  CLR     $ESCAPE
 8034  036250  005737  001410                  TST     LPFLG
 8035  036254  001402                          BEQ     79$
 8036  036256  000177  142626                  JMP     @$LPERR         ;SW 9 WAS SET.
 8037  036262  000177  142620          79$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 8038
 8039  036266                          10$:
 8040  036266  005237  001410                  INC     LPFLG
 8041  036272  032777  001000  142640          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8042  036300  001315                          BNE     8$              ;YES, RECONDITION DRIVE
 8043  036302  000137  034704                  JMP     2$              ;RETURN TO MAINLINE
 8044  036306                          12$:
 8045  036306  005237  001410                  INC     LPFLG
 8046  036312  032777  001000  142620          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8047  036320  001305                          BNE     8$              ;YES, RECONDITION DRIVE
 8048  036322  000137  035172                  JMP     3$              ;RETURN TO MAINLINE
 8049  036326                          14$:
 8050  036326  005237  001410                  INC     LPFLG
 8051  036332  032777  001000  142600          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8052  036340  001275                          BNE     8$              ;YES, RECONDITION DRIVE
 8053  036342  000137  035454                  JMP     4$              ;RETURN TO MAINLINE
 8054  036346                          16$:
 8055  036346  005237  001410                  INC     LPFLG
 8056  036352  032777  001000  142560          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8057  036360  001265                          BNE     8$              ;YES, RECONDITION DRIVE
 8058  036362  000137  035742                  JMP     5$              ;RETURN TO MAINLINE
 8059
 8060
 8061
 8062                                  ;;*******************************************************************
 8063                                  ;*TEST 37        SEEK TO ALL CYLS FROM 0 & READ HEADERS
 8064                                  ;;*******************************************************************
 8065  036366  000004                  TST37:  SCOPE
 8066  036370  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 8067  036376  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 8068
 8069  036402  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
 8070  036406  001402                          BEQ     DOSEEK          ;BR IF NO
```

```
8071  036410  000137  042210                    JMP     CYLINV          ;ELSE BYPASS TESTS 40 & 41
8072  036414                            DOSEEK:
8073
8074  036414  013737  012116  001350            MOV     FC,FRCYL        ;SETUP FROM CYL
8075  036422  013737  012120  001352            MOV     FCP1,TOCYL      ;SETUP TO CYL
8076
8077  036430                            1$:
8078  036430  104415                            SCOP1
8079  036432  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
8080
8081  036436  004737  045462                    JSR     PC,SUBCLR
8082  036442  104024                            ERROR   24              ;CERR AFTER SCLR
8083
8084  036444  012737  040152  001176            MOV     #10$,$ESCAPE
8085  036452  013737  001350  003364            MOV     FRCYL,TEMP3     ;SETUP
8086  036460  013737  001352  003366            MOV     TOCYL,TEMP4     ;CYL DIFF
8087  036466  163737  003364  003366            SUB     TEMP3,TEMP4     ;FOR
8088  036474  013737  003366  001360            MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
8089
8090  036502  013765  001352  000020            MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
8091
8092  036510  012737  000017  003322            MOV     #SEEK,HCS1
8093  036516  004737  043472                    JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
8094  036522  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD
8095  036524  012737  030140  003412            MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
8096  036532  005037  003414                    CLR     E.B0
8097  036536  012737  003720  003416            MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8098  036544  012737  000001  003420            MOV     #1,E.B1
8099
8100  036552  004737  044310                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8101  036556  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8102  036560  104203                            ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
8103  036562  104204                            ERROR   204             ;MSG B0 ERROR
8104  036564  104205                            ERROR   205             ;MSG A1 ERROR
8105  036566  104206                            ERROR   206             ;MSG B1 ERROR
8106
8107  036570  012737  040172  001176  2$:       MOV     #12$,$ESCAPE
8108  036576  013737  001426  003360            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
8109
8110  036604  004737  044176                    JSR     PC,FATT2        ;FIND ATTN
8111  036610  104132                            ERROR   132             ;NO ATTN AFTER SEEK CMD
8112  036612  032737  100000  003322            BIT     #CERR,HCS1
8113  036620  001401                            BEQ     64$
8114  036622  104210                            ERROR   210             ;CERR AFTER SEEK CMD
8115  036624                            64$:
8116
8117  036624  012737  050340  003412            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8118  036632  005037  003414                    CLR     E.B0            ;EXPECTED MSG B0
8119  036636  012737  001720  003416            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8120  036644  012737  000001  003420            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8121  036652  005037  003422                    CLR     E.A2            ;EXPECTED MSG A2
8122  036656  012737  000002  003424            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8123  036664  012737  000003  003430            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8124
8125  036672  004737  044310                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8126  036676  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
```

```
 8127  036700  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 8128  036702  104134                        ERROR   134             ;MSG B0 ERROR
 8129  036704  104135                        ERROR   135             ;MSG A1 ERROR
 8130  036706  104136                        ERROR   136             ;MSG B1 ERROR
 8131  036710  005737  001362                TST     CYLDIF
 8132  036714  001401                        BEQ     65$
 8133  036716  104137                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 8134
 8135  036720                        65$:
 8136
 8137  036720  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 8138  036726  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
 8139  036734  012737  000005  003322        MOV     #CLEAR,HCS1
 8140  036742  004737  043472                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 8141  036746  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 8142  036750  004737  044050                JSR     PC,TSTATN       ;TEST FOR ATTN
 8143  036754  000401                        BR      66$
 8144  036756  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 8145  036760                        66$:
 8146
 8147  036760  012737  010340  003412        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
 8148  036766  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
 8149  036772  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 8150  037000  012737  000001  003420        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 8151  037006  005037  003422                CLR     E.A2            ;EXPECTED MSG A2
 8152  037012  012737  000002  003424        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 8153  037020  012737  000003  003430        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 8154
 8155  037026  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8156  037032  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8157  037034  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 8158  037036  104265                        ERROR   265             ;MSG B0 ERROR
 8159  037040  104274                        ERROR   274             ;MSG A1 ERROR
 8160  037042  104266                        ERROR   266             ;MSG B1 ERROR
 8161
 8162  037044  023737  001364  001352        CMP     CYLADD,TOCYL
 8163  037052  001401                        BEQ     3$
 8164  037054  104207                        ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
 8165
 8166  037056                          3$:
 8167  037056  104415                        SCOP1
 8168  037060  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 8169
 8170  037064  004737  045462                JSR     PC,SUBCLR
 8171  037070  104024                        ERROR   24              ;CERR AFTER SCLR
 8172
 8173  037072  005037  001176                CLR     $ESCAPE
 8174  037076  013765  001352  00002C        MOV     TOCYL,RKDC(R5)  ;CYL #
 8175
 8176
 8177  037104  012700  001674                MOV     #RHTAB,R0
 8178  037110  012737  000025  003322        MOV     #<RDHEAD>,HCS1
 8179  037116  004737  043530                JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
 8180  037122  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
 8181  037124  032737  100000  003322        BIT     #CERR,HCS1
 8182  037132  001405                        BEQ     68$
```

```
8183  037134  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
8184  037136  104401  056436                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
8185  037142  000137  042706                JMP     $EOP            ;ABORT DRIVE
8186
8187  037146  016520  000024        68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
8188  037152  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
8189  037156  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
8190
8191
8192  037162  032765  100000  000010        BIT     #DLT,RKCS2(R5)
8193  037170  001407                        BEQ     69$
8194  037172  004737  045132                JSR     PC,GSTAT
8195  037176  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
8196  037200  104401  056436                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
8197  037204  000137  042706                JMP     $EOP            ;ABORT DRIVE
8198  037210                        69$:
8199
8200  037210  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
8201  037216  001401                        BEQ     67$             ;BR IF SAME
8202  037220  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
8203  037222                        67$:
8204
8205
8206  037222  104415                        SCOP1
8207  037224  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
8208
8209  037230  004737  045462                JSR     PC,SUBCLR
8210  037234  104024                        ERROR   24              ;CERR AFTER SCLR
8211
8212  037236  012737  040212  001176        MOV     #14$,$ESCAPE
8213  037244  013765  001350  000020        MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
8214  037252  013737  001350  001354        MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROU"INE
8215
8216  037260  012737  000017  003322        MOV     #SEEK,HCS1
8217  037266  004737  043472                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
8218  037272  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
8219  037274  012737  030140  003412        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
8220  037302  005037  003414                CLR     E.B0
8221  037306  012737  005720  003416        MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8222  037314  012737  000001  003420        MOV     #1,E.B1
8223
8224  037322  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8225  037326  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8226  037330  104203                        ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
8227  037332  104204                        ERROR   204             ;MSG B0 ERROR
8228  037334  104205                        ERROR   205             ;MSG A1 ERROR
8229  037336  104206                        ERROR   206             ;MSG B1 ERROR
8230
8231
8232  037340  012737  040232  001176  4$:   MOV     #16$,$ESCAPE
8233  037346  013737  001426  003360        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
8234
8235  037354  004737  044176                JSR     PC,FATT2        ;FIND ATTN
8236  037360  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
8237  037362  032737  100000  003322        BIT     #CERR,HCS1
8238  037370  001401                        BEQ     70$
```

```
8239  037372  104210                          ERROR   210             ;CERR AFTER SEEK CMD
8240  037374                          70$:
8241
8242  037374  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8243  037402  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
8244  037406  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8245  037414  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8246  037422  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
8247  037426  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8248  037434  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8249
8250  037442  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8251  037446  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8252  037450  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
8253  037452  104134                          ERROR   134             ;MSG B0 ERROR
8254  037454  104135                          ERROR   135             ;MSG A1 ERROR
8255  037456  104136                          ERROR   136             ;MSG B1 ERROR
8256  037460  005737  001362                  TST     CYLDIF
8257  037464  001401                          BEQ     71$
8258  037466  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8259
8260  037470                          71$:
8261
8262  037470  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
8263  037476  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
8264  037504  012737  000005  003322          MOV     #CLEAR,HCS1
8265  037512  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
8266  037516  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8267  037520  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
8268  037524  000401                          BR      72$
8269  037526  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8270  037530                          72$:
8271
8272  037530  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
8273  037536  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
8274  037542  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8275  037550  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8276  037556  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
8277  037562  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8278  037570  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8279
8280  037576  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8281  037602  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8282  037604  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8283  037606  104265                          ERROR   265             ;MSG B0 ERROR
8284  037610  104274                          ERROR   274             ;MSG A1 ERROR
8285  037612  104266                          ERROR   266             ;MSG B1 ERROR
8286
8287  037614  023737  001364  001350          CMP     CYLADD,FRCYL
8288  037622  001401                          BEQ     5$
8289  037624  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
8290
8291  037626                          5$:
8292  037626  104415                          SCOP1
8293  037630  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
8294
```

```
8295  037634  004737  045462                  JSR     PC,SUBCLR
8296  037640  104024                          ERROR   24              ;CERR AFTER SCLR
8297
8298  037642  005037  001176                  CLR     $ESCAPE
8299  037646  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;CYL #
8300
8301
8302  037654  012700  001674                  MOV     #RHTAB,R0
8303  037660  012737  000025  003322          MOV     #<RDHEAD>,HCS1
8304  037666  004737  043530                  JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
8305  037672  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
8306  037674  032737  100000  003322          BIT     #CERR,HCS1
8307  037702  001405                          BEQ     74$
8308  037704  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
8309  037706  104401  056436                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
8310  037712  000137  042706                  JMP     $EOP            ;ABORT DRIVE
8311
8312  037716  016520  000024          74$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
8313  037722  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
8314  037726  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
8315
8316
8317  037732  032765  100000  000010          BIT     #DLT,RKCS2(R5)
8318  037740  001407                          BEQ     75$
8319  037742  004737  045132                  JSR     PC,GSTAT
8320  037746  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
8321  037750  104401  056436                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
8322  037754  000137  042706                  JMP     $EOP            ;ABORT DRIVE
8323  037760                          75$:
8324
8325  037760  023737  001674  001350          CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
8326  037766  001401                          BEQ     73$             ;BR IF SAME
8327  037770  104311                          ERROR   311             ;READ CYL WORD HEADER ERROR
8328  037772                          73$:
8329
8330
8331  037772  023737  001352  012100          CMP     TOCYL,LC        ;ALL CYL DONE?
8332  040000  001404                          BEQ     6$              ;BR IF YES
8333  040002  005237  001352                  INC     TOCYL           ;ELSE DO ANOTHER
8334  040006  000137  036430                  JMP     1$
8335  040012                          6$:
8336  040012  004737  047466                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
8337  040016  000515                          BR      TST40           ;;GO TO NEXT TEST
8338                                                                  ;RETURN HERE IF SW 14 IS SET OR
8339                                                                  ;SW 8 WITH SWR <7:0> APPLY
8340  040020                          8$:
8341
8342  040020  004737  045462                  JSR     PC,SUBCLR
8343  040024  104024                          ERROR   24              ;CERR AFTER SCRL
8344
8345  040026  013765  001352  000020  76$:    MOV     TOCYL,RKDC(R5)  ;CYL#
8346
8347  040034  012737  000017  003322          MOV     #SEEK,HCS1
8348  040042  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
8349  040046  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
8350
```

```
040050  013737  001426  003360           MOV     T50000,TEMP1
040056  004737  044176                   JSR     PC,FATT2           ;FIND ATTN
040062  104132                           ERROR   132                ;NO ATTN AFTER SEEK CMD
040064  032737  100000  003322           BIT     #CERR,HCS1
040072  001401                           BEQ     78$
040074  104210                           ERROR   210                ;CERR AFTER SEEK CMD.

040076  004737  045462          78$:     JSR     PC,SUBCLR
040102  104024                           ERROR   24                 ;CERR AFTER SCLR

040104  023737  001352  012116           CMP     TOCYL,FC           ;LAST CYL DONE?
040112  001403                           BEQ     77$                ;BR IF YES
040114  005337  001352                   DEC     TOCYL              ;ELSE DO ANOTHER
040120  000742                           BR      76$

040122  004737  045462          77$:     JSR     PC,SUBCLR
040126  104024                           ERROR   24                 ;CERR AFTER SCLR

040130  005037  001176                   CLR     $ESCAPE
040134  005737  001410                   TST     LPFLG
040140  001402                           BEQ     79$
040142  000177  140742                   JMP     @$LPERR            ;SW 9 WAS SET.
040146  000177  140734          79$:     JMP     @$LPADR            ;SW 14 OR 8 WAS SET

040152                          10$:
040152  005237  001410                   INC     LPFLG
040156  032777  001000  140754           BIT     #SW9,@SWR          ;LOOP ON ERROR?
040164  001315                           BNE     8$                 ;YES, RECONDITION DRIVE
040166  000137  036570                   JMP     2$                 ;RETURN TO MAINLINE
040172                          12$:
040172  005237  001410                   INC     LPFLG
040176  032777  001000  140734           BIT     #SW9,@SWR          ;LOOP ON ERROR?
040204  001305                           BNE     8$                 ;YES, RECONDITION DRIVE
040206  000137  037056                   JMP     3$                 ;RETURN TO MAINLINE
040212                          14$:
040212  005237  001410                   INC     LPFLG
040216  032777  001000  140714           BIT     #SW9,@SWR          ;LOOP ON ERROR?
040224  001275                           BNE     8$                 ;YES, RECONDITION DRIVE
040226  000137  037340                   JMP     4$                 ;RETURN TO MAINLINE
040232                          16$:
040232  005237  001410                   INC     LPFLG
040236  032777  001000  140674           BIT     #SW9,@SWR          ;LOOP ON ERROR?
040244  001265                           BNE     8$                 ;YES, RECONDITION DRIVE
040246  000137  037626                   JMP     5$                 ;RETURN TO MAINLINE

                                ;;****  *************************************************************
                                ;*TEST 40        SEEK TO ALL CYLS FROM LAST CYL & READ HEADERS
                                ;;*******************************************************************
040252  000004                  TST40:  SCOPE
040254  012737  000001  001174           MOV     #1,$TIMES          ;;DO 1 ITERATION
040262  012706  001100                   MOV     #STACK,SP          ;RESTORE STK PTR


040266  004737  045462                   JSR     PC,SUBCLR
040272  104024                           ERROR   24                 ;CERR AFTER SCLR

```

```
 8407  040274  013765  012100  000020          MOV     LC,RKDC(R5)     ;QUICK SEEK TO LAST CYL
 8408
 8409  040302  012737  000017  003322          MOV     #SEEK,HCS1
 8410  040310  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 8411  040314  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
 8412
 8413  040316  013737  001426  003360          MOV     T50000,TEMP1
 8414  040324  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 8415  040330  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 8416  040332  032737  100000  003322          BIT     #CERR,HCS1
 8417  040340  001401                          BEQ     64$
 8418  040342  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
 8419
 8420  040344  004737  045462          64$:    JSR     PC,SUBCLR
 8421  040350  104024                          ERROR   24              ;CERR AFTER SCLR
 8422
 8423
 8424  040352  013737  012100  001350          MOV     LC,FRCYL        ;SETUP FROM CYL
 8425  040360  013737  012076  001352          MOV     LCM1,TOCYL      ;SETUP TO CYL
 8426
 8427  040366                          1$:
 8428  040366  104415                          SCOP1
 8429  040370  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 8430
 8431  040374  004737  045462                  JSR     PC,SUBCLR
 8432  040400  104024                          ERROR   24              ;CERR AFTER SCLR
 8433
 8434  040402  012737  042110  001176          MOV     #10$,$ESCAPE
 8435  040410  013737  001350  003364          MOV     FRCYL,TEMP3     ;SETUP
 8436  040416  013737  001352  003366          MOV     TOCYL,TEMP4     ;CYL DIFF
 8437  040424  163737  003366  003364          SUB     TEMP4,TEMP3     ;FOR
 8438  040432  013737  003364  001360          MOV     TEMP3,CALDIF    ;ERROR PRINTOUT
 8439
 8440  040440  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
 8441
 8442  040446  012737  000017  003322          MOV     #SEEK,HCS1
 8443  040454  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 8444  040460  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 8445  040462  012737  030140  003412          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 8446  040470  005037  003414                  CLR     E.B0
 8447  040474  012737  005720  003416          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 8448  040502  012737  000001  003420          MOV     #1,E.B1
 8449
 8450  040510  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8451  040514  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8452  040516  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 8453  040520  104204                          ERROR   204             ;MSG B0 ERROR
 8454  040522  104205                          ERROR   205             ;MSG A1 ERROR
 8455  040524  104206                          ERROR   206             ;MSG B1 ERROR
 8456
 8457  040526  012737  042130  001176  2$:     MOV     #12$,$ESCAPE
 8458  040534  013737  001426  003360          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 8459
 8460  040542  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 8461  040546  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 8462  040550  032737  100000  003322          BIT     #CERR,HCS1
```

```
8463  040556  001401                        BEQ     65$
8464  040560  104210                        ERROR   210             ;CERR AFTER SEEK CMD
8465  040562                        65$:
8466
8467  040562  012737  050340  003412        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8468  040570  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
8469  040574  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8470  040602  012737  000001  003420        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8471  040610  005037  003422                CLR     E.A2            ;EXPECTED MSG A2
8472  040614  012737  000002  003424        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8473  040622  012737  000003  003430        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8474
8475  040630  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8476  040634  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8477  040636  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
8478  040640  104134                        ERROR   134             ;MSG B0 ERROR
8479  040642  104135                        ERROR   135             ;MSG A1 ERROR
8480  040644  104136                        ERROR   136             ;MSG B1 ERROR
8481  040646  005737  001362                TST     CYLDIF
8482  040652  001401                        BEQ     66$
8483  040654  104137                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8484
8485  040656                        66$:
8486
8487  040656  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
8488  040664  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
8489  040672  012737  000005  003322        MOV     #CLEAR,HCS1
8490  040700  004737  043472                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
8491  040704  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8492  040706  004737  044050                JSR     PC,TSTATN       ;TEST FOR ATTN
8493  040712  000401                        BR      67$
8494  040714  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8495  040716                        67$:
8496
8497  040716  012737  010340  003412        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
8498  040724  005037  003414                CLR     E.B0            ;EXPECTED MSG B0
8499  040730  012737  001720  003416        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8500  040736  012737  000001  003420        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8501  040744  005037  003422                CLR     E.A2            ;EXPECTED MSG A2
8502  040750  012737  000002  003424        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8503  040756  012737  000003  003430        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8504
8505  040764  004737  044310                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8506  040770  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8507  040772  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8508  040774  104265                        ERROR   265             ;MSG B0 ERROR
8509  040776  104274                        ERROR   274             ;MSG A1 ERROR
8510  041000  104266                        ERROR   266             ;MSG B1 ERROR
8511
8512  041002  023737  001364  001352        CMP     CYLADD,TOCYL
8513  041010  001401                        BEQ     3$
8514  041012  104207                        ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
8515
8516  041014                        3$:
8517  041014  104415                        SCOP1
8518  041016  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
```

```
 8519
 8520  041022  004737  045462                JSR     PC,SUBCLR
 8521  041026  104024                        ERROR   24                  ;CERR AFTER SCLR
 8522
 8523  041030  005037  001176                CLR     $ESCAPE
 8524  041034  013765  001352  000020        MOV     TOCYL,RKDC(R5)      ;CYL #
 8525
 8526
 8527  041042  012700  001674                MOV     #RHTAB,R0
 8528  041046  012737  000025  003322        MOV     #<RDHEAD>,HCS1
 8529  041054  004737  043530                JSR     PC,DATCMD           ;DO DATA XFER CMD & GET CONTR RDY
 8530  041060  104171                        ERROR   171                 ;NO RDY AFTER READ HEADER CMD
 8531  041062  032737  100000  003322        BIT     #CERR,HCS1
 8532  041070  001405                        BEQ     69$
 8533  041072  104174                        ERROR   174                 ;CERR AFTER READ HEADER CMD
 8534  041074  104401  056436                TYPE    ,MSG18              ;ABORT BALANCE OF TESTS
 8535  041100  000137  042706                JMP     $EOP                ;ABORT DRIVE
 8536
 8537  041104  016520  000024        69$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
 8538  041110  016520  000024                MOV     RKDB(R5),(R0)+      ;2'ND WORD
 8539  041114  016520  000024                MOV     RKDB(R5),(R0)+      ;3'RD WORD
 8540
 8541
 8542  041120  032765  100000  000010        BIT     #DLT,RKCS2(R5)
 8543  041126  001407                        BEQ     70$
 8544  041130  004737  045132                JSR     PC,GSTAT
 8545  041134  104173                        ERROR   173                 ;DLT AFTER READ HEADER CMD
 8546  041136  104401  056436                TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
 8547  041142  000137  042706                JMP     $EOP                ;ABORT DRIVE
 8548  041146                        70$:
 8549
 8550  041146  023737  001674  001352        CMP     RHTAB,TOCYL         ;CHECK WORD 0 (CYL#) ONLY
 8551  041154  001401                        BEQ     68$                 ;BR IF SAME
 8552  041156  104310                        ERROR   310                 ;READ CYL WORD HEADER ERROR
 8553  041160                        68$:
 8554
 8555
 8556  041160  104415                        SCOP1
 8557  041162  012706  001100                MOV     #STACK,SP           ;RESTORE STK PTR
 8558
 8559  041166  004737  045462                JSR     PC,SUBCLR
 8560  041172  104024                        ERROR   24                  ;CERR AFTER SCLR
 8561
 8562  041174  012737  042150  001176        MOV     #14$,$ESCAPE
 8563  041202  013765  001350  000020        MOV     FRCYL,RKDC(R5)      ;RETURN TO CYL #
 8564  041210  013737  001350  001354        MOV     FRCYL,CCYL          ;CURRENT CYL FOR TRUERROR ROUTINE
 8565
 8566  041216  012737  000017  003322        MOV     #SEEK,HCS1
 8567  041224  004737  043472                JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
 8568  041230  104131                        ERROR   131                 ;NO RDY AFTER SEEK CMD
 8569  041232  012737  030140  003412        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 8570  041240  005037  003414                CLR     E.B0
 8571  041244  012737  003720  003416        MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 8572  041252  012737  000001  003420        MOV     #1,E.B1
 8573
 8574  041260  004737  044310                JSR     PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
```

```
 8575  041264  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8576  041266  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 8577  041270  104204                          ERROR   204             ;MSG B0 ERROR
 8578  041272  104205                          ERROR   205             ;MSG A1 ERROR
 8579  041274  104206                          ERROR   206             ;MSG B1 ERROR
 8580
 8581
 8582  041276  012737  042170  001176  4$:     MOV     #16$,$ESCAPE
 8583  041304  013737  001426  003360          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 8584
 8585  041312  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
 8586  041316  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 8587  041320  032737  100000  003322          BIT     #CERR,HCS1
 8588  041326  001401                          BEQ     71$
 8589  041330  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 8590  041332                          71$:
 8591
 8592  041332  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 8593  041340  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 8594  041344  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 8595  041352  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 8596  041360  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 8597  041364  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 8598  041372  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 8599
 8600  041400  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8601  041404  000003                          .WORD   T.A2!1.B2!0             ;& MSGS SPECIFIED HERE
 8602  041406  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 8603  041410  104134                          ERROR   134             ;MSG B0 ERROR
 8604  041412  104135                          ERROR   135             ;MSG A1 ERROR
 8605  041414  104136                          ERROR   136             ;MSG B1 ERROR
 8606  041416  005737  001362                  TST     CYLDIF
 8607  041422  001401                          BEQ     72$
 8608  041424  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 8609
 8610  041426                          72$:
 8611
 8612  041426  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8613  041434  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 8614  041442  012737  000005  003322          MOV     #CLEAR,HCS1
 8615  041450  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 8616  041454  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 8617  041456  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 8618  041462  000401                          BR      73$
 8619  041464  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 8620  041466                          73$:
 8621
 8622  041466  012737  010340  003412          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 8623  041474  005037  003414                  CLR     E.B0            ;EXPECTED MSG B0
 8624  041500  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 8625  041506  012737  000001  003420          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 8626  041514  005037  003422                  CLR     E.A2            ;EXPECTED MSG A2
 8627  041520  012737  000002  003424          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 8628  041526  012737  000003  003430          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 8629
 8630  041534  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
```

```
8631  041540  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8632  041542  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8633  041544  104265                        ERROR   265             ;MSG B0 ERROR
8634  041546  104274                        ERROR   274             ;MSG A1 ERROR
8635  041550  104266                        ERROR   266             ;MSG B1 ERROR
8636
8637  041552  023737  001364  001350        CMP     CYLADD,FRCYL
8638  041560  001401                        BEQ     5$
8639  041562  104243                        ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
8640
8641  041564                          5$:
8642  041564  104415                        SCOP1
8643  041566  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
8644
8645  041572  004737  045462                JSR     PC,SUBCLR
8646  041576  104024                        ERROR   24              ;CERR AFTER SCLR
8647
8648  041600  005037  001176                CLR     $ESCAPE
8649  041604  013765  001350  000020        MOV     FRCYL,RKDC(R5)  ;CYL #
8650
8651
8652  041612  012700  001674                MOV     #RHTAB,R0
8653  041616  012737  000025  003322        MOV     #<RDHEAD>,HCS1
8654  041624  004737  043530                JSR     PC,DATCMD       ;DO DATA XFER CMD & GET CONTR RDY
8655  041630  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
8656  041632  032737  100000  003322        BIT     #CERR,HCS1
8657  041640  001405                        BEQ     75$
8658  041642  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
8659  041644  104401  056436                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
8660  041650  000137  042706                JMP     $EOP            ;ABORT DRIVE
8661
8662  041654  016520  000024          75$:  MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
8663  041660  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
8664  041664  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
8665
8666
8667  041670  032765  100000  000010        BIT     #DLT,RKCS2(R5)
8668  041676  001407                        BEQ     76$
8669  041700  004737  045132                JSR     PC,GSTAT
8670  041704  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
8671  041706  104401  056436                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
8672  041712  000137  042706                JMP     $EOP            ;ABORT DRIVE
8673  041716                          76$:
8674
8675  041716  023737  001674  001350        CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
8676  041724  001401                        BEQ     74$             ;BR IF SAME
8677  041726  104311                        ERROR   311             ;READ CYL WORD HEADER ERROR
8678  041730                          74$:
8679
8680
8681  041730  023737  001352  012116        CMP     TOCYL,FC        ;ALL CYL DONE?
8682  041736  001404                        BEQ     6$              ;BR IF YES
8683  041740  005337  001352                DEC     TOCYL           ;ELSE DO ANOTHER
8684  041744  000137  040366                JMP     1$
8685  041750                          6$:
8686  041750  004737  047466                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
```

```
8687  041754  000515                          BR      TST41           ;;GO TO NEXT TEST
8688                                                                  ;RETURN HERE IF SW 14 IS SET OR
8689                                                                  ;SW 8 WITH SWR <7:0> APPLY
8690  041756                          8$:
8691
8692  041756  004737  045462                  JSR     PC,SUBCLR
8693  041762  104024                          ERROR   24              ;CERR AFTER SCRL
8694
8695  041764  013765  001352  000020  77$:    MOV     TOCYL,RKDC(R5)  ;CYL#
8696
8697  041772  012737  000017  003322          MOV     #SEEK,HCS1
8698  042000  004737  043472                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
8699  042004  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
8700
8701  042006  013737  001426  003360          MOV     T50000,TEMP1
8702  042014  004737  044176                  JSR     PC,FATT2        ;FIND ATTN
8703  042020  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
8704  042022  032737  100000  003322          BIT     #CERR,HCS1
8705  042030  001401                          BEQ     79$
8706  042032  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
8707
8708  042034  004737  045462          79$:    JSR     PC,SUBCLR
8709  042040  104024                          ERROR   24              ;CERR AFTER SCLR
8710
8711  042042  023737  001352  012100          CMP     TOCYL,LC        ;LAST CYL DONE?
8712  042050  001403                          BEQ     78$             ;BR IF YES
8713  042052  005237  001352                  INC     TOCYL           ;ELSE DO ANOTHER
8714  042056  000742                          BR      77$
8715
8716  042060  004737  045462          78$:    JSR     PC,SUBCLR
8717  042064  104024                          ERROR   24              ;CERR AFTER SCLR
8718
8719  042066  005037  001176                  CLR     $ESCAPE
8720  042072  005737  001410                  TST     LPFLG
8721  042076  001402                          BEQ     80$
8722  042100  000177  137004                  JMP     @$LPERR         ;SW 9 WAS SET.
8723  042104  000177  136776          80$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
8724
8725  042110                          10$:
8726  042110  005237  001410                  INC     LPFLG
8727  042114  032777  001000  137016          BIT     #SW9,@SWR       ;LOOP ON ERROR?
8728  042122  001315                          BNE     8$              ;YES, RECONDITION DRIVE
8729  042124  000137  040526                  JMP     2$              ;RETURN TO MAINLINE
8730  042130                          12$:
8731  042130  005237  001410                  INC     LPFLG
8732  042134  032777  001000  136776          BIT     #SW9,@SWR       ;LOOP ON ERROR?
8733  042142  001305                          BNE     8$              ;YES, RECONDITION DRIVE
8734  042144  000137  041014                  JMP     3$              ;RETURN TO MAINLINE
8735  042150                          14$:
8736  042150  005237  001410                  INC     LPFLG
8737  042154  032777  001000  136756          BIT     #SW9,@SWR       ;LOOP ON ERROR?
8738  042162  001275                          BNE     8$              ;YES, RECONDITION DRIVE
8739  042164  000137  041276                  JMP     4$              ;RETURN TO MAINLINE
8740  042170                          16$:
8741  042170  005237  001410                  INC     LPFLG
8742  042174  032777  001000  136736          BIT     #SW9,@SWR       ;LOOP ON ERROR?
```

```
                        BNE     8$              ;YES, RECONDITION DRIVE
                        JMP     5$              ;RETURN TO MAINLINE

042210                  CYLINV:


                        ;;*****************************************************************
                        ;*TEST 41       SFEK TO ALL INVALID CYLINDERS
                        ;*
                        ;*      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 633 THRU 777 (8) FOR THE RK06
                        ;*      & 1457 THRU 1777 FOR THE RK07.
                        ;*
                        ;*****************************************************************
042210  000004         TST41:  SCOPE
042212  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
042220  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR

042224  004737  045462          JSR     PC,SUBCLR
042230  104024                  ERROR   24              ;CERR AFTER SCLR

042232  012737  000017  003322  MOV     #SEEK,HCS1
042240  004737  043472          JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
042244  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD.

042246  013737  001426  003360  MOV     T50000,TEMP1
042254  004737  044176          JSR     PC,FATT2        ;FIND ATTN
042260  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
042262  032737  100000  003322  BIT     #CERR,HCS1
042270  001401                  BEQ     64$
042272  104210                  ERROR   210             ;CERR AFTER SEEK CMD.

042274  004737  045462  64$:    JSR     PC,SUBCLR
042300  104024                  ERROR   24              ;CERR AFTER SCLR

042302  005000                  CLR     R0
042304  005037  001350          CLR     FRCYL           ;FROM CYL 0
042310  013737  012102  001352  MOV     LCP1,TOCYL              ;1'ST INV CYL

042316                  1$:
042316  104415                  SCOP1
042320  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR

042324  004737  045462          JSR     PC,SUBCLR
042330  104024                  ERROR   24              ;CERR AFTER SCLR

042332  013737  001352  001360  MOV     TOCYL,CALDIF
042340  013765  001352  000020  MOV     TOCYL,RKDC(R5)
042346  012737  000017  003322  MOV     #SEEK,HCS1
042354  004737  043472          JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR RDY
042360  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD
042362  004737  044050          JSR     PC,TSTATN
042366  104245                  ERROR   245             ;NO ATTN AFTER SEEK TO INV CYL
```

```
 8799
 8800  042370  032737  000040  003352          BIT     #D.IDAE,HMR3
 8801  042376  001001                          BNE     2$
 8802  042400  104246                          ERROR   246             ;IDAE NOT SET AFTER SEEK TO INVALID ADDR
 8803  042402  032737  000200  003352  2$:     BIT     #D.FLT,HMR3
 8804  042410  001001                          BNE     4$
 8805  042412  104247                          ERROR   247             ;FLT NOT SET AFTER SEEK TO INV ADDR
 8806  042414  032737  020000  003350  4$:     BIT     #D.PIP,HMR2
 8807  042422  001401                          BEQ     5$
 8808  042424  104250                          ERROR   250             ;PIP SET AFTER SEEK TO INV ADDR
 8809  042426  032737  040000  003350  5$:     BIT     #D.DSC,HMR2
 8810  042434  001001                          BNE     6$
 8811  042436  104251                          ERROR   251             ;DSC NOT SET AFTER SEEK TO INV ADDR
 8812
 8813  042440  005237  001462          6$:     INC     BYPCERR         ;BYPASS CHECKING FOR CERR IN GSTAT1
 8814  042444  012737  050340  003412          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
 8815  042452  012737  002240  003414          MOV     #<D.SKI!D.FLT!D.IDAE>,E.B0
 8816  042460  012737  001720  003416          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 8817  042466  012737  000001  003420          MOV     #1,E.B1
 8818
 8819  042474  004737  044310                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 8820  042500  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 8821  042502  104252                          ERROR   252             ;MSG A0 ERROR AFTER SEEK TO INV CYL
 8822  042504  104253                          ERROR   253             ;MSG B0 ERROR
 8823  042506  104254                          ERROR   254             ;MSG A1 ERROR
 8824  042510  104255                          ERROR   255             ;MSG B1 ERROR
 8825  042512  023737  001362  001352          CMP     CYLDIF,TOCYL
 8826  042520  001401                          BEQ     7$
 8827  042522  104256                          ERROR   256             ;CYL DIFF IN RKMR2 NOT=CYL DIF
 8828  042524  023737  001364  001352  7$:     CMP     CYLADD,TOCYL
 8829  042532  001401                          BEQ     8$
 8830  042534  104257                          ERROR   257             ;CYL ADDR IN RKMR3 NOT=RKDC
 8831
 8832  042536                          8$:
 8833
 8834  042536  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8835  042544  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 8836  042552  012737  000013  003322          MOV     #RECAL,HCS1
 8837  042560  004737  043472                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
 8838  042564  104124                          ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
 8839
 8840  042566  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8841  042574  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 8842  042602  012737  000005  003322          MOV     #CLEAR,HCS1
 8843  042610  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 8844  042614  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 8845  042616  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 8846  042622  000401                          BR      66$
 8847  042624  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 8848  042626                          66$:
 8849
 8850
 8851  042626  004737  045132                  JSR     PC,GSTAT
 8852  042632  032737  000040  003352          BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
 8853  042640  001401                          BEQ     65$             ;BR IF YES
 8854  042642  104155                          ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
```

```
 8855
 8856  042644  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5)
 8857  042652  013737  001412  003362          MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
 8858  042660  004737  044102                  JSR     PC,FATT1
 8859  042664  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
 8860
 8861
 8862  042666  005237  001352                  INC     TOCYL
 8863  042672  023737  001352  012110          CMP     TOCYL,MASK
 8864  042700  001402                          BEQ     $EOP
 8865  042702  000137  042316                  JMP     1$
 8866
 8867
```

```
                        .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 ST5XY

042706                  $EOP:

042706 000004                   SCOPE
042710 005037 001176            CLR     $ESCAPE
042714 012737 000001 001174     MOV     #1,$TIMES
042722 012706 001100            MOV     #STACK,SP
042726 005237 001220            INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
042732 023737 003442 001220     CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
042740 001404                   BEQ     1$              ;BR IF YES
042742 005037 001456            CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
042746 000137 011544            JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
042752 005037 001456    1$:     CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
042756 000401                   BR      $EOP1+2         ;GO TO $EOP1+2
042760 000004           $EOP1:  SCOPE
042762 005037 001102            CLR     $TSTNM          ;;ZERO THE TEST NUMBER
042766 005037 001174            CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
042772 005237 001216            INC     $PASS           ;;INCREMENT THE PASS NUMBER
042776 042737 100000 001216     BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
043004 005327                   DEC     (PC)+           ;;LOOP?
043006 000001           $EOPCT: .WORD   1
043010 003022                   BGT     $DOAGN          ;;YES
043012 012737                   MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
043014 000001           $ENDCT: .WORD   1
043016 043006                   $EOPCT
043020 104401 043065            TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
043024 013746 001216            MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
043030 104405                   TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
043032 104401 043062            TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
043036 013700 000042    $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
043042 001405                   BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
043044 000005                   RESET                   ;;CLEAR THE WORLD
043046 004710           $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
043050 000240                   NOP                     ;;SAVE ROOM
043052 000240                   NOP                     ;;FOR
043054 000240                   NOP                     ;;ACT11
043056                  $DOAGN:
043056 000137                   JMP     @(PC)+          ;;RETURN
043060 043102           $RTNAD: .WORD   ST5XY
043062     377      377 000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
043065     015 042412 042116   $ENDMG: .ASCIZ  <15><12>/END PASS #/
043072 050040 051501 020123
043100 000043
043102 122737 000001 001230 ST5XY:  CMPB    #APTENV,$ENV            ;RUN UNDER APT ?
043110 001007                   BNE     2$              ;BRANCH IF NOT
043112 022737 000002 001216     CMP     #2,$PASS                ;TWO PASSES DONE ?
043120 101003                   BHI     2$              ;BRANCH IF NOT
043122 005237 001102    1$:     INC     $TSTNM          ;CHANGE THE TEST NUMBER
043126 000775                   BR      1$              ;LOOP WAIT FOR LOADING THE NEXT PRG
```

```
                                                                D 14
  8924  043130  000137  007744          2$:     JMP     ST5             ;EXIT
```

```
                                        .SBTTL  SUBROUTINES

                                        ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                        ;

043134  012700  003432          CLRFLG: MOV     #DDUMP,R0
043140  012701  177757                  MOV     #-17.,R1
043144  005020                  1$:     CLR     (R0)+
043146  005201                          INC     R1
043150  001375                          BNE     1$
043152  000207                          RTS     PC

                                        ;
                                        ;TYPE PROGRAM ID IF FTITLE=0
                                        ;

043154  005737  001344          TITLE:  TST     FTITLE
043160  001024                          BNE     1$
043162  005237  001344                  INC     FTITLE
043166  104401  054640                  TYPE    ,MSG1                   ;PROGRAM ID
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
043172  005737  000042                  TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
043176  001012                          BNE     64$                     ;;BRANCH IF YES
043200  123727  001230  000001          CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
043206  001406                          BEQ     64$                     ;;BRANCH IF YES
043210  023727  001140  000176          CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
043216  001005                          BNE     65$                     ;;BRANCH IF NO
043220  104406                          GTSWR                           ;;GET SOFT-SWR SETTINGS
043222  000403                          BR      65$
043224  112737  000001  001134  64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
043232                          65$:
043232  000207                  1$:     RTS     PC

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

043234  104411                  GDRVS:  RDLIN
043236  012600                          MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
043240  012701  177770                  MOV     #-8.,R1                 ;SET UP COUNT
043244  112002                  1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
043246  042702  177400                  BIC     #177400,R2              ;MASK HI BYTE
043252  012703  003444                  MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
043256  012704  000060                  MOV     #60,R4

043262  020402                  2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
043264  001415                          BEQ     3$                      ;BRANCH IF YES
043266  005723                          TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
043270  005204                          INC     R4
043272  020427  000070                  CMP     R4,#70
043276  001371                          BNE     2$                      ;S/B 0-7 OR TERMINATOR
043300  005702                          TST     R2
043302  001022                          BNE     4$
043304  020127  177770                  CMP     R1,#-8.
043310  001426                          BEQ     6$                      ;DEFAULT ALL DRIVES
```

```
 8981  043312  005037  003472          7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
 8982  043316  000207                          RTS     PC              ;FOUND TERMINATOR, EXIT
 8983
 8984  043320  005213                  3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
 8985  043322  005237  003442                  INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
 8986  043326  112002                          MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
 8987  043330  042702  177400                  BIC     #177400,R2      ;MASK
 8988  043334  022702  000054                  CMP     #54,R2          ;IS IT A COMMA?
 8989  043340  001407                          BEQ     5$              ;YES, GO TO NEXT WORD.
 8990  043342  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
 8991  043344  001001                          BNE     4$              ;IF NOT, SOMETHING WRONG.
 8992  043346  000761                          BR      7$              ;FOUND TERMINATOR, EXIT
 8993
 8994  043350  104401  057145          4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
 8995  043354  000137  007134                  JMP     PRGSRT          ;START ALL OVER
 8996
 8997  043360  005201                  5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
 8998  043362  001330                          BNE     1$              ;DRIVES TYPED IN.
 8999  043364  000771                          BR      4$              ;IF NORE, HAVE ERROR.
 9000
 9001  043366  005237  003472          6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
 9002  043372  000207                          RTS     PC              ;EXIT.
 9003
 9004                                  ;
 9005                                  ;ROUTINE TO INPUT RKBAS OR DEFAULT.
 9006                                  ;
 9007
 9008  043374  104412                  GBA:    RDOCT
 9009  043376  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 9010  043400  005700                          TST     R0
 9011  043402  001403                          BEQ     1$              ;BRANCH IF DEFAULT.
 9012  043404  010037  001264                  MOV     R0,$BASE
 9013  043410  000207                          RTS     PC
 9014  043412  012737  177440  001264  1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
 9015  043420  000207                          RTS     PC
 9016
 9017                                  ;
 9018                                  ;ROUTINE TO INPUT RKVEC OR DEFAULT
 9019                                  ;
 9020
 9021  043422  104412                  GINT:   RDOCT
 9022  043424  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 9023  043426  005700                          TST     R0
 9024  043430  001405                          BEQ     1$              ;BRANCH IF DEFAULT
 9025  043432  010037  001314                  MOV     R0,RKVEC
 9026  043436  004737  043454          2$:     JSR     PC,SETINT
 9027  043442  000207                          RTS     PC
 9028  043444  012737  000210  001314  1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
 9029  043452  000771                          BR      2$
 9030
 9031                                  ;
 9032                                  ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
 9033                                  ;
 9034
 9035  043454  013700  001314          SETINT: MOV     RKVEC,R0
 9036  043460  012720  050214                  MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
 9037  043464  013710  001316                MOV     RKPRI,(R0)      ;PR5 TO RKVEC+2
 9038  043470  000207                        RTS     PC
 9039
 9040                                ;
 9041                                ;THIS ROUTINE SETS CDT IN RKCS1 IF DRIVE UNDER TEST IS AN RK07.
 9042                                ;ENTER WITH COMMAND IN HCS1
 9043                                ;
 9044  043472  053737  001170  003322 DOCMD:  BIS     $TMP4,HCS1      ;ADD CDT IF RK07
 9045  043500  013765  003322  000000        MOV     HCS1,RKCS1(R5)  ;DO COMMAND
 9046  043506  013737  001414  003360        MOV     T10,TEMP1
 9047  043514  004737  043566              JSR     PC,FRDY         ;FIND CONTR READY
 9048  043520  000207                        RTS     PC              ;SET HERE IF NOT RDY
 9049  043522  062716  000002              ADD     #2,(SP)         ;ELSE SKIP OVER ERROR
 9050  043526  000207                        RTS     PC
 9051                                ;
 9052                                ;THIS ROUTINE IS SIMILAR TO THE ABOVE BUT IS USED FOR DATA TRANSFERS
 9053                                ;& REQUIRES A LONGER TIMEOUT
 9054                                ;
 9055  043530  053737  001170  003322 DATCMD: BIS     $TMP4,HCS1      ;ADD CDT IF RK07
 9056  043536  013765  003322  000000        MOV     HCS1,RKCS1(R5)  ;DO CMD
 9057  043544  013737  001426  003360        MOV     T50000,TEMP1
 9058  043552  004737  043566              JSR     PC,FRDY         , IND CONTR RDY
 9059  043556  000207                        RTS     PC
 9060  043560  062716  000002              ADD     #2,(SP)
 9061  043564  000207                        RTS     PC
 9062
 9063                                ;
 9064                                ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
 9065                                ;ENTER WITH A COUNT IN TEMP1
 9066                                ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
 9067                                ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
 9068                                ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 9069                                ;
 9070  043566  032765  000200  000000 FRDY:   BIT     #RDY,RKCS1(R5)
 9071  043574  001010                        BNE     1$
 9072  043576  005337  003360              DEC     TEMP1
 9073  043602  001371                        BNE     FRDY
 9074  043604  004737  043722              JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
 9075  043610  004737  045050              JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
 9076  043614  000207                        RTS     PC              ;NO RDY, EXIT
 9077  043616  062716  000002        1$:   ADD     #2,(SP)         ;SKIP OVER ERROR
 9078  043622  004737  043722              JSR     PC,HOLD
 9079  043626  004737  045050              JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
 9080  043632  000207                        RTS     PC
 9081
 9082                                ;
 9083                                ; ROUTINE TO FIND CONTROLLER READY & STORE DRIVE REGS ONLY
 9084                                ;
 9085  043634  032765  000200  000000 FRDY1:  BIT     #RDY,RKCS1(R5)
 9086  043642  001014                        BNE     1$
 9087  043644  005337  003360              DEC     TEMP1
 9088  043650  001371                        BNE     FRDY1
 9089  043652  016537  000034  003350        MOV     RKMR2(R5),HMR2
 9090  043660  016537  000036  003352        MOV     RKMR3(R5),HMR3
 9091  043666  004737  045050              JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
 9092  043672  000207                        RTS     PC              ;NO RDY, EXIT
```

```
 9093  043674  062716  000002            1$:     ADD     #2,(SP)          ;SKIP OVER ERROR
 9094  043700  016537  000034  003350            MOV     RKMR2(R5),HMR2
 9095  043706  016537  000036  003352            MOV     RKMR3(R5),HMR3
 9096  043714  004737  045050                    JSR     PC,CKCERR        ;CHECK FOR SPECIAL CERR CONDITIONS
 9097  043720  000207                            RTS     PC
 9098                                    ;
 9099                                    ;STORE ALL RK611 REGISTERS IN HOLDING REGS
 9100                                    ;
 9101
 9102  043722  016537  000000  003322  HOLD:   MOV     RKCS1(R5),HCS1
 9103  043730  016537  000010  003324            MOV     RKCS2(R5),HCS2
 9104  043736  016537  000002  003326            MOV     RKWC(R5),HWC
 9105  043744  016537  000004  003330            MOV     RKBA(R5),HBA
 9106  043752  016537  000006  003332            MOV     RKDA(R5),HDA
 9107  043760  016537  000012  003334            MOV     RKDS(R5),HDS
 9108  043766  016537  000014  003336            MOV     RKER(R5),HER
 9109  043774  016537  000016  003340            MOV     RKASOF(R5),HASOF
 9110  044002  016537  000020  003342            MOV     RKDC(R5),HDC
 9111  044010  016537  000026  003346            MOV     RKMR1(R5),HMR1
 9112  044016  016537  000034  003350            MOV     RKMR2(R5),HMR2
 9113  044024  016537  000036  003352            MOV     RKM%.3(R5),HMR3
 9114  044032  016537  000030  003354            MOV     RKECPS(R5),HPOS
 9115  044040  016537  000032  003356            MOV     RKECPT(R5),HPAT
 9116  044046  000207                            RTS     PC
 9117
 9118                                    ;
 9119                                    ;ROUTINE TO CHECK FOR CORRECT ATTN
 9120                                    ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
 9121                                    ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
 9122                                    ;
 9123  044050  010446                    TSTATN: MOV     R4,-(SP)         ;SAV R4
 9124  044052  013704  001222                    MOV     $UNIT,R4
 9125  044056  136437  003312  003341            BITB    ATTN(R4),HASOF+1
 9126  044064  001404                            BEQ     1$               ;BRANCH IF ATTN NOT PRESENT
 9127  044066  012604                            MOV     (SP)+,R4         ;RESTOR R4
 9128  044070  062716  000002                    ADD     #2,(SP)          ;INCR RET ADDR TO JUMP OVER ERROR.
 9129  044074  000207                            RTS     PC
 9130  044076  012604                    1$:     MOV     (SP)+,R4         ;RESTOR R4
 9131  044100  000207                            RTS     PC
 9132
 9133                                    ;
 9134                                    ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
 9135                                    ;ENTER WITH TIME IN SECONDS IN TEMP2
 9136                                    ;RETURN IF NO ATTN (ERROR CONDITION)
 9137                                    ;RETURN +2 IF ATTN FOUND
 9138                                    ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 9139                                    ;
 9140
 9141  044102  010446                    FATT1:  MOV     R4,-(SP)         ;SAV R4
 9142  044104  012737  177777  003360    3$:     MOV     #-1,TEMP1
 9143  044112  013704  001222                    MOV     $UNIT,R4
 9144  044116  136465  003312  000017    1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
 9145  044124  001014                            BNE     2$
 9146  044126  005337  003360                    DEC     TEMP1
 9147  044132  001371                            BNE     1$
 9148  044134  005337  003362                    DEC     TEMP2
```

```
                        BNE     3$
                        CLR     RKMR1(R5)       ;SELECT WORD 0
                        JSR     PC,GSTAT        ;GET LATEST STATUS
                        MOV     (SP)+,R4        ;RESTOR R4
                        RTS     PC
                2$:     CLR     RKMR1(R5)
                        JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
                        MOV     (SP)+,R4        ;RESTOR R4
                        ADD     #2,(SP)         ;SKIP OVER ERROR
                        RTS     PC

                ;
                ;ROUTINE TO FIND ATTN WITHIN 1 SEC
                ;ENTER WITH COUNT IN TEMP1
                ;RETURN IF NO ATTN (ERROR)
                ;RETURN +2 IF ATTN FOUND
                ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
                ;

                FATT2:  MOV     R4,-(SP)        ;SAV R4
                2$:     MOV     $UNIT,R4
                        BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
                        BNE     1$
                        DEC     TEMP1
                        BNE     2$
                        CLR     RKMR1(R5)       ;SELECT WORD 0
                        JSR     PC,GSTAT        ;GET LATEST STATUS.
                        MOV     (SP)+,R4        ;RESTOR R4
                        RTS     PC
                1$:     CLR     RKMR1(R5)
                        JSR     PC,GSTAT
                        MOV     (SP)+,R4        ;RESTOR R4
                        ADD     #2,(SP)         ;SKIP OVER ERROR
                        RTS     PC
                ;
                ;ENTER WITH A COUNT IN TEMP1
                ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
                ;WHEN COUNT IS 0...BASED ON AN 11/05.
                ;
                DLY:    TST     TEMP1           ;5.6 US
                        BEQ     1$              ;2.5 US
                        DEC     TEMP1           ;6.8 US
                        BR      DLY             ;2.5 US
                1$:     RTS     PC              ;3.8 US
                ;
                ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
                ;

                BYP:    TYPE    ,MSG14          ;BYPASS DRIVE
                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                ;;TYPE DR#
                        TYPOS                   ;;GO TYPE--OCTAL ASCII
                        .BYTE   1               ;;TYPE 1 DIGIT(S)
                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
                        RTS     PC
                ;
```

```
9205                                    ; THIS ROUTINE READS ALL MSG A&B WORDS & CHECKS THEM AS REQ'D
9206                                    ;
9207  044310  017637  000000  001466  CHKMSG: MOV     @(SP),CHKFLG    ;PASS MSGS TO BE TESTED
9208  044316  062716  000002                  ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
9209  044322  004737  045166                  JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS
9210
9211  044326  053737  001222  003412          BIS     $UNIT,E.A0      ;SET UNIT #
9212  044334  053737  001222  003416          BIS     $UNIT,E.A1
9213  044342  053737  001222  003422          BIS     $UNIT,E.A2
9214  044350  053737  001222  003426          BIS     $UNIT,E.A3
9215  044356  053737  012114  003412          BIS     E.DDT,E.A0      ;ADD EXP. DRV TYPE
9216
9217  044364  013746  003360                  MOV     TEMP1,-(SP)     ;SAVE TEMP 1
9218
9219  044370  013737  003412  003360          MOV     E.A0,TEMP1
9220  044376  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A0
9221  044402  013737  003360  003412          MOV     TEMP1,E.A0
9222
9223  044410  013737  003416  003360          MOV     E.A1,TEMP1
9224  044416  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A1
9225  044422  013737  003360  003416          MOV     TEMP1,E.A1
9226
9227  044430  013737  003422  003360          MOV     E.A2,TEMP1
9228  044436  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
9229  044442  013737  003360  003422          MOV     TEMP1,E.A2
9230
9231  044450  013737  003414  003360          MOV     E.B0,TEMP1
9232  044456  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
9233  044462  013737  003360  003414          MOV     TEMP1,E.B0
9234
9235  044470  013737  003420  003360          MOV     E.B1,TEMP1
9236  044476  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
9237  044502  013737  003360  003420          MOV     TEMP1,E.B1
9238
9239  044510  013737  003424  003360          MOV     E.B2,TEMP1
9240  044516  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
9241  044522  013737  003360  003424          MOV     TEMP1,E.B2
9242
9243  044530  013737  003430  003360          MOV     E.B3,TEMP1
9244  044536  004737  047362                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
9245  044542  013737  003360  003430          MOV     TEMP1,E.B3
9246
9247  044550  012637  003360                  MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
9248  044554  013737  001176  001172          MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE
9249
9250  044562  023737  003372  003412          CMP     H.A0,E.A0       ;TEST MSG A0
9251  044570  001411                          BEQ     2$              ;BR IF OK
9252  044572  012737  044604  001176          MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
9253  044600  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR
9254  044602  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
9255
9256  044604  032777  001000  134326  1$:     BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
9257  044612  001107                          BNE     20$             ;& BR IF LOOP ON ERROR
9258  044614  062716  000002          2$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
9259
9260  044620  023737  003374  003414          CMP     H.B0,E.B0       ;TEST MSG B0
```

```
 9261  044626  001411                          BEQ     5$              ;BR IF OK
 9262  044630  012737  044642  001176          MOV     #4$,$ESCAPE     ;ELSE SETUP ESCAPE
 9263  044636  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR
 9264  044640  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
 9265
 9266  044642  032777  001000  134270  4$:     BIT     #SW9,@SWR       ;RETURN HERE FROM MAINLINE ERROR
 9267  044650  001070                          BNE     20$             ;& BR IF LOOP ON ERROR
 9268  044652  062716  000002          5$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
 9269
 9270  044656  023737  003376  003416          CMP     H.A1,E.A1       ;TEST MSG A1
 9271  044664  001411                          BEQ     8$              ;BR IF OK
 9272  044666  012737  044700  001176          MOV     #7$,$ESCAPE
 9273  044674  011646                          MOV     (SP),-(SP)
 9274  044676  000207                          RTS     PC
 9275
 9276  044700  032777  001000  134232  7$:     BIT     #SW9,@SWR
 9277  044706  001051                          BNE     20$
 9278  044710  062716  000002          8$:     ADD     #2,(SP)
 9279
 9280  044714  023737  003400  003420          CMP     H.B1,E.B1       ;TEST MSG B1
 9281  044722  001411                          BEQ     11$             ;BR IF OK
 9282  044724  012737  044736  001176          MOV     #10$,$ESCAPE
 9283  044732  011646                          MOV     (SP),-(SP)
 9284  044734  000207                          RTS     PC
 9285
 9286  044736  032777  001000  134174  10$:    BIT     #SW9,@SWR
 9287  044744  001032                          BNE     20$
 9288  044746  062716  000002          11$:    ADD     #2,(SP)
 9289
 9290  044752  032737  000001  001466  12$:    BIT     #T.A2,CHKFLG    ;TEST MSG A2?
 9291  044760  001402                          BEQ     13$             ;BR IF NO
 9292  044762  004737  046122                  JSR     PC,RCYLD        ;PUT INFO IN CYLDIF, DO NOT CHECK
 9293
 9294  044766  032737  000002  001466  13$:    BIT     #T.B2,CHKFLG    ;TEST MSG B2?
 9295  044774  001402                          BEQ     14$             ;BR IF NO
 9296  044776  004737  046174                  JSR     PC,RCYLA        ;PUT INFO IN CYLADD, DO NOT CHECK
 9297
 9298  045002  032737  000004  001466  14$:    BIT     #T.B3,CHKFLG    ;TEST MSG B3?
 9299  045010  001404                          BEQ     15$
 9300  045012  004737  046232                  JSR     PC,RSEC         ;PUT INFO IN SECTOR, DO NOT CHECK
 9301  045016  004737  046270                  JSR     PC,RHEAD        ;PUT INFO IN HEADA, DO NOT CHECK
 9302
 9303  045022  013737  001172  001176  15$:    MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
 9304  045030  000207                          RTS     PC
 9305
 9306  045032  012706  001100          20$:    MOV     #STACK,SP       ;RESET STACK PTR
 9307  045036  013737  001172  001176          MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
 9308  045044  000177  134040                  JMP     @$LPERR
 9309
 9310                                  ;
 9311                                  ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
 9312                                  ; IE: IF NED, CTO OR MDS SET, MSG A&B ARE INVALID
 9313                                  ;
 9314  045050  005737  001462          CKCERR: TST     BYPCERR
 9315  045054  001025                          BNE     4$
 9316  045056  032737  100000  003322          BIT     #CERR,HCS1
```

```
 9317  045064  001001                        BNE     1$              ;BR IF CERR
 9318  045066  000207                        RTS     PC
 9319
 9320  045070  032737  004000  003322  1$:   BIT     #CTO,HCS1
 9321  045076  001402                        BEQ     2$              ;BR IF NOT CTO
 9322  045100  104326                        ERROR   326             ;CTO ERROR, MSG A&B INVALID
 9323  045102  000207                        RTS     PC
 9324
 9325  045104  032737  010000  003324  2$:   BIT     #NED,HCS2
 9326  045112  001401                        BEQ     3$              ;BR IF NOT NED
 9327  045114  104327                        ERROR   327             ;NED ERROR, MSG A&B INVALID
 9328
 9329  045116  032737  001000  003324  3$:   BIT     #MDS,HCS2
 9330  045124  001401                        BEQ     4$
 9331  045126  104330                        ERROR   330             ;MDS ERROR, MSG A&B INVALID
 9332
 9333  045130  000207                4$:     RTS     PC
 9334
 9335                                        ;
 9336                                        ;THIS ROUTINE DOES THE SELECT DRIVE CMD TO GET STATUS
 9337                                        ;IT THEN WAITS FOR CONTROLLER  READY.
 9338                                        ;IF RDY NOT RECEIVED BY A TIMEOUT, AN ERROR IS FLAGGED
 9339                                        ;
 9340  045132  013746  003360        GSTAT:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
 9341  045136  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 9342  045144  012737  000001  003322        MOV     #SELDRV,HCS1
 9343  045152  004737  043472                JSR     PC,DOCMD        ;DR SELDRV (STATUS) CMD & GET CONTR RDY
 9344  045156  104117                        ERROR   117             ;RDY NOT SET BY END OF SELECT DRIVE CMD
 9345  045160  012637  003360                MOV     (SP)+,TEMP1     ;RESTOR TEMP1
 9346  045164  000207                        RTS     PC
 9347
 9348                                        ;
 9349                                        ; THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
 9350                                        ; & ALL CONTROLLER REGISTERS
 9351                                        ;
 9352  045166  013746  003360        GSTAT1: MOV     TEMP1,-(SP)     ;SAVE TEMP 1
 9353  045172  004737  043722                JSR     PC,HOLD         ;GET ALL CONTR REGS
 9354  045176  012765  100000  000000        MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
 9355  045204  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 9356  045212  012765  000003  000026        MOV     #3,RKMR1(R5)    ;SELECT WORD 3
 9357  045220  004737  045416                JSR     PC,GSTAT2
 9358  045224  104117                        ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
 9359  045226  013737  003350  003406        MOV     HMR2,H.A3       ;STORE MSG A3
 9360  045234  013737  003352  003410        MOV     HMR3,H.B3       ;STORE MSG B3
 9361
 9362  045242  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 9363  045250  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
 9364  045256  012765  000002  000026        MOV     #2,RKMR1(R5)    ;SELECT WORD 2
 9365  045264  004737  045416                JSR     PC,GSTAT2
 9366  045270  104117                        ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
 9367  045272  013737  003350  003402        MOV     HMR2,H.A2       ;STORE MSG A2
 9368  045300  013737  003352  003404        MOV     HMR3,H.B2       ;STORE MSG B2
 9369
 9370  045306  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 9371  045314  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
 9372  045322  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
```

```
 9373  045330  004737  045416            JSR     PC,GSTAT2
 9374  045334  104117                    ERROR   117                 ;RDY NOT SET BY END OF SELECT DRV CMD
 9375  045336  013737  003350  003376    MOV     HMR2,H.A1           ;STORE MSG A1
 9376  045344  013737  003352  003400    MOV     HMR3,H.B1           ;STORE MSG B1
 9377
 9378  045352  012765  100000  000000    MOV     #CCLR,RKCS1(R5)
 9379  045360  013765  001222  000010    MOV     $UNIT,RKCS2(R5)
 9380  045366  004737  045416            JSR     PC,GSTAT2
 9381  045372  104117                    ERROR   117                 ;RDY NOT SET BY END OF SEL DRV CMD
 9382  045374  013737  003350  003372    MOV     HMR2,H.A0           ;STORE MSG A0
 9383  045402  013737  003352  003374    MOV     HMR3,H.B0           ;STORE MSG B0
 9384
 9385  045410  012637  003360            MOV     (SP)+,TEMP1         ;RESTORE TEMP1
 9386  045414  000207                    RTS     PC
 9387
 9388
 9389  045416  012737  000001  003322  GSTAT2: MOV     #SELDRV,HCS1
 9390  045424  053737  001170  003322          BIS     $TMP4,HCS1          ;ADD CDT IF RK07
 9391  045432  013765  003322  000000          MOV     HCS1,RKCS1(R5)      ;GET STATUS
 9392  045440  013737  001414  003360          MOV     T10,TEMP1
 9393  045446  004737  043634                  JSR     PC,FRDY1            ;FIND CONTR RDY & STORE DRIVE REGS ONLY
 9394  045452  000207                          RTS     PC                  ;RET HERE IF NOT RDY
 9395  045454  062716  000002                  ADD     #2,(SP)             ;RET HERE IF OK
 9396  045460  000207                          RTS     PC
 9397                                  ;
 9398                                  ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
 9399                                  ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
 9400                                  ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
 9401                                  ;RETURN IF CERR SET
 9402                                  ;RETURN +2 IF CERR CLEAR
 9403                                  ;
 9404  045462  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
 9405  045470  013737  001414  003360          MOV     T10,TEMP1
 9406  045476  004737  043566                  JSR     PC,FRDY             ;FIND RDY
 9407  045502  104120                          ERROR   120                 ;RDY NOT SET BY END OF SCLR
 9408  045504  013765  001222  000010          MOV     $UNIT,RKCS2(R5)     ;CURRENT DRIVE #
 9409  045512  005065  000026                  CLR     RKMR1(R5)           ;SELECT WORD 0
 9410  045516  004737  045132                  JSR     PC,GSTAT            ;GET STATUS
 9411  045522  032737  100000  003322          BIT     #CERR,HCS1          ;CHECK FOR CONT ERROR
 9412  045530  001401                          BEQ     1$
 9413  045532  000207                          RTS     PC
 9414  045534  062716  000002          1$:     ADD     #2,(SP)             ;SKIP OVER ERROR
 9415  045540  000207                          RTS     PC
 9416
 9417                                  ;
 9418                                  ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
 9419                                  ;
 9420  045542  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)        ;WORD 3
 9421  045550  004737  045132                  JSR     PC,GSTAT
 9422  045554  013737  003352  001406          MOV     HMR3,SECTOR
 9423  045562  042737  177017  001406          BIC     #^C<M.SECT>,SECTOR
 9424  045570  006237  001406                  ASR     SECTOR              ;RIGHT JUSTIFY
 9425  045574  006237  001406                  ASR     SECTOR              ;SECTOR
 9426  045600  006237  001406                  ASR     SECTOR              ;INFO
 9427  045604  006237  001406                  ASR     SECTOR
 9428  045610  000207                          RTS     PC
```

```
 9429
 9430
 9431                                   ;
 9432                                   ;FIND SECTOR 0 IN 22 SECTOR FORMAT.
 9433                                   ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
 9434                                   ;
 9435  045612  013746  003360          FS022:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
 9436  045616  013737  001424  003360          MOV     T5000,TEMP1     ;SETUP TIMEOUT
 9437  045624  004737  045542          1$:     JSR     PC,RDSEC                ;READ SECTOR
 9438  045630  005737  001406                  TST     SECTOR          ;LOOK FOR SECTOR 0
 9439  045634  001005                          BNE     2$
 9440  045636  004737  045542                  JSR     PC,RDSEC
 9441  045642  005737  001406                  TST     SECTOR
 9442  045646  001406                          BEQ     3$              ;BR IF SAME TWICE
 9443  045650  005337  003360          2$:     DEC     TEMP1
 9444  045654  001363                          BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
 9445  045656  012637  003360                  MOV     (SP)+,TEMP1     ;ELSE RESTORE TEMP1
 9446  045662  000207                          RTS     PC              ;EXIT
 9447  045664  012637  003360          3$:     MOV     (SP)+,TEMP1
 9448  045670  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
 9449  045674  000207                          RTS     PC
 9450
 9451                                   ;
 9452                                   ;FIND NEXT SECTOR IN 22 SECTOR FORMAT
 9453                                   ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
 9454                                   ;
 9455
 9456  045676  013746  003360          FNS22:  MOV     TEMP1,-(SP)     ;SAVE TEMP 1
 9457  045702  013737  001420  003360          MOV     T500,TEMP1      ;SETUP TIMEOUT
 9458  045710  004737  045542          1$:     JSR     PC,RDSEC                ;READ SECTOR
 9459  045714  023737  001402  001406          CMP     PSEC,SECTOR
 9460  045722  001406                          BEQ     3$              ;BR IF SAME
 9461  045724  004737  045542                  JSR     PC,RDSEC                ;ELSE TRY READ DIFFERENT TWICE
 9462  045730  023737  001402  001406          CMP     PSEC,SECTOR
 9463  045736  001006                          BNE     2$              ;BR IF DIFFERENT TWICE
 9464  045740  005337  003360          3$:     DEC     TEMP1           ;ELSE TRY AGAIN IF TIME LEFT
 9465  045744  001361                          BNE     1$
 9466  045746  012637  003360                  MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
 9467  045752  000207                          RTS     PC
 9468  045754  012637  003360          2$:     MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
 9469  045760  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
 9470  045764  000207                          RTS     PC
 9471                                   ;
 9472                                   ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
 9473                                   ;
 9474  045766  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
 9475  045774  004737  045132                  JSR     PC,GSTAT
 9476  046000  013737  003350  001362          MOV     HMR2,CYLDIF
 9477  046006  043737  012112  001362          BIC     MASK1,CYLDIF
 9478  046014  006237  001362                  ASR     CYLDIF          ;RIGHT JUSTIFY
 9479  046020  006237  001362                  ASR     CYLDIF          ;CYL DIFF/OFFSET
 9480  046024  006237  001362                  ASR     CYLDIF          ;INFO
 9481  046030  006237  001362                  ASR     CYLDIF
 9482  046034  023737  001362  012110          CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
 9483  046042  001002                          BNE     1$              ;BR IF NOT
 9484  046044  005037  001362                  CLR     CYLDIF          ;CLR IF YES
```

```
 9485  046050  000207                    1$:     RTS     PC
 9486                                      ;
 9487                                      ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
 9488                                      ;
 9489  046052  012765  000002  000026    RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
 9490  046060  004737  045132              JSR     PC,GSTAT
 9491  046064  013737  003352  001364            MOV     MMR3,CYLADD
 9492  046072  043737  012112  001364            BIC     MASK1,CYLADD
 9493  046100  006237  001364                    ASR     CYLADD          ;RIGHT JUSTIFY
 9494  046104  006237  001364                    ASR     CYLADD          ;CYL ADDR
 9495  046110  006237  001364                    ASR     CYLADD          ;INFO
 9496  046114  006237  001364                    ASR     CYLADD
 9497  046120  000207                            RTS     PC
 9498                                      ;
 9499                                      ; READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
 9500                                      ;
 9501  046122  013737  003402  001362    RCYLD:  MOV     H.A2,CYLDIF
 9502  046130  043737  012112  001362            BIC     MASK1,CYLDIF
 9503  046136  006237  001362                    ASR     CYLDIF          ;RIGHT JUSTIFY
 9504  046142  006237  001362                    ASR     CYLDIF
 9505  046146  006237  001362                    ASR     CYLDIF
 9506  046152  006237  001362                    ASR     CYLDIF
 9507  046156  023737  001362  012110            CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
 9508  046164  001002                            BNE     1$              ;BR IF NO
 9509  046166  005037  001362                    CLR     CYLDIF          ;ELSE CLEAR
 9510  046172  000207                    1$:     RTS     PC
 9511                                      ;
 9512                                      ; READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
 9513                                      ;
 9514  046174  013737  003404  001364    RCYLA:  MOV     H.B2,CYLADD
 9515  046202  043737  012112  001364            BIC     MASK1,CYLADD
 9516  046210  006237  001364                    ASR     CYLADD          ;RIGHT JUSTIFY
 9517  046214  006237  001364                    ASR     CYLADD
 9518  046220  006237  001364                    ASR     CYLADD
 9519  046224  006237  001364                    ASR     CYLADD
 9520  046230  000207                            RTS     PC
 9521                                      ;
 9522                                      ; READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
 9523                                      ;
 9524  046232  013737  003410  001406    RSEC:   MOV     H.B3,SECTOR
 9525  046240  042737  177017  001406            BIC     #^C<M.SECT>,SECTOR      ;CLEAR UNWANTED INFO
 9526  046246  006237  001406                    ASR     SECTOR          ;RIGHT JUSTIFY
 9527  046252  006237  001406                    ASR     SECTOR
 9528  046256  006237  001406                    ASR     SECTOR
 9529  046262  006237  001406                    ASR     SECTOR
 9530  046266  000207                            RTS     PC
 9531                                      ;
 9532                                      ; READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
 9533                                      ;
 9534  046270  013737  003410  001432    RHEAD:  MOV     H.B3.HEADA
 9535  046276  042737  170777  001432            BIC     #^C<M.HEAD>,HEADA       ;CLEAR UNWANTED INFO
 9536  046304  006237  001432                    ASR     HEADA           ;RIGHT JUSTIFY IT
 9537  046310  000337  001432                    SWAB    HEADA
 9538  046314  000207                            RTS     PC
 9539                                      ;
 9540                                      ;FIND LIMIT DETECT ON SEEK IN RKMR3 BEFORE TIMEOUT
```

```
                                        ;RETURN IF NOT FOUND: ERROR
                                        ;RETURN+2 IF FOUND: SKIP OVER ERROR
                                        ;
046316  005037  001460        FLIM:   CLR     LIMERR          ;LIMIT DETECT ERROR FLAG
046322  012737  177777  003360        MOV     #-1,TEMP1       ;SETUP TIMEOUT
046330  012765  000001  000026        MOV     #1,RKMR1(R5)    ;WORD 1
046336  004737  045132        1$:     JSR     PC,GSTAT
046342  032737  020000  003352        BIT     #D.LIMD,HMR3
046350  001006                        BNE     2$              ;EXIT IF SET
046352  005337  003360                DEC     TEMP1
046356  001367                        BNE     1$
046360  005237  001460                INC     LIMERR          ;SET LIMIT DETECT FLAG
046364  000207                        RTS     PC
046366  062716  000002        2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
046372  000207                        RTS     PC
                                        ;
                                        ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE TIMEOUT
                                        ;ENTER WITH TIME IN SECONDS IN TEMP2
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND - SKIP OVER ERROR
                                        ;
046374  012737  177777  003360  FHDHM:   MOV     #-1,TEMP1       ;ALL 1'S
046402  012765  000001  000026        MOV     #1,RKMR1(R5)    ;WORD 1
046410  004737  045132        1$:     JSR     PC,GSTAT
046414  032737  000040  003350        BIT     #D.HDHM,HMR2
046422  001007                        BNE     2$
046424  005337  003360                DEC     TEMP1
046430  001367                        BNE     1$
046432  005337  003362                DEC     TEMP2
046436  001356                        BNE     FHDHM
046440  000207                        RTS     PC
046442  062716  000002        2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
046446  000207                        RTS     PC
                                        ;
                                        ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE TIMEOUT
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND: SKIP OVER ERROR
                                        ;
046450  012737  177777  003360  FLOAD:   MOV     #-1,TEMP1       ;SETUP TIMEOUT
046456  012765  000001  000026        MOV     #1,RKMR1(R5)    ;WORD 1
046464  004737  045132        1$:     JSR     PC,GSTAT
046470  032737  010000  003350        BIT     #D.LOAD,HMR2
046476  001004                        BNE     2$
046500  005337  003360                DEC     TEMP1
046504  001367                        BNE     1$
046506  000207                        RTS     PC
046510  062716  000002        2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
046514  000207                        RTS     PC
                                        ;
                                        ;ROUTINE TO FIND SPOK BEFORE TIMEOUT
                                        ;ENTER WITH APPROX TIME IN TEMP2
                                        ;RETURN IF NOT CLEARED
                                        ;RETURN +2 IF CLEARED TO SKIP OVER ERROR
                                        ;
046516  012737  177777  003360  FSPOK:   MOV     #-1,TEMP1       ;ALL 1'S
046524  012765  000001  000026        MOV     #1,RKMR1(R5)    ;WORD 1
```

```
 9597  046532  004737  045132          1$:     JSR     PC,GSTAT
 9598  046536  032737  001000  003350          BIT     #D.SPOK,HMR2    ;SEE IF SPOK CLEARED
 9599  046544  001407                          BEQ     2$
 9600  046546  005337  003360                  DEC     TEMP1
 9601  046552  001367                          BNE     1$
 9602  046554  005337  003362                  DEC     TEMP2
 9603  046560  001356                          BNE     FSPOK
 9604  046562  000207                          RTS     PC
 9605  046564  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 9606  046570  000207                          RTS     PC
 9607                                  ;
 9608                                  ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
 9609                                  ;ENTER WITH CYL # IN 'CALADD'
 9610                                  ;ENTER WITH HEAD # IN 'HEAD'
 9611                                  ;ENTER WITH FORMAT IN 'FORMAT'
 9612                                  ;
 9613  046572  010046                  FHDTAB: MOV     R0,-(SP)        ;SAV R0
 9614  046574  010146                          MOV     R1,-(SP)        ;SAV R1
 9615  046576  012700  001470                  MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
 9616  046602  005001                          CLR     R1              ;SECTOR COUNTER
 9617  046604  013737  001430  001434          MOV     HEAD,HD1
 9618  046612  006337  001434                  ASL     HD1
 9619  046616  006337  001434                  ASL     HD1
 9620  046622  006337  001434                  ASL     HD1
 9621  046626  006337  001434                  ASL     HD1
 9622  046632  006337  001434                  ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
 9623  046636  013737  001436  001440          MOV     FORMAT,FMT1
 9624  046644  000337  001440                  SWAB    FMT1
 9625  046650  006337  001440                  ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
 9626
 9627  046654  013720  001366          1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
 9628  046660  010110                          MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
 9629  046662  053710  001434                  BIS     HD1,(R0)        ;               -HEAD NO
 9630  046666  053710  001440                  BIS     FMT1,(R0)       ;               -FORMAT
 9631  046672  005737  001464                  TST     BYPFMT
 9632  046676  001403                          BEQ     2$              ;BR IF TRUE FORMAT
 9633  046700  052710  140000                  BIS     #<BIT14!BIT15>,(R0)     ;SET GOOD SECTOR FLAGS
 9634  046704  000402                          BR      3$
 9635  046706  004737  046766          2$:     JSR     PC,SECFLG       ;GET  SECTOR FLAGS
 9636
 9637  046712  013737  001366  003360  3$:     MOV     CALADD,TEMP1
 9638  046720  011037  003362                  MOV     (R0),TEMP2
 9639  046724  043737  001366  003362          BIC     CALADD,TEMP2
 9640  046732  042037  003360                  BIC     (R0)+,TEMP1
 9641  046736  053737  003360  003362          BIS     TEMP1,TEMP2
 9642  046744  013720  003362                  MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
 9643
 9644  046750  005201                          INC     R1              ;SECTOR CTR
 9645  046752  020127  000026                  CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
 9646  046756  001336                          BNE     1$              ;BR IF NO
 9647
 9648  046760  012601                          MOV     (SP)+,R1        ;RESTOR R1
 9649  046762  012600                          MOV     (SP)+,R0        ;RESTOR R0
 9650  046764  000207                          RTS     PC
 9651
 9652                                  ;
```

```
                        ; THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS TEST
                        ; & SETS BITS 14 & 15 APPROPRIATELY.
                        ;
047766  010246          SECFLG: MOV     R2,-(SP)        ;SAVE R2
047770  005737  001436          TST     FORMAT
047774  001016                  BNE     1$              ;BR IF 20 SECTOR FORMAT
047776  012702  002314          MOV     #BSE22H+8.,R2
047002  004737  047036          JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
047006  052710  100000          BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR

047012  012702  054504          MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
047016  004737  047036          JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
047022  052710  040000          BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR

047026  012602                  MOV     (SP)+,R2        ;ELSE RETURN HERE
047030  000207                  RTS     PC

047032  012602          1$:     MOV     (SP)+,R2        ;RESTORE R2
047034  000207                  RTS     PC

                        ;
                        ; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
                        ; ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC) IN TEMP1
                        ; RETURN IF NO COMPARE
                        ; RETURN +4 IF COMPARE
                        ;
047036  010346          FLGTST: MOV     R3,-(SP)        ;SAVE R3

047040  021227  177777  1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
047044  001002                  BNE     2$              ;BR IF NO
047046  012603                  MOV     (SP)+,R3        ;RESTORE R3
047050  000207                  RTS     PC

047052  022237  001366  2$:     CMP     (R2)+,CALADD    ;SEE IF = CYL #, & ADV PTR TO TRACK/SECTOR WORD
047056  001403                  BEQ     3$
047060  062702  000002          ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
047064  000765                  BR      1$

047066  013703  001430  3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
047072  000303                  SWAB    R3
047074  050103                  BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
047076  022203                  CMP     (R2)+,R3        ;SECTOR/HEAD COMPARE? & INCR TO NEXT CYL WORD
047100  001401                  BEQ     4$              ;BR IF YES
047102  000756                  BR      1$              ;TRY NEXT CYL

047104  012603          4$:     MOV     (SP)+,R3        ;RESTORE R3
047106  062716  000004          ADD     #4,(SP)         ;INCREMENT RET ADDR
047112  000207                  RTS     PC
                        ;
                        ;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
                        ;
047114  010046          SORT:   MOV     R0,-(SP)        ;SAVE R0
047116  010146                  MOV     R1,-(SP)        ;SAVE R1
047120  004737  045542          JSR     PC,RDSEC
```

```
 9709  047124  062737  000001  001406          ADD     #1,SECTOR
 9710  047132  004737  047222                  JSR     PC,MULT6        ;MULT SECTOR BY 6
 9711
 9712  047136  012700  000204                  MOV     #132.,R0
 9713  047142  163700  001406                  SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
 9714  047146  010037  001406                  MOV     R0,SECTOR
 9715  047152  062737  001674  001406          ADD     #RHTAB,SECTOR   ;SAVE INDEX
 9716
 9717  047160  062700  001674                  ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
 9718  047164  012701  002100                  MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
 9719
 9720  047170  012021                  1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
 9721  047172  020027  002100                  CMP     R0,#RHTAB+132.
 9722  047176  001374                          BNE     1$
 9723
 9724  047200  012700  001674                  MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
 9725  047204  012021                  2$:     MOV     (R0)+,(R1)+
 9726  047206  020037  001406                  CMP     R0,SECTOR
 9727  047212  001374                          BNE     2$
 9728
 9729  047214  012601                          MOV     (SP)+,R1        ;RESTOR R1
 9730  047216  012600                          MOV     (SP)+,R0        ;RESTOR R0
 9731  047220  000207                          RTS     PC
 9732
 9733                                  ;
 9734                                  ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
 9735                                  ;
 9736  047222  006337  001406          MULT6:  ASL     SECTOR          ;2 X SECTOR
 9737  047226  013746  001406                  MOV     SECTOR,-(SP)
 9738  047232  006337  001406                  ASL     SECTOR          ;4 X SECTOR
 9739  047236  062637  001406                  ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
 9740  047242  000207                          RTS     PC
 9741
 9742                                  ;
```

```
 9743                                   ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
 9744                                   ;
 9745  047244  005037  001376           CLKON:  CLR     TIMUP
 9746  047250  005737  003466                   TST     PCLKF
 9747  047254  001004                           BNE     1$              ;BRANCH IF P-CLOCK PRESENT
 9748  047256  012777  000100  132042           MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
 9749  047264  000207                           RTS     PC
 9750  047266  012777  177777  132026  1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
```

```
 9751  047274  012777  000135  132016          MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
 9752  047302  000207                          RTS     PC              ;LINE FREQ & RUN
 9753                                  ;
 9754                                  ;KW11-L & KW11-P INTERRUPT HANDLER
 9755                                  ;
 9756  047304  005037  001376          CLOCK:  CLR     TIMUP
 9757  047310  005337  001372                  DEC     COUNT
 9758  047314  001010                          BNE     1$
 9759  047316  013737  001370  001372          MOV     HZ,COUNT
```

```
 9760  047324  005337  001374                 DEC     SEC
 9761  047330  001002                         BNE     1$
 9762  047332  005237  001376                 INC     TIMUP           ;SORRY, TIME IS UP
 9763  047336  000002                 1$:     RTI
 9764                                 ;
 9765                                 ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
 9766                                 ;
 9767  047340  005737  003466         CLKOF:  TST     PCLKF
 9768  047344  001003                         BNE     1$              ;BRACH IF P-CLOCK PRESENT
 9769  047346  005077  131754                 CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
 9770  047352  000207                         RTS     PC
 9771  047354  005077  131740         1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
 9772  047360  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
                                        ;
047362  010046                  SBPAR:  MOV     R0,-(SP)        ;SAVE R0
047364  010146                          MOV     R1,-(SP)        ;SAVE R1
047366  012700  000021                  MOV     #17.,R0         ;SHIFT COUNTER
047372  005001                          CLR     R1              ;COUNT # OF 1'S IN TEMP1
047374  000241                          CLC                     ;CLEAR CARRY
                                        
047376  006137  003360          1$:     ROL     TEMP1
047402  103001                          BCC     2$              ;BR IF CARRY CLEAR
047404  005201                          INC     R1              ;COUNT # OF 1'S
047406  005300                  2$:     DEC     R0              ;SHIFT COUNTER
047410  001372                          BNE     1$
                                        
047412  032701  000001                  BIT     #BIT0,R1
047416  001003                          BNE     3$              ;BR IF ODD # IN R0
047420  052737  100000  003360          BIS     #M.PAR,TEMP1    ;SET PARITY BIT
047426  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
047430  012600                          MOV     (SP)+,R0        ;RESTORE R0
047432  000207                          RTS     PC
                                        
                                        ;
                                        ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                        ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                        ; IE: MY LOOP MACRO
                                        ;
047434  032777  001000  131476  SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
047442  001406                          BEQ     1$              ;BR IF NO
047444  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
047450  001403                          BEQ     1$              ;BR IF NO
047452  013716  001110                  MOV     $LPERR,(SP)
047456  000002                          RTI
                                        
047460  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
047464  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
                                        ;
047466  005037  001176          $WTST:  CLR     $ESCAPE
047472  005037  001410                  CLR     LPFLG
```

```
 9829  047476  032777  040000  131434          BIT     #SW14,@SWR      ;LOOP ON TEST?
 9830  047504  001403                          BEQ     3$              ;BR IF NO
 9831  047506  062716  000002          1$:     ADD     #2,(SP)
 9832  047512  000207                  2$:     RTS     PC
 9833
 9834  047514  032777  000400  131416  3$:     BIT     #SW8,@SWR       ;LOOP ON SPECIFIC TEST?
 9835  047522  001773                          BEQ     2$              ;BR IF NO
 9836  047524  127737  131410  001102          CMPB    @SWR,$TSTNM     ;RIGHT TEST? SWR <7:0>
 9837  047532  001765                          BEQ     1$              ;BR IF YES
 9838  047534  000207                          RTS     PC
 9839
 9840                                  ;
 9841                                  ;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
 9842                                  ;IT IS USED TO ALLOW THE OPERATOR TO HALT THE CPU WHILE INSURING
 9843                                  ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
 9844                                  ;THE CPU.
 9845                                  ;
 9846  047536  022626                  STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT
 9847
 9848  047540  004737  045462                  JSR     PC,SUBCLR
 9849  047544  104024                          ERROR   24              ;CERR AFTER
 9850
 9851  047546  005737  003304                  TST     UNLD            ;SEE IF HEADS UNLOADED
 9852  047552  001431                          BEQ     3$              ;BR IF NO
 9853  047554  005737  000042                  TST     42              ;SEE IF MANUAL OR AUTO MODE
 9854  047560  001403                          BEQ     1$              ;BR IF MANUAL MODE
 9855  047562  104401  057041                  TYPE    ,MSG74          ;PGM ABORT PENDING
 9856  047566  000402                          BR      2$
 9857  047570  104401  057070          1$:     TYPE    ,MSG75          ;HALT PENDING
 9858  047574                          2$:
 9859
 9860  047574  004737  045462                  JSR     PC,SUBCLR
 9861  047600  104024                          ERROR   24              ;CERR AFTER SCLR
 9862
 9863  047602  012737  000011  003322          MOV     #SRTSPL,HCS1
 9864  047610  004737  043472                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
 9865  047614  104121                          ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
 9866
 9867  047616  013737  001420  003362          MOV     T500,TEMP2      ;SETUP  TIMEOUT
 9868  047624  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
 9869  047630  104067                          ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
 9870
 9871  047632  005037  003304                  CLR     UNLD
 9872
 9873  047636  005737  003306          3$:     TST     BADHDR          ;SEE IF HEADERS VALID
 9874  047642  001460                          BEQ     4$              ;BR IF YES
 9875  047644  005237  003310                  INC     HPEND
 9876
 9877  047650  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 9878  047656  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 9879  047664  012737  000013  003322          MOV     #RECAL,HCS1
 9880  047672  004737  043472                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
 9881  047676  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
 9882
 9883  047700  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 9884  047706  004737  045132                  JSR     PC,GSTAT
```

```
 9885  047712  032737  020000  003350          BIT     #D.RTZ,HMR2
 9886  047720  001001                          BNE     64$
 9887  047722  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
 9888  047724  013737  001414  003362  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
 9889  047732  004737  044102                  JSR     PC,FATT1        ;FIND ATTN
 9890  047736  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
 9891
 9892  047740  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 9893  047746  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 9894  047754  012737  000005  003322          MOV     #CLEAR,HCS1
 9895  047762  004737  043472                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 9896  047766  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 9897  047770  004737  044050                  JSR     PC,TSTATN       ;TEST FOR ATTN
 9898  047774  000401                          BR      66$
 9899  047776  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 9900  050000                          66$:
 9901
 9902
 9903  050000  000137  031366                  JMP     FORM            ;WRITE VALID FORMATS
 9904
 9905  050004  005737  000042          4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
 9906  050010  001410                          BEQ     5$              ;BR IF MANUAL MODE
 9907  050012  104401  057112                  TYPE    ,MSG76          ;PGM ABORTED
 9908  050016  005037  043006                  CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
 9909  050022  005037  001176                  CLR     $ESCAPE
 9910  050026  000137  042760                  JMP     $EOP1           ;ABORT PROGRAM
 9911
 9912  050032  104401  057130          5$:     TYPE    ,MSG77          ;CPU HALTED
 9913  050036  000000                          HALT
 9914  050040  000137  007744                  JMP     ST5             ;START OVER IF CONTINUE PRESSED
 9915
 9916                                  ;
 9917                                  ;CHECK IF HALT PENDING
 9918                                  ;RET IF YES
 9919                                  ;RET+4 IF NO
 9920                                  ;
 9921  050044  005037  003306          HPEN:   CLR     BADHDR          ;CLR VALID HALT FLAG
 9922  050050  005737  003310                  TST     HPEND           ;SEE IF HALT PENDING
 9923  050054  001002                          BNE     1$              ;BR IF YES
 9924  050056  062716  000004                  ADD     #4,(SP)         ;ELSE BUMP RET ADDR
 9925  050062  000207                  1$:     RTS     PC              ;& RET
 9926
 9927                                  .SBTTL  UNEXPECTED TIMEOUT HANDLER
 9928
 9929                                  ;
 9930                                  ;THIS ROUTINE IS ENTERED IF THERE IS
 9931                                  ;       A.  NON EXISTANT MEMORY (NO SSYN)
 9932                                  ;       B.  BOUNDRY ERROR
 9933                                  ;       C.  STACK OVERFLOW
 9934                                  ;
 9935
 9936  050064  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
 9937  050066  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE
 9938  050070  032777  020000  131042          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPOUT?
 9939  050076  001005                          BNE     1$              ;YES, DON'T TYPE
 9940  050100  104401  057306                  TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
```

```
 9941  050104  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 9942                                                                  ;;TYPE PC
 9943  050106  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 9944  050110     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
 9945  050111     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 9946  050112  032777  001000  131020  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
 9947  050120  001403                          BEQ     2$              ;NO, BRANCH
 9948  050122  022626                          CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
 9949  050124  000177  130756                  JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
 9950                                                                  ;THAT GAVE BAD TIMEOUT
 9951  050130  032777  040000  131002  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
 9952  050136  001401                          BEQ     3$              ;NO BRANCH
 9953  050140  000002                          RTI                     ;YES
 9954
 9955  050142  000000                  3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
 9956                                                                  ;AS INDICATED. YOU CAN LOOP ON
 9957                                                                  ;ERROR, LOOP ON TEST OR INHIBIT
 9958                                                                  ;ERROR TYPEOUT BY SETTING THOSE
 9959                                                                  ;SWITCHES.
 9960                                                                  ;
 9961  050144  ^22626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
 9962  050146  000137  042760                  JMP     $EOP1           ;ABORT TESTS
 9963
 9964                          .SBTTL  MEMORY CHECK ENABLE TRAP
 9965
 9966  050152  012737  050166  001176  MEMERR: MOV     #1$,$ESCAPE     ;LOAD ESCAPE
 9967  050160  011637  001334                  MOV     (SP),TRAPPC     ;STORE PC
 9968  050164  104236                          ERROR   236             ;UNEXP MEM PARITY TRAP
 9969
 9970  050166  005037  001176          1$:     CLR     $ESCAPE
 9971  050172  032777  001000  130740          BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
 9972  050200  001001                          BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
 9973  050202  000002                          RTI                     ;ELSE RETURN
 9974
 9975  050204  012706  001100          2$:     MOV     #STACK,SP       ;INIT STACK
 9976  050210  000177  130674                  JMP     @$LPERR         ;LOOP ON ERROR
 9977
 9978                          .SBTTL  RK06 INTERRUPT HANDLER
 9979
 9980  050214  011600                  INTER:  MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
 9981  050216  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE.
 9982  050220  104401  055774                  TYPE    ,MSG6           ;INT AT PC=
 9983  050224  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 9984                                                                  ;;TYPE PC
 9985  050226  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 9986  050230     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
 9987  050231     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 9988  050232  000002                          RTI
 9989
 9990                          .SBTTL  POWER DOWN AND UP ROUTINES
 9991
 9992                          ;POWER DOWN ROUTINE
 9993
 9994  050234  012737  050246  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
 9995  050242  000000                          HALT
 9996  050244  000776                          BR      .-2             ;HANG UP.
```

```
 9997
 9998                                   ;POWER UP ROUTINE
 9999
10000  050246  005037  050320          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
10001  050252  005237  050320          1$:     INC     $PWRCT          ;WAIT FOR THE INCR
10002  050256  001375                          BNE     1$              ;OF WORD
10003  050260  012737  050234  000024          MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
10004  050266  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;PRIORITY 7
10005  050274  012737  000340  000036          MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
10006  050302  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
10007  050306  104401  056162                  TYPE    ,MSG11          ;REPORT POWER FAIL
10008  050312  000005                          RESET
10009  050314  000137  012214                  JMP     PFSRT
10010
10011  050320  000000                  $PWRCT: 0                       ;WAIT COUNT FOR TTY
10012
```

```
                        .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

050322                  $SCOPE:
050322  104407                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
050324  032777  040000  130606 1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
050332  001114                 BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
050334  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
050336  013746  000004         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
050342  012737  050362  000004 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
050350  005737  177060         TST     @#177060        ;;TIME OUT ON XOR?
050354  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050360  000463                 BR      $SVLAD          ;;GO TO THE NEXT TEST
050362  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
050364  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050370  000423                 BR      7$              ;;LOOP ON THE PRESENT TEST
050372                  6$:;#####END OF CODE FOR THE XOR TESTER#####
050372  032777  000400  130540         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
050400  001404                 BEQ     2$              ;;BR IF NO
050402  127737  130532  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
050410  001465                 BEQ     $OVER           ;;BR IF YES
050412  105737  001103  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
050416  001421                 BEQ     3$              ;;BR IF NO
050420  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
050426  101015                 BHI     3$              ;;BR IF NO
050430  032777  001000  130502         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
050436  001404                 BEQ     4$              ;;BR IF NO
050440  013737  001110  001106 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
050446  000446                 BR      $OVER
050450  105037  001103  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
050454  005037  001174         CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
050460  000415                 BR      1$              ;;ESCAPE TO THE NEXT TEST
050462  032777  004000  130450 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
050470  001011                 BNE     1$              ;;BR IF YES
050472  005737  001216         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
050476  001406                 BEQ     1$              ;;      INHIBIT ITERATIONS
050500  005237  001104         INC     $ICNT           ;;INCREMENT ITERATION COUNT
050504  023737  001174  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
050512  002024                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
050514  012737  000001  001104 1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
050522  013737  050600  001174         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
050530  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
050534  113737  001102  001214         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
10069  050542  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
10070  050546  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
10071  050552  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
10072  050556  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10073  050564  013777  001102  130350  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
10074  050572  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
10075  050576  000002                          RTI                     ;;FIXES PS
10076  050600  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
10077                                   .SBTTL  ERROR HANDLER ROUTINE
10078
10079                                   ;;**************************************************************************
10080                                   ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
10081                                   ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10082                                   ;*AND GO TO TYPERR ON ERROR
10083                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10084                                   ;*SW15=1        HALT ON ERROR
10085                                   ;*SW13=1        INHIBIT ERROR TYPEOUTS
10086                                   ;*SW10=1        BELL ON ERROR
10087                                   ;*SW09=1        LOOP ON ERROR
10088                                   ;*CALL
10089                                   ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
10090
10091  050602                           $ERROR:
10092  050602  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
10093  050604  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
10094  050610  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
10095  050612  013777  001102  130322          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
10096  050620  032777  002000  130312          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
10097  050626  001402                          BEQ     1$              ;;NO - SKIP
10098  050630  104401  001200                  TYPE    ,$BELL          ;;RING BELL
10099  050634  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
10100  050640  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
10101  050644  162737  000002  001116          SUB     #2,$ERRPC
10102  050652  117737  130240  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
10103  050660  032777  020000  130252          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
10104  050666  001004                          BNE     20$             ;;SKIP TYPEOUTS
10105  050670  004737  067604                  JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
10106  050674  104401  001205                  TYPE    ,$CRLF
10107  050700                          20$:
10108  050700  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
10109  050706  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
10110  050710  113737  001114  050722          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
10111  050716  004737  051600                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
10112  050722     000                  21$:    .BYTE   0
10113  050723     000                          .BYTE   0
10114  050724  000777                  22$:    BR      22$             ;;APT ERROR LOOP
10115  050726  005777  130206          2$:     TST     @SWR            ;;HALT ON ERROR
10116  050732  100002                          BPL     3$              ;;SKIP IF CONTINUE
10117  050734  000000                          HALT                    ;;HALT ON ERROR!
10118  050736  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
10119  050740  032777  001000  130172  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
10120  050746  001402                          BEQ     4$              ;;BR IF NO
10121  050750  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
10122  050754  005737  001176          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
10123  050760  001402                          BEQ     5$              ;;BR IF NONE
10124  050762  013716  001176                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
10125  050766                          5$:
10126  050766  022737  043046  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
10127  050774  001001                          BNE     6$              ;;BRANCH IF NO
10128  050776  000000                          HALT                    ;;YES
10129  051000                          6$:
10130  051000  000002                          RTI                     ;;RETURN
10131                                  .SBTTL  TYPE ROUTINE
10132
10133                                  ;;**********************************************************************
10134                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10135                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10136                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10137                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10138                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10139                                  ;*
10140                                  ;*CALL:
10141                                  ;*1) USING A TRAP INSTRUCTION
10142                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10143                                  ;*OR
10144                                  ;*      TYPE
10145                                  ;*      MESADR
10146                                  ;*
10147
10148  051002  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
10149  051006  100002                          BPL     1$              ;;BR IF YES
10150  051010  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
10151  051012  000430                          BR      3$              ;;LEAVE
10152  051014  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
10153  051016  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
10154  051022  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
10155  051030  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
10156  051032  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
10157  051040  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
10158  051042  010037  051052                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
10159  051046  004737  051570                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
10160  051052  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
10161  051054  132737  000040  001231  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
10162  051062  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
10163  051064  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
10164  051066  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
10165  051070  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
10166  051072  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
10167  051074  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
10168  051100  000002                          RTI                     ;;RETURN
10169  051102  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
10170  051106  001430                          BEQ     8$
10171  051110  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
10172  051114  001006                          BNE     5$
10173  051116  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
10174  051120  104401                          TYPE                    ;;TYPE A CR AND LF
10175  051122  001205                          $CRLF
10176  051124  105037  051332                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
10177  051130  000755                          BR      2$              ;;GET NEXT CHARACTER
10178  051132  004737  051214          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
10179  051136  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
10180  051142  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
10181  051144  013746  001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
10182                                                          ;;AND THE NULL CHAR.
10183  051150  105366  000001    7$:    DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
10184  051154  002770                   BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
10185  051156  004737  051214          JSR     PC,$TYPEC       ;;GO TYPE A NULL
10186  051162  105337  051332          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
10187  051166  000770                   BR      7$              ;;LOOP
10188
10189                                  ;HORIZONTAL TAB PROCESSOR
10190
10191  051170  112716  000040    8$:    MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
10192  051174  004737  051214    9$:    JSR     PC,$TYPEC       ;;TYPE A SPACE
10193  051200  132737  000007  051332   BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
10194  051206  001372                   BNE     9$              ;;TAB STOP
10195  051210  005726                   TST     (SP)+           ;;POP SPACE OFF STACK
10196  051212  000724                   BR      2$              ;;GET NEXT CHARACTER
10197  051214                  $TYPEC:
10198  051214  105777  127724          TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
10199  051220  100022                   BPL     10$             ;;BR IF NOT                     ;MJD001
10200  051222  017746  127720          MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
10201  051226  042716  177600          BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
10202  051232  122716  000023          CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
10203  051236  001012                   BNE     102$            ;;BR IF NOT                     ;MJD001
10204  051240                  101$:                                                           ;MJD001
10205  051240  105777  127700          TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
10206  051244  100375                   BPL     101$                                            ;MJD001
10207  051246  117716  127674          MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
10208  051252  042716  177600          BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
10209  051256  122716  000021          CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
10210  051262  001366                   BNE     101$            ;;BR IF NOT                     ;MJD001
10211  051264                  102$:                                                           ;MJD001
10212  051264  005726                   TST     (SP)+           ;;FIX STACK                     ;MJD001
10213  051266                  10$:                                                            ;MJD001
10214  051266  105777  127656          TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
10215  051272  100375                   BPL     10$                                             ;MJD001
10216  051274  116677  000002  127650   MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
10217  051302  122766  000015  000002   CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
10218  051310  001003                   BNE     1$              ;;BRANCH IF NO
10219  051312  105037  051332          CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
10220  051316  000406                   BR      $TYPEX          ;;EXIT
10221  051320  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
10222  051326  001402                   BEQ     $TYPEX          ;;BRANCH IF YES
10223  051330  105227                   INCB    (PC)+           ;;COUNT THE CHARACTER
10224  051332  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
10225  051334  000207          $TYPEX: RTS     PC
10226
10227                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10228
10229                                  ;;**************************************************************
10230                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
10231                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
10232                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
10233                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
10234                                  ;*REPLACED WITH SPACES.
10235                                  ;*CALL:
10236                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
```

```
                                ;*      TYPDS                   ;;GO TO THE ROUTINE
                                
                                $TYPDS:
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                        MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
                                        MOV     20(SP),R5       ;;GET THE INPUT NUMBER
                                        BPL     1$              ;;BR IF INPUT IS POS.
                                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
                                        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
                                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
                                        MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
                                        MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
                                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
                                        MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
                                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
                                        BLT     4$              ;;BR IF DONE
                                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
                                        BR      3$
                                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
                                        TST     R2              ;;CHECK IF BCD DIGIT=0
                                        BNE     5$              ;;FALL THROUGH IF 0
                                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
                                        BMI     7$              ;;BR IF YES
                                5$:     ASLB    (SP)            ;;MSD?
                                        BCC     6$              ;;BR IF NO
                                        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
                                6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
                                7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
                                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
                                        TST     (R0)+           ;;JUST INCREMENTING
                                        CMP     R0,#10          ;;CHECK THE TABLE INDEX
                                        BLT     2$              ;;GO DO THE NEXT DIGIT
                                        BGT     8$              ;;GO TO EXIT
                                        MOV     R5,R2           ;;GET THE LSD
                                        BR      6$              ;;GO CHANGE TO ASCII
                                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
                                        BPL     9$              ;;BR IF NO
                                        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
                                        MOV     (SP)+,R5        ;;POP STACK INTO R5
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
                                        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
                                        MOV     (SP)+,(SP)
                                        RTI                     ;;RETURN TO USER
                                $DTBL:  10000.
                                        1000.
                                        100.
                                        10.
```

```
10293  051552  000004                  $DBLK:  .BLKW   4
10294                                  .SBTTL  APT COMMUNICATIONS ROUTINE
10295
10296                                  ;;*****************************************************************
10297  051562  112737  000001  052026  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
10298  051570  112737  000001  052024  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
10299  051576  000403                          BR      $ATYC
10300  051600  112737  000001  052026  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
10301  051606                          $ATYC:
10302  051606  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10303  051610  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10304  051612  105737  052024                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
10305  051616  001450                          BEQ     5$              ;;IF NOT:  BR
10306  051620  122737  000001  001230          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
10307  051626  001031                          BNE     3$              ;;IF NOT:  BR
10308  051630  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
10309  051636  001425                          BEQ     3$              ;;IF NOT:  BR
10310  051640  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
10311  051644  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10312  051652  005737  001210          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
10313  051656  001375                          BNE     1$              ;;IF NOT:  WAIT
10314  051660  010037  001224                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
10315  051664  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
10316  051666  001376                          BNE     2$
10317  051670  163700  001224                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
10318  051674  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
10319  051676  010037  001226                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
10320  051702  012737  000004  001210          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
10321  051710  000413                          BR      5$
10322  051712  017637  000004  051736  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
10323  051720  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
10324  051726  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
10325  051732  004737  051002                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
10326  051736  000000                  4$:     .WORD   0
10327  051740                          5$:
10328  051740  105737  052026          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
10329  051744  001416                          BEQ     12$             ;;IF NOT:  BR
10330  051746  005737  001230                  TST     $ENV            ;;RUNNING UNDER APT?
10331  051752  001413                          BEQ     12$             ;;IF NOT:  BR
10332  051754  005737  001210          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
10333  051760  001375                          BNE     11$             ;;IF NOT:  WAIT
10334  051762  017637  000004  001212          MOV     @4(SP),$FATAL   ;;GET ERROR #
10335  051770  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10336  051776  005237  001210                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
10337  052002  105037  052026          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
10338  052006  105037  052025                  CLRB    $LFLG           ;;CLEAR LOG FLAG
10339  052012  105037  052024                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
10340  052016  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10341  052020  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10342  052022  000207                          RTS     PC              ;;RETURN
10343  052024     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
10344  052025     000          $LFLG:  .BYTE   0               ;;LOG FLAG
10345  052026     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
10346          052030                  .EVEN
10347          000200          APTSIZE=200
10348          000001          APTENV=001
```

```
 10349           000100                  APTSPOOL=100
 10350           000040                  APTCSUP=040
 10351                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 10352
 10353                                   ;;******************************************************************
 10354                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 10355                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 10356                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 10357                                   ;*CALL:
 10358                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 10359                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
 10360                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 10361                                   ;*      .BYTE   M               ;;M=1 OR 0
 10362                                   ;*                                      ;;1=TYPE LEADING ZEROS
 10363                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
 10364                                   ;*
 10365                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 10366                                   ;*$TYPOS OR $TYPOC
 10367                                   ;*CALL:
 10368                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 10369                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
 10370                                   ;*
 10371                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 10372                                   ;*CALL:
 10373                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 10374                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
 10375
 10376  052030  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 10377  052034  116637  000001  052253          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 10378  052042  112637  052255                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 10379  052046  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 10380  052052  000406                          BR      $TYPON
 10381  052054  112737  000001  052253  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 10382  052062  112737  000006  052255          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 10383  052070  112737  000005  052252  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 10384  052076  010346                          MOV     R3,-(SP)        ;;SAVE R3
 10385  052100  010446                          MOV     R4,-(SP)        ;;SAVE R4
 10386  052102  010546                          MOV     R5,-(SP)        ;;SAVE R5
 10387  052104  113704  052255                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 10388  052110  005404                          NEG     R4
 10389  052112  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 10390  052116  110437  052254                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 10391  052122  113704  052253                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 10392  052126  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 10393  052132  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 10394  052134  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 10395  052136  000404                          BR      3$              ;;GO DO MSB
 10396  052140  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 10397  052142  006105                          ROL     R5
 10398  052144  006105                          ROL     R5
 10399  052146  010503                          MOV     R5,R3
 10400  052150  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 10401  052152  105337  052254                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 10402  052156  100016                          BPL     7$              ;;BR IF NO
 10403  052160  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 10404  052164  001002                          BNE     4$              ;;TEST FOR 0
```

```
10405  052166  005704                        TST     R4              ;;SUPPRESS THIS 0?
10406  052170  001403                        BEQ     5$              ;;BR IF YES
10407  052172  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
10408  052174  052703  000060              BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
10409  052200  052703  000040        5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
10410  052204  110337  052250              MOVB    R3,8$           ;;SAVE FOR TYPING
10411  052210  104401  052250              TYPE    ,8$             ;;GO TYPE THIS DIGIT
10412  052214  105337  052252        7$:   DECB    $OCNT           ;;COUNT BY 1
10413  052220  003347                        BGT     2$              ;;BR IF MORE TO DO
10414  052222  002402                        BLT     6$              ;;BR IF DONE
10415  052224  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
10416  052226  000744                        BR      2$              ;;GO DO THE LAST DIGIT
10417  052230  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
10418  052232  012604                        MOV     (SP)+,R4        ;;RESTORE R4
10419  052234  012603                        MOV     (SP)+,R3        ;;RESTORE R3
10420  052236  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
10421  052244  012616                        MOV     (SP)+,(SP)
10422  052246  000002                        RTI                     ;;RETURN
10423  052250     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
10424  052251     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
10425  052252     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
10426  052253     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
10427  052254  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
10428                                  .SBTTL  TTY INPUT ROUTINE
10429
10430                                  ;;******************************************************************
10431                                  .ENABL  LSB
10432  052256  000000                $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
10433  052260  000000                $TKQIN: .WORD   0               ;;INPUT POINTER
10434  052262  000000                $TKQOUT: .WORD  0               ;;OUTPUT POINTER
10435  052264  000001                $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
10436           052265                $TKQEND=.
10437           052266                .EVEN
10438
10439                                  ;*TK INITIALIZE ROUTINE
10440                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
10441                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
10442                                  ;
10443                                  ;*CALL:
10444                                  ;*      JSR     PC,$TKINT
10445                                  ;*      RETURN
10446                                  ;
10447  052266  005037  052256          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
10448  052272  012737  052264  052260          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
10449  052300  013737  052260  052262          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
10450  052306  012737  052336  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
10451  052314  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
10452  052322  005777  126620                  TST     @$TKB           ;;CLEAR DONE FLAG
10453  052326  012777  000100  126610          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
10454  052334  000207                          RTS     PC              ;;RETURN TO CALLER
10455
10456                                  ;*TK SERVICE ROUTINE
10457                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
10458                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
10459                                  ;*IT IN THE QUEUE.
10460                                  ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
```

```
10461                                   ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
10462                                   ;
10463  052336  117746  126604         $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
10464  052342  042716  177600                 BIC     #^C177,(SP)     ;;STRIP THE JUNK
10465  052346  021627  000021                 CMP     (SP),#$XON      ;;IS IT A RANDOM XON?                   ;RAN001
10466  052352  001002                         BNE     30$             ;;BRANCH IF NO                          ;RAN001
10467  052354  005726                         TST     (SP)+           ;;CLEAN RANDOM XON OFF STACK            ;RAN001
10468  052356  000002                         RTI                     ;;RETURN                                ;RAN001
10469  052360                          30$:                                                                    ;RAN001
10470  052360  021627  000003                 CMP     (SP),#3         ;;IS IT A CONTROL C?
10471  052364  001007                         BNE     1$              ;;BRANCH IF NO
10472  052366  104401  053476                 TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
10473  052372  004737  052266                 JSR     PC,$TKINT       ;;INIT THE KEYBOARD
10474  052376  005726                         TST     (SP)+           ;;CLEAN UP STACK
10475  052400  000137  047536                 JMP     STOP            ;;CONTROL C RESTART
10476  052404  021627  000007         1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
10477  052410  001004                         BNE     2$              ;;BRANCH IF NO
10478  052412  022737  000176  001140         CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
10479  052420  001500                         BEQ     6$              ;;GO TO SWR CHANGE
10480
10481  052422                          2$:
10482  052422  022737  000001  052256         CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
10483  052430  001004                         BNE     3$              ;;BRANCH IF NO
10484  052432  104401  001200                 TYPE    ,$BELL          ;;RING THE TTY BELL
10485  052436  005726                         TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
10486  052440  000451                         BR      5$              ;;EXIT
10487  052442  021627  000023         3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
10488  052446  001021                         BNE     32$             ;;BRANCH IF NO
10489  052450  005077  126470                 CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
10490  052454  005726                         TST     (SP)+           ;;CLEAN CHAR OFF STACK
10491  052456  105777  126462         31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
10492  052462  100375                         BPL     31$             ;;LOOP UNTIL ITS THERE
10493  052464  117746  126456                 MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
10494  052470  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10495  052474  022627  000021                 CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
10496  052500  001366                         BNE     31$             ;;BRANCH IF NO
10497  052502  012777  000100  126434         MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
10498  052510  000002                         RTI                     ;;RETURN
10499  052512  005237  052256         32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
10500  052516  021627  000140                 CMP     (SP),#140       ;;IS IT UPPER CASE?
10501  052522  002405                         BLT     4$              ;;BRANCH IF YES
10502  052524  021627  000175                 CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
10503  052530  003002                         BGT     4$              ;;BRANCH IF YES
10504  052532  042716  000040                 BIC     #40,(SP)        ;;MAKE IT UPPER CASE
10505  052536  112677  177516         4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
10506  052542  005237  052260                 INC     $TKQIN          ;;UPDATE THE POINTER
10507  052546  023727  052260  052265         CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
10508  052554  001003                         BNE     5$              ;;BRANCH IF NO
10509  052556  012737  052264  052260         MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
10510  052564  000002                 5$:     RTI                     ;;RETURN
10511
10512                                   ;;*****************************************************************
10513                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
10514                                   ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
10515                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
10516                                   ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
```

```
10517  052566  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
10518  052574  001124                          BNE     15$             ;;EXIT IF NOT
10519  052576  105777  126342                  TSTB    @$TKS           ;;IS A CHAR WAITING?
10520  052602  100121                          BPL     15$             ;;IF NOT, EXIT
10521  052604  117746  126336                  MOVB    @$TKB,-(SP)     ;;YES
10522  052610  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
10523  052614  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
10524  052620  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
10525                                                                  ;;AND EXIT
10526
10527                                  ;;**************************************************************
10528                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
10529                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
10530                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
10531  052622  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
10532  052630  001674                          BEQ     2$              ;;BRANCH IF YES
10533  052632  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
10534  052634  004737  052266                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
10535  052640  005077  126300                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
10536  052644  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
10537
10538  052652  104401  053510                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
10539  052656  104401  053515          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
10540  052662  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
10541  052666  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
10542  052670  104401  053526                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
10543  052674  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
10544  052676  005046                          CLR     -(SP)           ;;THE NEW SWR
10545  052700  105777  126240          7$:     TSTB    @$TKS           ;;CHAR THERE?
10546  052704  100375                          BPL     7$              ;;IF NOT TRY AGAIN
10547
10548  052706  117746  126234                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
10549  052712  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10550
10551  052716  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
10552  052722  001015                          BNE     9$              ;;BRANCH IF NOT
10553  052724  104401  053476                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
10554  052730  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
10555  052734  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
10556  052742  001003                          BNE     8$              ;;BRANCH IF NO
10557  052744  012777  000100  126172          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
10558  052752  000137  047536          8$:     JMP     STOP            ;;CONTROL-C RESTART
10559
10560
10561  052756  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
10562  052762  001005                          BNE     10$             ;;BRANCH IF NOT
10563  052764  104401  053503                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
10564  052770  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
10565  052774  000737                          BR      19$             ;;LET'S TRY IT AGAIN
10566
10567
10568  052776  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
10569  053002  001022                          BNE     16$             ;;BRANCH IF NO
10570  053004  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
10571  053010  001403                          BEQ     11$             ;;BRANCH IF YES
10572  053012  016677  000002  126120          MOV     2(SP),@SWR      ;;SAVE NEW SWR
```

```
10573  053020  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
10574  053024  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
10575  053030  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
10576  053036  001003                          BNE     15$             ;;BRANCH IF NOT
10577  053040  012777  000100  126076          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
10578  053046  000002                  15$:    RTI                     ;;RETURN
10579  053050  004737  051214          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
10580  053054  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
10581  053060  002420                          BLT     18$             ;;BRANCH IF YES
10582  053062  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
10583  053066  003015                          BGT     18$             ;;BRANCH IF YES
10584  053070  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
10585  053074  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
10586  053100  001403                          BEQ     17$             ;;BRANCH IF YES
10587  053102  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
10588  053104  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
10589  053106  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
10590  053110  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
10591  053114  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
10592  053120  000667                          BR      7$              ;;GET THE NEXT ONE
10593  053122  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
10594  053126  000720                          BR      20$             ;;SIMULATE CONTROL-U
10595                                  .DSABL  LSB
10596
10597
10598                                  ;;********************************* *************************************
10599                                  ;*THIS ROUTINE WILL INPUT A S NGLE CHARACTER FROM THE TTY
10600                                  ;*CALL:
10601                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
10602                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10603                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
10604                                  ;
10605
10606  053130  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
10607  053132  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
10608  053140  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
10609  053144  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
10610  053146  012746  053154                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
10611  053152  000002                          RTI                     ;;POP NEW PC AND PS
10612  053154                          64$:
10613  053154  005737  052256          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
10614  053160  001775                          BEQ     1$
10615  053162  005337  052256                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
10616  053166  117766  177070  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
10617  053174  005237  052262                  INC     $TKQOUT         ;;UPDATE THE POINTER
10618  053200  023727  052262  052265          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
10619  053206  001003                          BNE     2$              ;;BRANCH IF NO
10620  053210  012737  052264  052262          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
10621  053216  000002          2$:     RTI                     ;;RETURN
10622                                  ;;*****************************************************************
10623                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10624                                  ;*CALL:
10625                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10626                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10627                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10628
```

```
10629  053220  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
10630  053222  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
10631  053224  012703  053454            1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
10632  053230  022703  053476            2$:   CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
10633  053234  101456                          BLOS    4$              ;;BR IF YES
10634  053236  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10635  053240  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
10636  053242  122713  000177            10$:  CMPB    #177,(R3)       ;;IS IT A RUBOUT
10637  053246  001022                          BNE     5$              ;;BR IF NO
10638  053250  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
10639  053252  001007                          BNE     6$              ;;BR IF NO
10640  053254  112737  000134  053452        MOVB    #'\,9$          ;;TYPE A BACK SLASH
10641  053262  104401  053452                TYPE    ,9$
10642  053266  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
10643  053272  005303                  6$:   DEC     R3              ;;BACKUP BY ONE
10644  053274  020327  053454                CMP     R3,#$TTYIN      ;;STACK EMPTY?
10645  053300  103434                        BLO     4$              ;;BR IF YES
10646  053302  111337  053452                MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
10647  053306  104401  053452                TYPE    ,9$             ;;GO TYPE
10648  053312  000746                        BR      2$              ;;GO READ ANOTHER CHAR.
10649  053314  005716                  5$:   TST     (SP)            ;;RUBOUT KEY SET?
10650  053316  001406                        BEQ     7$              ;;BR IF NO
10651  053320  112737  000134  053452        MOVB    #'\,9$          ;;TYPE A BACK SLASH
10652  053326  104401  053452                TYPE    ,9$
10653  053332  005016                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
10654  053334  122713  000025            7$:   CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
10655  053340  001003                        BNE     8$              ;;BR IF NO
10656  053342  104401  053503                TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U'
10657  053346  000726                        BR      1$              ;;GO START OVER
10658  053350  122713  000022            8$:   CMPB    #22,(R3)        ;;IS CHARACTER A '^R'?
10659  053354  001011                        BNE     3$              ;;BRANCH IF NO
10660  053356  105013                        CLRB    (R3)            ;;CLEAR THE CHARACTER
10661  053360  104401  001205                TYPE    ,$CRLF          ;;TYPE A 'CR' & 'LF'
10662  053364  104401  053454                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
10663  053370  000717                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
10664  053372  104401  001204            4$:   TYPE    ,$QUES          ;;TYPE A '?'
10665  053376  000712                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
10666  053400  111337  053452            3$:   MOVB    (R3),9$         ;;ECHO THE CHARACTER
10667  053404  104401  053452                TYPE    ,9$
10668  053410  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10669  053414  001305                        BNE     2$              ;;LOOP IF NOT RETURN
10670  053416  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10671  053422  104401  001206                TYPE    ,$LF            ;;TYPE A LINE FEED
10672  053426  005726                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
10673  053430  012603                        MOV     (SP)+,R3        ;;RESTORE R3
10674  053432  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10675  053434  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10676  053442  012766  053454  000004        MOV     #$TTYIN,4(SP)
10677  053450  000002                        RTI                     ;;RETURN
10678  053452     000                  9$:   .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
10679  053453     000                        .BYTE   0               ;;TERMINATOR
10680  053454  000022            $TTYIN: .BLKB   22                  ;;RESERVE 22 BYTES FOR TTY INPUT
10681  053476  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL 'C'
10682  053503     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U'
10683  053510  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL 'G'
10684  053515     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
```

```
10685  053522  036440  000040
10686  053526  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
10687  053534  020075     000
10688          053540                  .EVEN
10689                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
10690
10691                                  ;;*****************************************************************
10692                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
10693                                  ;*CHANGE IT TO BINARY.
10694                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
10695                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
10696                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
10697                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
10698                                  ;*CALL:
10699                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
10700                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
10701                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
10702
10703  053540  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
10704  053542  016666  000004  000002  MOV     4(SP),2(SP)     ;;INPUT NUMBER
10705  053550  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10706  053552  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10707  053554  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10708  053556  104411                  1$:     RDLIN           ;;READ AN ASCIZ LINE
10709  053560  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
10710  053562  010037  053666          MOV     R0,5$           ;;AND SAVE IT
10711  053566  005001                  CLR     R1              ;;CLEAR DATA WORD
10712  053570  005002                  CLR     R2
10713  053572  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
10714  053574  001420                  BEQ     3$              ;;IF ZERO GET OUT
10715  053576  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
10716  053602  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
10717  053604  122716  000067          CMPB    #'7,(SP)
10718  053610  002423                  BLT     4$
10719  053612  006301                  ASL     R1              ;;*2
10720  053614  006102                  ROL     R2
10721  053616  006301                  ASL     R1              ;;*4
10722  053620  006102                  ROL     R2
10723  053622  006301                  ASL     R1              ;;*8
10724  053624  006102                  ROL     R2
10725  053626  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
10726  053632  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
10727  053634  000756                  BR      2$              ;;LOOP
10728  053636  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
10729  053640  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
10730  053644  010237  053676          MOV     R2,$HIOCT
10731  053650  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10732  053652  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10733  053654  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10734  053656  000002                  RTI                     ;;RETURN
10735  053660  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
10736  053662  105010                  CLRB    (R0)            ;;SET A TERMINATOR
10737  053664  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
10738  053666  000000          5$:     .WORD   0
10739  053670  104401  001204        TYPE    ,$QUES          ;;"?" "CR" & "LF"
10740  053674  000730                  BR      1$              ;;TRY AGAIN
```

```
10741  053676  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
10742                                  .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
10743
10744                                  ;;**************************************************************************
10745                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
10746                                  ;*UNSIGNED OCTAL ASCIZ NUMBER.
10747                                  ;*CALL
10748                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
10749                                  ;*      JSR     PC,@#$DB2O      ;;CALL THE ROUTINE
10750                                  ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
10751
10752
10753  053700  104413                  $DB2O:  SAVREG                  ;;SAVE ALL REGISTERS
10754  053702  016601  000002                  MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
10755  053706  012705  054017                  MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
10756  053712  012704  000014                  MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
10757  053716  012703  177770                  MOV     #^C7,R3         ;;MASK
10758  053722  012100                          MOV     (R1)+,R0        ;;LOWER WORD
10759  053724  012101                          MOV     (R1)+,R1        ;;HIGH WORD
10760  053726  005002                          CLR     R2              ;;TERMINATOR
10761  053730  110245                  1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
10762  053732  010002                          MOV     R0,R2           ;;GET THIS DIGIT
10763  053734  005304                          DEC     R4              ;;COUNT THIS CHARACTER
10764  053736  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
10765  053740  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
10766  053742  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
10767  053744  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
10768  053750  104414                          RESREG                  ;;RESTORE ALL REGISTERS
10769  053752  000207                          RTS     PC              ;;RETURN TO USER
10770  053754  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
10771  053756  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
10772  053760  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
10773  053762  006001                          ROR     R1
10774  053764  006000                          ROR     R0
10775  053766  006001                          ROR     R1
10776  053770  006000                          ROR     R0
10777  053772  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
10778  053774  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
10779  054000  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
10780  054002  000016                  $OCTVL: .BLKB   14.              ;;RESERVE DATA TABLE
10781                                  .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
10782
10783                                  ;;**************************************************************************
10784                                  ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
10785                                  ;*UNSIGNED DECIMAL ASCIZ NUMBER.
10786                                  ;*CALL
10787                                  ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
10788                                  ;*      JSR     PC,@#$SB2D      ;;CALL
10789                                  ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
10790
10791
10792  054020  016637  000002  054050   $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
10793  054026  012746  054050                  MOV     #1$,-(SP)       ;;SET POINTER
10794  054032  004737  054054                  JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
10795  054036  062716  000005                  ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
10796  054042  012666  000002                  MOV     (SP)+,2(SP)     ;;PICKUP POINTER
```

```
10797  054046  000207                          RTS     PC              ;;RETURN
10798  054050  000000  000000          1$:     .WORD   0,0
10799                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
10800
10801                                  ;;*****************************************************************
10802                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
10803                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
10804                                  ;*POSITIVE.
10805                                  ;*CALL
10806                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
10807                                  ;*      JSR     PC,@#$DB2D
10808                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
10809                                                                  ;;IS ON THE STACK
10810
10811
10812  054054  104413          $DB2D:  SAVREG                          ;;SAVE REGISTERS
10813  054056  016602  000002          MOV     2(SP),R2                ;;PICKUP THE DATA POINTER
10814  054062  012700  054234          MOV     #$DECVL,R0              ;;GET ADDRESS OF "$DECVL" STRING
10815  054066  010066  000002          MOV     R0,2(SP)                ;;PUT ADDRESS OF ASCIZ STRING ON STACK
10816  054072  012201                  MOV     (R2)+,R1                ;;PICKUP THE BINARY NUMBER
10817  054074  012202                  MOV     (R2)+,R2
10818  054076  012737  000012  054152  MOV     #10.,4$                 ;;SET UP TO DO 10 CONVERSIONS
10819  054104  012704  054164          MOV     #$TNPWR,R4              ;;ADDRESS OF TEN POWER
10820  054110  012705  054166          MOV     #$TNPWR+2,R5
10821  054114  005003          1$:     CLR     R3                      ;;CLEAR PARTIAL
10822  054116  161401          2$:     SUB     (R4),R1                 ;;SUBTRACT TEN POWER
10823  054120  005602                  SBC     R2
10824  054122  161502                  SUB     (R5),R2
10825  054124  002402                  BLT     3$                      ;;BR IF TEN POWER TO LARGE
10826  054126  005203                  INC     R3                      ;;ADD 1 TO PARTIAL
10827  054130  000772                  BR      2$                      ;;LOOP
10828  054132  062401          3$:     ADD     (R4)+,R1                ;;RESTORE SUBTRACTED VALUE
10829  054134  005502                  ADC     R2
10830  054136  062402                  ADD     (R4)+,R2
10831  054140  022525                  CMP     (R5)+,(R5)+             ;;MOVE TO NEXT TEN POWER
10832  054142  052703  000060          BIS     #'0,R3                  ;;CHANGE PARTIAL TO ASCII
10833  054146  110320                  MOVB    R3,(R0)+                ;;SAVE IT
10834  054150  005327                  DEC     (PC)+                   ;;DONE?
10835  054152  000000          4$:     .WORD   0
10836  054154  001357                  BNE     1$                      ;;BR IF NO
10837  054156  105020                  CLRB    (R0)+                   ;;TERMINATOR
10838  054160  104414                  RESREG                          ;;RESTORE REGISTERS
10839  054162  000207                  RTS     PC                      ;;RETURN
10840  054164  145000          $TNPWR: 145000                          ;;1.0E09
10841  054166  035632                  35632
10842  054170  160400                  160400                          ;;1.0E08
10843  054172  002765                  2765
10844  054174  113200                  113200                          ;;1.0E07
10845  054176  000230                  230
10846  054200  041100                  041100                          ;;1.0E06
10847  054202  000017                  17
10848  054204  103240                  103240                          ;;1.0E05
10849  054206  000001                  1
10850  054210  023420                  23420                           ;;1.0E04
10851  054212  000000                  0
10852  054214  001750                  1750                            ;;1.0E03
```

```
10853  054216  000000                          0
10854  054220  000144                          144                     ;;1.0E02
10855  054222  000000                          0
10856  054224  000012                          12                      ;;1.0E01
10857  054226  000000                          0
10858  054230  000001                          1                       ;;1.0E00
10859  054232  000000                          0
10860  054234  000014                $DECVL: .BLKB   12.               ;;RESERVE STORAGE FOR ASCIZ STRING
10861                                .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
10862
10863                                ;;******************************************************************
10864                                ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
10865                                ;*LEADING NUMBERS.
10866                                ;*CALL
10867                                ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
10868                                ;*      JSR     PC,@#$SUPRS
10869
10870
10871  054250  010046                $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
10872  054252  016600  000004                MOV     4(SP),R0        ;;PICKUP THE POINTER
10873  054256  105710                1$:     TSTB    (R0)            ;;TERMINATEOR?
10874  054260  001403                        BEQ     2$              ;;BR IF YES
10875  054262  122720  000060              CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
10876  054266  001773                        BEQ     1$              ;;BR IF YES
10877  054270  005300                2$:     DEC     R0              ;;BACKUP BY "1"
10878  054272  010037  054300              MOV     R0,3$           ;;SAVE FOR TYPING
10879  054276  104401                        TYPE                    ;;GO TYPE
10880  054300  000000                3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
10881  054302  012600                        MOV     (SP)+,R0        ;;RESTORE R0
10882  054304  012616                        MOV     (SP)+,(SP)      ;;RESTORE THE STACK
10883  054306  000207                        RTS     PC              ;;RETURN
10884                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
10885
10886                                ;;******************************************************************
10887                                ;*SAVE R0-R5
10888                                ;*CALL:
10889                                ;*      SAVREG
10890                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
10891                                ;*
10892                                ;*TOP---(+16)
10893                                ;* +2---(+18)
10894                                ;* +4---R5
10895                                ;* +6---R4
10896                                ;* +8---R3
10897                                ;*+10---R2
10898                                ;*+12---R1
10899                                ;*+14---R0
10900
10901  054310                        $SAVREG:
10902  054310  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10903  054312  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10904  054314  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10905  054316  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10906  054320  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10907  054322  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10908  054324  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
```

```
10909  054330  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
10910  054334  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
10911  054340  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
10912  054344  000002                          RTI
10913
10914                                  ;*RESTORE R0-R5
10915                                  ;*CALL:
10916                                  ;*      RESREG
10917  054346                          $RESREG:
10918  054346  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
10919  054352  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
10920  054356  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
10921  054362  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
10922  054366  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10923  054370  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10924  054372  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10925  054374  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10926  054376  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10927  054400  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10928  054402  000002                          RTI
10929                                  .SBTTL  TRAP DECODER
10930
10931                                  ;;**************************************************************************
10932                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10933                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10934                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10935                                  ;*GO TO THAT ROUTINE.
10936
10937  054404  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
10938  054406  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
10939  054412  005740                          TST     -(R0)           ;;BACKUP BY 2
10940  054414  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
10941  054416  006300                          ASL     R0              ;;POSITION FOR INDEXING
10942  054420  016000  054440                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
10943  054424  000200                          RTS     R0              ;;GO TO ROUTINE
10944
10945
10946                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
10947
10948  054426  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
10949  054430  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
10950  054436  000002                          RTI                     ;;RESTORE THE PSW
10951
10952                                  .SBTTL  TRAP TABLE
10953
10954                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10955                                  ;*BY THE "TRAP" INSTRUCTION.
10956
10957                                  ;       ROUTINE
10958                                  ;       -------
10959  054440  054426                  $TRPAD: .WORD   $TRAP2
10960  054442  051002                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
10961  054444  052054                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10962  054446  052030                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
10963  054450  052070                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
10964  054452  051336                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
```

```
 10965
 10966  054454  052656                    $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 10967
 10968  054456  052566                    $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 10969  054460  053130                    $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 10970  054462  053220                    $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 10971  054464  053540                    $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 10972  054466  054310                    $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
 10973  054470  054346                    $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
 10974  054472  047434                    SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
 10975
```

```
10976
10977                                   .SBTTL  SERVICE MSGS
10978                                   .EVEN
10979  054474  000062                   BSE22S: .BLKW   50.              ;22 SECTOR SOFTWARE INFO
10980
10981
10982  054640  005015  045522  033060   MSG1:   .ASCII  <CR><LF>/RK06\07 DRV TST PRT 1/
10983  054646  030134  020067  051104
10984  054654  020126  051524  020124
10985  054662  051120  020124     061
10986  054667     015  041412  051132           .ASCII  <CR><LF>/CZR6HF0/<CR><LF>
10987  054674  044066  030106  005015
10988  054702  005015  025011  025052           .ASCII  <CR><LF>/       *** CAUTION ***/<CR><LF>
10989  054710  041440  052501  044524
10990  054716  047117  025040  025052
10991  054724  005015
10992  054726  005015  044124  051511           .ASCII  <CR><LF>/THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING CONTROL-C/
10993  054734  050040  047522  051107
10994  054742  046501  051440  047510
10995  054750  046125  020104  042502
10996  054756  044040  046101  042524
10997  054764  020104  047117  054514
10998  054772  041040  020131  054524
10999  055000  044520  043516  041440
11000  055006  047117  051124  046117
11001  055014  041455
11002  055016  005015  052117  042510           .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND,OR THE DRIVE/
11003  055024  053522  051511  026105
11004  055032  041440  051101  051124
11005  055040  042111  042507  043040
11006  055046  051117  040515  052124
11007  055054  047111  020107  047101
11008  055062  026104  051117  052040
11009  055070  042510  042040  044522
11010  055076  042526
11011  055100  005015  040515  020131           .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/<CR><LF>
11012  055106  042502  046040  043105
11013  055114  020124  047111  040440
11014  055122  020116  047125  042504
11015  055130  042524  046522  047111
11016  055136  042105  051440  040524
11017  055144  042524  005015
11018  055150  005015  047111  052111           .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
11019  055156  040511  046114  026131
11020  055164  042040  044522  042526
11021  055172  020123  047524  041040
11022  055200  020105  042524  052123
11023  055206  042105  051440  047510
11024  055214  046125  020104  040510
11025  055222  042526  006472     012
11026  055227     015  040412  020056           .ASCII  <CR><LF>/A.  HEADS MANUALLY LOADED/
11027  055234  044040  040505  051504
11028  055242  046440  047101  040525
11029  055250  046114  020131  047514
11030  055256  042101  042105
11031  055262  005015  027102  020040           .ASCII  <CR><LF>/B.  CORRECT PORT SELECTED/
```

```
11032  055270  047503  051122  041505
11033  055276  020124  047520  052122
11034  055304  051440  046105  041505
11035  055312  042524     104
11036  055315     015  041412  020056          .ASCII  <CR><LF>/C.  WRITE LOCK DISABLED/
11037  055322  053440  044522  042524
11038  055330  046040  041517  020113
11039  055336  044504  040523  046102
11040  055344  042105
11041  055346  005015  027104  020040          .ASCII  <CR><LF>/D.  DRIVE READY INDICATOR ON/<CR><LF>
11042  055354  051104  053111  020105
11043  055362  042522  042101  020131
11044  055370  047111  044504  040503
11045  055376  047524  020122  047117
11046  055404  005015
11047  055406  005015  051104  053111          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE BOTH/
11048  055414  051505  047040  052117
11049  055422  052040  020117  042502
11050  055430  052040  051505  042524
11051  055436  020104  052515  052123
11052  055444  044040  053101  020105
11053  055452  047502  044124
11054  055456  005015  047520  052122          .ASCIZ  <CR><LF>/PORTS DESELECTED/<CR><LF>
11055  055464  020123  042504  042523
11056  055472  042514  052103  042105
11057  055500  005015     000
11058  055503     015  052012  020117  MSG2:   .ASCII  <CR><LF>/TO TEST DRIVE 0, REMOVE XXDP MEDIA,/
11059  055510  042524  052123  042040
11060  055516  044522  042526  030040
11061  055524  020054  042522  047515
11062  055532  042526  054040  042130
11063  055540  020120  042515  044504
11064  055546  026101
11065  055550  005015  046103  040505          .ASCII  <CR><LF>/CLEAR LOC 40, & HIT CONT. ./
11066  055556  020122  047514  020103
11067  055564  030064  020054  020046
11068  055572  044510  020124  047503
11069  055600  052116  020056     056
11070  055605     015  044412  020106          .ASCII  <CR><LF>/IF DRIVE 0 ISN'T TO BE TESTED JUST HIT CONT. ./<CR><LF>
11071  055612  051104  053111  020105
11072  055620  020060  051511  023516
11073  055626  020124  047524  041040
11074  055634  020105  042524  052123
11075  055642  042105  045040  051525
11076  055650  020124  044510  020124
11077  055656  047503  052116  020056
11078  055664  006456     012
11079  055667     015  042012  044522  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
11080  055674  042526  051450  020051
11081  055702  047524  041040  020105
11082  055710  042524  052123  042105
11083  055716  020072     000
11084  055721     015  041012  051525  MSG4:   .ASCIZ  <CR><LF>/BUS ADDR (177440): /
11085  055726  040440  042104  020122
11086  055734  030450  033467  032064
11087  055742  024460  020072     000
```

```
11088  055747     015  041412  047117  MSG5:   .ASCIZ  <CR><LF>/CONTR ADDR (210): /
11089  055754  051124  040440  042104
11090  055762  020122  031050  030061
11091  055770  035051  000040
11092  055774  005015  047111  051124  MSG6:   .ASCIZ  <CR><LF>/INTR AT PC=/
11093  056002  040440  020124  041520
11094  056010  000075
11095  056012  005015  051104  053111  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
11096  056020  020105  020060  044527
11097  056026  046114  047040  052117
11098  056034  041040  020105  042524
11099  056042  052123  042105     000
11100  056047     015  052012  051505  MSG8:   .ASCIZ  <CR><LF>/TEST 16 TAKES 2 TO 4 MIN./<CR><LF>
11101  056054  020124  033061  052040
11102  056062  045501  051505  031040
11103  056070  052040  020117  020064
11104  056076  044515  027116  005015
11105  056104     000
11106  056105     015  041012  050131  MSG9:   .ASCIZ  <CR><LF>/BYPASSING TEST 16/<CR><LF>
11107  056112  051501  044523  043516
11108  056120  052040  051505  020124
11109  056126  033061  005015     000
11110  056133     015  005012  044527  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVE(S):/
11111  056140  046114  052040  051505
11112  056146  020124  051104  053111
11113  056154  024105  024523  000072
11114  056162  005015  050012  053517  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
11115  056170  051105  052440  020120
11116  056176  042522  052123  051101
11117  056204  020124  047524  052040
11118  056212  051505  020124  006461
11119  056220  000012
11120  056222  005015  040520  045503  MSG12:  .ASCIZ  <CR><LF>/PACK BEING FORMATTED/<CR><LF>
11121  056230  041040  044505  043516
11122  056236  043040  051117  040515
11123  056244  052124  042105  005015
11124  056252     000
11125  056253     015  047012  020117  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS/
11126  056260  020114  051117  050040
11127  056266  041440  047514  045503
11128  056274     123
11129  056275     015  040412  046114          .ASCIZ  <CR><LF>/ALL TIMING TESTS BYPASSED/
11130  056302  052040  046511  047111
11131  056310  020107  042524  052123
11132  056316  020123  054502  040520
11133  056324  051523  042105     000
11134  056331     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
11135  056336  051501  044523  043516
11136  056344  042040  044522  042526
11137  056352  000040
11138  056354  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
11139  056362  042526  000040
11140  056366  005015  051104  053111  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL #/
11141  056374  020105  042523  044522
11142  056402  046101  021440     000
11143  056407     015  041412  051101  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO./
```

```
11144  056411  051124  042111  042507
11145  056422  051440  051105  040511
11146  056430  020114  047516  000056
11147  056436  005015  040412  047502  MSG18:  .ASCIZ  <CR><LF><LF>/ABORTING BAL OF TESTS/<CR><LF><LF>
11148  056444  052122  047111  020107
11149  056452  040502  020114  043117
11150  056460  052040  051505  051524
11151  056466  005015  000012
11152  056472  005015  040412  046114  MSG19:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
11153  056500  042040  044522  042526
11154  056506  020123  042524  052123
11155  056514  042105  005015  000012
11156  056522  005015  047515  044504  MSG20:  .ASCII  <CR><LF>/MODIFIED VERSION OF FORMAT PACK TEST FOR MODULE TESTING/
11157  056530  044506  042105  053040
11158  056536  051105  044523  047117
11159  056544  047440  020106  047506
11160  056552  046522  052101  050040
11161  056560  041501  020113  042524
11162  056566  052123  043040  051117
11163  056574  046440  042117  046125
11164  056602  020105  042524  052123
11165  056610  047111     107
11166  056613     015  052012  020117          .ASCIZ  <CR><LF>/TO RESTORE HEADERS ON CYL 0 & 1, ALL TRACKS/
11167  056620  042522  052123  051117
11168  056626  020105  042510  042101
11169  056634  051105  020123  047117
11170  056642  041440  046131  030040
11171  056650  023040  030440  020054
11172  056656  046101  020114  051124
11173  056664  041501  051513     000
11174  056671     015  041012  050131  MSG21:  .ASCIZ  <CR><LF>/BYPASSING TESTS 36,40,41 FOR MODULE TESTING/<CR><LF>
11175  056676  051501  044523  043516
11176  056704  052040  051505  051524
11177  056712  031440  026066  030064
11178  056720  032054  020061  047506
11179  056726  020122  047515  052504
11180  056734  042514  052040  051505
11181  056742  044524  043516  005015
11182  056750     000
11183  056751     015  043012  051117  MSG22:  .ASCIZ  <CR><LF>/FORMATTING FINISHED/<CR><LF>
11184  056756  040515  052124  047111
11185  056764  020107  044506  044516
11186  056772  044123  042105  005015
11187  057000     000
11188  057001     015  052012  051505  MSG23:  .ASCIZ  <CR><LF>/TEST 16 TAKES 10 TO 12 MIN./<CR><LF>
11189  057006  020124  033061  052040
11190  057014  045501  051505  030440
11191  057022  020060  047524  030440
11192  057030  020062  044515  027116
11193  057036  005015     000
11194  057041     015  050012  046507  MSG74:  .ASCIZ  <CR><LF>/PGM ABORT PENDING.../
11195  057046  040440  047502  052122
11196  057054  050040  047105  044504
11197  057062  043516  027056  000056
11198  057070  005015  040510  052114  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING.../
11199  057076  050040  047105  044504
```

```
 11200  057104  043516  027056  000056
 11201  057112  005015  043520  020115  MSG76:  .ASCIZ  <CR><LF>/PGM ABORTED/
 11202  057120  041101  051117  042524
 11203  057126  000104
 11204  057130  005015  050103  020125  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
 11205  057136  040510  052114  042105
 11206  057144     000
 11207
 11208
 11209                                  .SBTTL  ERR MSGS
 11210
 11211  057145     015  042412  051122  EM1:    .ASCIZ  <CR><LF>/ERR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
 11212  057152  020054  047117  054514
 11213  057160  030040  052040  051110
 11214  057166  020125  020067  046101
 11215  057174  047514  042527  026104
 11216  057202  052040  054522  040440
 11217  057210  040507  047111  005015
 11218  057216     000
 11219  057217     123  046105  041505  EM2:    .ASCIZ  /SELECTED DRIVE # IN RKCS2 CANNOT BE READ BACK IN RKMR2/
 11220  057224  042524  020104  051104
 11221  057232  053111  020105  020043
 11222  057240  047111  051040  041513
 11223  057246  031123  041440  047101
 11224  057254  047516  020124  042502
 11225  057262  051040  040505  020104
 11226  057270  040502  045503  044440
 11227  057276  020116  045522  051115
 11228  057304  000062
 11229  057306  005015  041101  051117  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXP TIME OUT AT PC=/
 11230  057314  020124  042524  052123
 11231  057322  027123  027056  047125
 11232  057330  054105  020120  044524
 11233  057336  042515  047440  052125
 11234  057344  040440  020124  041520
 11235  057352  000075
 11236  057354  040506  040524  020114  EM4:    .ASCII  /FATAL ERROR/<CR><LF>
 11237  057362  051105  047522  006522
 11238  057370     012
 11239  057371     101  047502  052122          .ASCII  /ABORTING BALANCE OF TESTS/<CR><LF>
 11240  057376  047111  020107  040502
 11241  057404  040514  041516  020105
 11242  057412  043117  052040  051505
 11243  057420  051524  005015
 11244  057424  042510  042101  050040          .ASCIZ  /HEAD POSITION CANNOT BE DETERMINED/
 11245  057432  051517  052111  047511
 11246  057440  020116  040503  047116
 11247  057446  052117  041040  020105
 11248  057454  042504  042524  046522
 11249  057462  047111  042105     000
 11250  057467     115  051504  051440  EM5:    .ASCIZ  /MDS SET IN RKCS2/
 11251  057474  052105  044440  020116
 11252  057502  045522  051503  000062
 11253  057510  043125  020105  042523  EM6:    .ASCIZ  /UFE SET IN RKCS2/
 11254  057516  020124  047111  051040
 11255  057524  041513  031123     000
```

```
11256  057531     104  040522  044440  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
11257  057536  020116  045522  051504
11258  057544  023040  047040  042105
11259  057552  044440  020116  045522
11260  057560  051503  020062  042522
11261  057566  042523  035524  053440
11262  057574  047522  043516  050040
11263  057602  051117  020124  042523
11264  057610  042514  052103  042105
11265  057616  000077
11266  057620  051104  020126  054524  EM8:    .ASCIZ  /DRV TYPE NOT SET IN RKMR2/
11267  057626  042520  047040  052117
11268  057634  051440  052105  044440
11269  057642  020116  045522  051115
11270  057650  000062
11271  057652  051104  053111  020105  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT TYPED BY OPERATOR/
11272  057660  047516  020124  051120
11273  057666  051505  047105  020124
11274  057674  052502  020124  054524
11275  057702  042520  020104  054502
11276  057710  047440  042520  040522
11277  057716  047524  000122
11278  057722  041101  051117  020124  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REF CONTR REG/
11279  057730  042524  052123  027123
11280  057736  027056  040503  047116
11281  057744  052117  051040  043105
11282  057752  041440  047117  051124
11283  057760  051040  043505     000
11284  057765     104  040522  044440  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
11285  057772  020116  045522  051504
11286  060000  023040  047040  042105
11287  060006  044440  020116  045522
11288  060014  051503  020062  047502
11289  060022  044124  051440  052105
11290  060030     000
11291  060031     103  047117  051124  EM12:   .ASCIZ  /CONTR NOT READY IN RKCS1/
11292  060036  047040  052117  051040
11293  060044  040505  054504  044440
11294  060052  020116  045522  051503
11295  060060  000061
11296  060062  047516  040440  052124  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
11297  060070  020116  047111  051040
11298  060076  040513  047523  000106
11299  060104  051127  047117  020107  EM14:   .ASCIZ  /WRONG ATTN IN RKASOF/
11300  060112  052101  047124  044440
11301  060120  020116  045522  051501
11302  060126  043117     000
11303  060131     104  042122  020131  EM15:   .ASCIZ  /DRDY NOT CLEARED IN RKMR2/
11304  060136  047516  020124  046103
11305  060144  040505  042522  020104
11306  060152  047111  051040  046513
11307  060160  031122     000
11308  060163     104  041523  047040  EM16:   .ASCIZ  /DSC NOT SET IN RKMR2/
11309  060170  052117  051440  052105
11310  060176  044440  020116  045522
11311  060204  051115  000062
```

```
 11312  060210  051515  020107  030101  EM17:   .ASCIZ  /MSG A0 ERROR/
 11313  060216  042440  051122  051117
 11314  060224     000
 11315  060225     115  043523  041040  EM18:   .ASCIZ  /MSG B0 ERROR/
 11316  060232  020060  051105  047522
 11317  060240  000122
 11318  060242  051515  020107  030501  EM19:   .ASCIZ  /MSG A1 ERROR/
 11319  060250  042440  051122  051117
 11320  060256     000
 11321  060257     115  043523  041040  EM20:   .ASCIZ  /MSG B1 ERROR/
 11322  060264  020061  051105  047522
 11323  060272  000122
 11324  060274  042503  051122  051440  EM21:   .ASCIZ  /CERR SET IN PKCS1/
 11325  060302  052105  044440  020116
 11326  060310  045522  051503  000061
 11327  060316  046122  020123  047111  EM22:   .ASCIZ  /RLS IN RKCS2 SET CERR IN RKCS1/
 11328  060324  051040  041513  031123
 11329  060332  051440  052105  041440
 11330  060340  051105  020122  047111
 11331  060346  051040  041513  030523
 11332  060354     000
 11333  060355     125  042506  044440  EM23:   .ASCIZ  /UFE IN RKCS2 SET (SACK) AFTER RLS IN RKCS2 SENT/
 11334  060362  020116  045522  051503
 11335  060370  020062  042523  020124
 11336  060376  051450  041501  024513
 11337  060404  040440  052106  051105
 11338  060412  051040  051514  044440
 11339  060420  020116  045522  051503
 11340  060426  020062  042523  052116
 11341  060434     000
 11342  060435     126  020126  047516  EM24:   .ASCIZ  /VV NOT SET IN RKMR2/
 11343  060442  020124  042523  020124
 11344  060450  047111  051040  046513
 11345  060456  031122     000
 11346  060461     104  053122  052040  EM25:   .ASCIZ  /DRV TYPE SET IN RKMR2/
 11347  060466  050131  020105  042523
 11348  060474  020124  047111  051040
 11349  060502  046513  031122     000
 11350  060507     104  052104  051440  EM26:   .ASCIZ  /DDT SET IN RKDS/
 11351  060514  052105  044440  020116
 11352  060522  045522  051504     000
 11353  060527     104  054524  020105  EM27:   .ASCIZ  /DTYE SET IN RKER/
 11354  060534  042523  020124  047111
 11355  060542  051040  042513  000122
 11356  060550  052104  042531  047040  EM28:   .ASCIZ  /DTYE NOT SET IN RKER AFTER WRONG CDT IN RKCS1/
 11357  060556  052117  051440  052105
 11358  060564  044440  020116  045522
 11359  060572  051105  040440  052106
 11360  060600  051105  053440  047522
 11361  060606  043516  041440  052104
 11362  060614  044440  020116  045522
 11363  060622  051503  000061
 11364  060626  052104  042531  044440  EM29:   .ASCIZ  /DTYE IN RKER DID NOT SET CERR IN RKCS1/
 11365  060634  020116  045522  051105
 11366  060642  042040  042111  047040
 11367  060650  052117  051440  052105
```

```
 11368  060656  041440  051105  020122
 11369  060664  047111  051040  041513
 11370  060672  030523     000
 11371  060675     103  042055  050040  EM30:   .ASCIZ  /C-D PAR ERR SET IN RKMR3/
 11372  060702  051101  042440  051122
 11373  060710  051440  052105  044440
 11374  060716  020116  045522  051115
 11375  060724  000063
 11376  060726  026504  020103  040520  EM31:   .ASCIZ  /D-C PAR SET IN RKCS1/
 11377  060734  020122  042523  020124
 11378  060742  047111  051040  041513
 11379  060750  030523     000
 11380  060753     106  052114  047040  EM32:   .ASCIZ  /FLT NOT SET IN RKMR3/
 11381  060760  052117  051440  052105
 11382  060766  044440  020116  045522
 11383  060774  051115  000063
 11384  061000  026503  020104  040520  EM33:   .ASCIZ  /C-D PAR ERR NOT SET IN RKMR3/
 11385  061006  020122  051105  020122
 11386  061014  047516  020124  042523
 11387  061022  020124  047111  051040
 11388  061030  046513  031522     000
 11389  061035     104  041455  050040  EM34:   .ASCIZ  /D-C PAR NOT SET IN RKCS1/
 11390  061042  051101  047040  052117
 11391  061050  051440  052105  044440
 11392  061056  020116  045522  051503
 11393  061064  000061
 11394  061066  026504  020103  040520  EM35:   .ASCIZ  /D-C PAR IN RKCS1 DID NOT SET CERR IN RKCS1/
 11395  061074  020122  047111  051040
 11396  061102  041513  030523  042040
 11397  061110  042111  047040  052117
 11398  061116  051440  052105  041440
 11399  061124  051105  020122  047111
 11400  061132  051040  041513  030523
 11401  061140     000
 11402  061141     103  046131  040440  EM36:   .ASCIZ  /CYL ADDR IN B2 NOT SAME AS RKDC/
 11403  061146  042104  020122  047111
 11404  061154  041040  020062  047516
 11405  061162  020124  040523  042515
 11406  061170  040440  020123  045522
 11407  061176  041504     000
 11408  061201     103  046131  042040  EM37:   .ASCIZ  /CYL DIFF IN A2 NOT SAME AS RKDC/
 11409  061206  043111  020106  047111
 11410  061214  040440  020062  047516
 11411  061222  020124  040523  042515
 11412  061230  040440  020123  045522
 11413  061236  041504     000
 11414  061241     103  046131  042040  EM38:   .ASCIZ  /CYL DIFF IN RKMR2 NOT SAME AS 'CYL DIFF'/
 11415  061246  043111  020106  047111
 11416  061254  051040  046513  031122
 11417  061262  047040  052117  051440
 11418  061270  046501  020105  051501
 11419  061276  023440  054503  020114
 11420  061304  044504  043106  000047
 11421  061312  054503  020114  044504  EM39:   .ASCIZ  /CYL DIFF & OFST IN A2 NOT =0/
 11422  061320  043106  023040  047440
 11423  061326  051506  020'24  047111
```

```
11424  061334  040440  020062  047516
11425  061342  020124  030075     000
11426  061347     103  046131  040440  EM40:   .ASCIZ  /CYL ADDR IN B2 NOT =0/
11427  061354  042104  020122  047111
11428  061362  041040  020062  047516
11429  061370  020124  030075     000
11430  061375     103  046131  040440  EM41:   .ASCIZ  /CYL ADDR IN B2 DID NOT REMAIN =0/
11431  061402  042104  020122  047111
11432  061410  041040  020062  044504
11433  061416  020104  047516  020124
11434  061424  042522  040515  047111
11435  061432  036440  000060
11436  061436  044514  044515  020124  EM42:   .ASCIZ  /LIMIT DET NOT SET IN B1/
11437  061444  042504  020124  047516
11438  061452  020124  042523  020124
11439  061460  047111  041040  000061
11440  061466  042510  042101  040440  EM43:   .ASCIZ  /HEAD ADDR IN B3 NOT =0/
11441  061474  042104  020122  047111
11442  061502  041040  020063  047516
11443  061510  020124  030075     000
11444  061515     110  040505  020104  EM44:   .ASCIZ  /HEAD DECODE IN B3 INCORRECT/
11445  061522  042504  047503  042504
11446  061530  044440  020116  031502
11447  061536  044440  041516  051117
11448  061544  042522  052103     000
11449  061551     104  044522  042526  EM45:   .ASCII  /DRIVE READY IN RKMR2 NOT SET BY 1 SEC FROM FWD/
11450  061556  051040  040505  054504
11451  061564  044440  020116  045522
11452  061572  051115  020062  047516
11453  061600  020124  042523  020124
11454  061606  054502  030440  051440
11455  061614  041505  043040  047522
11456  061622  020115  053506     104
11457  061627     015  044412  020116          .ASCIZ  <CR><LF>/IN RTZ PORTION OF START SPIN CMD/
11458  061634  052122  020132  047520
11459  061642  052122  047511  020116
11460  061650  043117  051440  040524
11461  061656  052122  051440  044520
11462  061664  020116  046503  000104
11463  061672  051515  020107  031101  EM46:   .ASCIZ  /MSG A2 ERR/
11464  061700  042440  051122     000
11465  061705     115  043523  041040  EM47:   .ASCIZ  /MSG B2 ERR/
11466  061712  020062  051105  000122
11467  061720  051515  020107  031502  EM48:   .ASCIZ  /MSG B3 ERR/
11468  061726  042440  051122     000
11469  061733     106  042127  047040  EM49:   .ASCIZ  /FWD NOT SET IN RKMR2 IN RTZ PORTION OF START SPIN CMD/
11470  061740  052117  051440  052105
11471  061746  044440  020116  045522
11472  061754  051115  020062  047111
11473  061762  051040  055124  050040
11474  061770  051117  044524  047117
11475  061776  047440  020106  052123
11476  062004  051101  020124  050123
11477  062012  047111  041440  042115
11478  062020     000
11479  062021     106  042127  047040  EM50:   .ASCIZ  /FWD NOT SET IN RKMR2 FROM START SPIN CMD/
```

```
11480  062026  052117  051440  052105
11481  062034  044440  020116  045522
11482  062042  051115  020062  051106
11483  062050  046517  051440  040524
11484  062056  052122  051440  044520
11485  062064  020116  046503  000104
11486  062072  053506  020104  047516  EM51:   .ASCIZ  /FWD NOT CLEARED IN RKMR2 BY 5 SEC OF MOTION FROM START SPIN CMD/
11487  062100  020124  046103  040505
11488  062106  042522  020104  047111
11489  062114  051040  046513  031122
11490  062122  041040  020131  020065
11491  062130  042523  020103  043117
11492  062136  046440  052117  047511
11493  062144  020116  051106  046517
11494  062152  051440  040524  052122
11495  062160  051440  044520  020116
11496  062166  046503  000104
11497  062172  030062  051440  041505  EM52:   .ASCIZ  /20 SEC FORMAT NOT SET IN RKMR2/
11498  062200  043040  051117  040515
11499  062206  020124  047516  020124
11500  062214  042523  020124  047111
11501  062222  051040  046513  031122
11502  062230     000
11503  062231     123  041505  030040  EM53:   .ASCIZ  /SEC 0 NOT FOUND BY 50 MS/
11504  062236  047040  052117  043040
11505  062244  052517  042116  041040
11506  062252  020131  030065  046440
11507  062260  000123
11508  062262  044504  043106  051440  EM54:   .ASCIZ  /DIFF SEC NOT FOUND BY 3 MS/
11509  062270  041505  047040  052117
11510  062276  043040  052517  042116
11511  062304  041040  020131  020063
11512  062312  051515     000
11513  062315     101  052124  020116  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOr/
11514  062322  047516  020124  046103
11515  062330  040505  042522  020104
11516  062336  047111  051040  040513
11517  062344  047523  000106
11518  062350  047125  054105  020120  EM56:   .ASCIZ  /UNEXP MEM PAR TRAP/
11519  062356  042515  020115  040520
11520  062364  020122  051124  050101
11521  062372     000
11522  062373     127  042503  040040  EM57:   .ASCIZ  /WCE @ CYL 411, TRK 2, SEC 2/
11523  062400  041440  046131  032040
11524  062406  030461  020054  051124
11525  062414  020113  026062  051440
11526  062422  041505  031040     000
11527  062427     015  051412  042520  EM58:   .ASCIZ  <CR><LF>/SPEED OK IN RKMR2 NOT =0 BY TIMEOUT/
11528  062434  042105  047440  020113
11529  062442  047111  051040  046513
11530  062450  031122  047040  052117
11531  062456  036440  020060  054502
11532  062464  052040  046511  047505
11533  062472  052125     000
11534  062475     114  046511  042040  EM59:   .ASCIZ  /LIM DET NOT SET IN RKMR3/
11535  062502  052105  047040  052117
```

```
11536  062510  051440  052105  044440
11537  062516  020116  045522  051115
11538  062524  000063
11539  062526  042510  042101  020123  EM60:   .ASCIZ  /HEADS HOME NOT SET IN RKMR2/
11540  062534  047510  042515  047040
11541  062542  052117  051440  052105
11542  062550  044440  020116  045522
11543  062556  051115  000062
11544  062562  047514  042101  044040  EM61:   .ASCIZ  /LOAD HEADS NOT SET IN RKMR2/
11545  062570  040505  051504  047040
11546  062576  052117  051440  052105
11547  062604  044440  020116  045522
11548  062612  051115  000062
11549  062616  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
11550  062624  020124  047111  051040
11551  062632  041513  031123     000
11552  062637     115  043523  041040  EM64:   .ASCIZ  /MSG B3 HEAD REG NOT =0/
11553  062644  020063  042510  042101
11554  062652  051040  043505  047040
11555  062660  052117  036440  000060
11556  062666  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERR/
11557  062674  040505  042504  020122
11558  062702  051105  000122
11559  062706  054503  020114  042101  EM66:   .ASCIZ  /CYL ADDR IN RKMR3 INCORRECT/
11560  062714  051104  044440  020116
11561  062722  045522  051115  020063
11562  062730  047111  047503  051122
11563  062736  041505  000124
11564  062742  042522  042101  047111  EM67:   .ASCIZ  /READING CYL 0 HEADERS ON CYL 1/
11565  062750  020107  054503  020114
11566  062756  020060  042510  042101
11567  062764  051105  020123  047117
11568  062772  041440  046131  030440
11569  063000     000
11570  063001     122  040505  044504  EM68:   .ASCIZ  /READING CYL 1 HEADERS ON CYL 0/
11571  063006  043516  041440  046131
11572  063014  030440  044040  040505
11573  063022  042504  051522  047440
11574  063030  020116  054503  020114
11575  063036  000060
11576  063040  046101  043511  020116  EM69:   .ASCIZ  /ALIGN CART USED/
11577  063046  040503  052122  052440
11578  063054  042523  000104
11579  063060  047125  054105  020120  EM70:   .ASCIZ  /UNEXP ATTN/
11580  063066  052101  047124     000
11581  063073     104  041523  051440  EM71:   .ASCIZ  /DSC SET IN RKMR2/
11582  063100  052105  044440  020116
11583  063106  045522  051115  000062
11584  063114  047506  046522  052101  EM72:   .ASCIZ  /FORMAT TEST BYPASSED/
11585  063122  052040  051505  020124
11586  063130  054502  040520  051523
11587  063136  042105     000
11588  063141     103  047524  051440  EM73:   .ASCIZ  /CTO SET IN RKCS1/
11589  063146  052105  044440  020116
11590  063154  045522  051503  000061
11591  063162  052122  020132  047516  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
```

```
 11592  063170  020124  042523  020124
 11593  063176  047111  051040  046513
 11594  063204  031122     000
 11595  063207     111  040504  020105  EM75:   .ASCIZ  /IDAE NOT SET IN RKMR3/
 11596  063214  047516  020124  042523
 11597  063222  020124  047111  051040
 11598  063230  046513  031522     000
 11599  063235     120  050111  051440  EM76:   .ASCIZ  /PIP SET IN RKMR2/
 11600  063242  052105  044440  020116
 11601  063250  045522  051115  000062
 11602  063256  040506  046125  020124  EM77:   .ASCIZ  /FAULT NOT =0 IN RKMR3/
 11603  063264  047516  020124  030075
 11604  063272  044440  020116  045522
 11605  063300  051115  000063
 11606  063304  054503  020114  044504  EM78:   .ASCIZ  /CYL DIFF IN RKMR2 DID NOT REMAIN = 1 IN SEEK TO SELF/
 11607  063312  043106  044440  020116
 11608  063320  045522  051115  020062
 11609  063326  044504  020104  047516
 11610  063334  020124  042522  040515
 11611  063342  047111  036440  030440
 11612  063350  044440  020116  042523
 11613  063356  045505  052040  020117
 11614  063364  042523  043114     000
 11615  063371     116  042105  051440  EM79:   .ASCIZ  /NED SET IN RKCS2/
 11616  063376  052105  044440  020116
 11617  063404  045522  051503  000062
 11618  063412  047125  047514  042101  EM80:   .ASCIZ  /UNLOAD NOT SET IN RKMR2/
 11619  063420  047040  052117  051440
 11620  063426  052105  044440  020116
 11621  063434  045522  051115  000062
 11622  063442  050123  047111  047040  EM81:   .ASCIZ  /SPIN NOT SET IN RKMR2/
 11623  063450  052117  051440  052105
 11624  063456  044440  020116  045522
 11625  063464  051115  000062
 11626  063470  052122  020132  047516  EM82:   .ASCIZ  /RTZ NOT SET IN RKMR2/
 11627  063476  020124  042523  020124
 11628  063504  047111  051040  046513
 11629  063512  031122     000
 11630  063515     122  040505  020104  EM83:   .ASCIZ  /READ HEADER ERR   WORD 0 (CYL#)/
 11631  063522  042510  042101  051105
 11632  063530  042440  051122  020040
 11633  063536  053440  051117  020104
 11634  063544  020060  041450  046131
 11635  063552  024443     000
 11636  063555     106  051117  040515  EM84:   .ASCIZ  /FORMAT IN RKMR3 NOT SET/
 11637  063562  020124  047111  051040
 11638  063570  046513  031522  047040
 11639  063576  052117  051440  052105
 11640  063604     000
 11641  063605     111  046114  040440  EM85:   .ASCIZ  /ILL ADDR IN RKMR3 NOT =0/
 11642  063612  042104  020122  047111
 11643  063620  051040  046513  031522
 11644  063626  047040  052117  036440
 11645  063634  000060
 11646  063636  044127  046111  020105  EM86:   .ASCIZ  /WHILE WAITING FOR CONTR RDY OR AFTER CONTR RDY REC'D/
 11647  063644  040527  052111  047111
```

```
 11648  063652  020107  047506  020122
 11649  063660  047503  052116  020122
 11650  063666  042122  020131  051117
 11651  063674  040440  052106  051105
 11652  063702  041440  047117  051124
 11653  063710  051040  054504  051040
 11654  063716  041505  042047     000
 11655  063723     103  047101  047516  EM87:    .ASCIZ  /CANNOT READ BSE INFO/
 11656  063730  020124  042522  042101
 11657  063736  041040  042523  044440
 11658  063744  043116  000117
 11659  063750  047516  042040  044522  EM88:    .ASCII  /NO DRIVES FOUND ON BUS/<CR><LF>
 11660  063756  042526  020123  047506
 11661  063764  047125  020104  047117
 11662  063772  041040  051525  005015
 11663  064000  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY & PRESS 'CONT'/<CR><LF>
 11664  064006  047503  051122  041505
 11665  064014  046124  020131  020046
 11666  064022  051120  051505  020123
 11667  064030  041447  047117  023524
 11668  064036  005015     000
 11669  064041     116  020117  051104  EM89:    .ASCII  /NO DRIVES FOUND ,N DEVICE MAP ($DEVM)/<CR><LF>
 11670  064046  053111  051505  043040
 11671  064054  052517  042116  044440
 11672  064062  020116  042504  044526
 11673  064070  042503  046440  050101
 11674  064076  024040  042044  053105
 11675  064104  024515  005015
 11676  064110  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY & RESTART/<CR><LF>
 11677  064116  047503  051122  041505
 11678  064124  046124  020131  020046
 11679  064132  042522  052123  051101
 11680  064140  006524  000012
 11681  064144  042104  020124  047516  EM90:    .ASCIZ  /DDT NOT SET IN RKDS/
 11682  064152  020124  042523  020124
 11683  064160  047111  051040  042113
 11684  064166  000123
 11685
 11686                                  .SBTTL  DATA HEADERS
 11687
 11688  064170  042524  052123  047040  DH1:     .ASCIZ  /TEST NO.  PC/
 11689  064176  027117  020040  041520
 11690  064204     000
 11691  064205     122  046513  030522  DH2:     .ASCIZ  /RKMR1  RKMR2  RKMR3  RKER  RKDS  RKCS1  RKCS2/
 11692  064212  051011  046513  031122
 11693  064220  051011  046513  031522
 11694  064226  051011  042513  004522
 11695  064234  045522  051504  051011
 11696  064242  041513  030523  051011
 11697  064250  041513  031123     000
 11698  064255     122  053513  004503  DH3:     .ASCIZ  /RKWC  RKBA  RKDA  RKASOF  RKDC  RKECPS  RKECPT/
 11699  064262  045522  040502  051011
 11700  064270  042113  004501  045522
 11701  064276  051501  043117  051011
 11702  064304  042113  004503  045522
 11703  064312  041505  051520  051011
```

```
11704  064320  042513  050103  000124
11705  064326  051106  046517  041440  DH6:    .ASCIZ  /FROM CYL  TO CYL  CYL DIFF/
11706  064334  046131  020040  047524
11707  064342  041440  046131  020040
11708  064350  054503  020114  044504
11709  064356  043106     000
11710  064361     127  042510  020116  DH8:    .ASCIZ  /WHEN DRIVE UNLOADED/
11711  064366  051104  053111  020105
11712  064374  047125  047514  042101
11713  064402  042105     000
11714  064405     101  052106  051105  DH9:    .ASCIZ  /AFTER START SPIN CMD REC'D BY DRIVE/
11715  064412  051440  040524  052122
11716  064420  051440  044520  020116
11717  064426  046503  020104  042522
11718  064434  023503  020104  054502
11719  064442  042040  044522  042526
11720  064450     000
11721  064451     101  020124  047105  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
11722  064456  020104  043117  044040
11723  064464  040505  020104  047514
11724  064472  042101  047111  000107
11725  064500  043101  042524  020122  DH11:   .ASCIZ  /AFTER START SPIN CMD & FWD SET/
11726  064506  052123  051101  020124
11727  064514  050123  047111  041440
11728  064522  042115  023040  043040
11729  064530  042127  051440  052105
11730  064536     000
11731  064537     101  020124  047111  DH12:   .ASCIZ  /AT INNER LIM FROM START SPIN CMD/
11732  064544  042516  020122  044514
11733  064552  020115  051106  046517
11734  064560  051440  040524  052122
11735  064566  051440  044520  020116
11736  064574  046503  000104
11737  064600  051106  046517  047440  DH13:   .ASCIZ  /FROM OUTER LIM TO CYL 0 DURING LOADING/
11738  064606  052125  051105  046040
11739  064614  046511  052040  020117
11740  064622  054503  020114  020060
11741  064630  052504  044522  043516
11742  064636  046040  040517  044504
11743  064644  043516     000
11744  064647     101  052106  051105  DH14:   .ASCIZ  /AFTER SEEK WITH BAD PAR/
11745  064654  051440  042505  020113
11746  064662  044527  044124  041040
11747  064670  042101  050040  051101
11748  064676     000
11749  064677     101  052106  051105  DH16:   .ASCIZ  /AFTER LOADING HEAD REG & SEEK CMD/
11750  064704  046040  040517  044504
11751  064712  043516  044040  040505
11752  064720  020104  042522  020107
11753  064726  020046  042523  045505
11754  064734  041440  042115     000
11755  064741     101  052106  051105  DH17:   .ASCIZ  /AFTER RECAL CMD/
11756  064746  051040  041505  046101
11757  064754  041440  042115     000
11758  064761     101  052106  051105  DH18:   .ASCIZ  /AFTER UNLOAD CMD/
11759  064766  052440  046116  040517
```

```
11760  064774  020104  046503  000104
11761  065002  043101  042524  020122  DH19:   .ASCIZ  /AFTER PACK CMD/
11762  065010  040520  045503  041440
11763  065016  042115     000
11764  065021     101  052106  051105  DH20:   .ASCIZ  /AFTER SELECT DRIVE CMD/
11765  065026  051440  046105  041505
11766  065034  020124  051104  053111
11767  065042  020105  046503  000104
11768  065050  043101  042524  020122  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
11769  065056  052523  051502  051531
11770  065064  042524  020115  046103
11771  065072  040505  000122
11772  065076  043101  042524  020122  DH22:   .ASCIZ  /AFTER DRIVE CLEAR CMD/
11773  065104  051104  053111  020105
11774  065112  046103  040505  020122
11775  065120  046503  000104
11776  065124  042524  052123  047040  DH23:   .ASCIZ  /TEST NO.       TRAP PC/
11777  065132  027117  052011  040522
11778  065140  020120  041520     000
11779  065145     101  052106  051105  DH25:   .ASCIZ  /AFTER SEEK CMD/
11780  065152  051440  042505  020113
11781  065160  046503  000104
11782  065164  043101  042524  020122  DH26:   .ASCIZ  /AFTER READ DATA CMD/
11783  065172  042522  042101  042040
11784  065200  052101  020101  046503
11785  065206  000104
11786  065210  042411  050130  041505  DH28:   .ASCIZ  /       EXPECT/
11787  065216  000124
11788  065220  040411  052103  040525  DH29:   .ASCIZ  /       ACTUAL/
11789  065226  000114
11790  065230  043101  042524  020122  DH30:   .ASCIZ  /AFTER READ HEADER CMD/
11791  065236  042522  042101  044040
11792  065244  040505  042504  020122
11793  065252  046503  000104
11794  065256  030101  041011  004460  DH32:   .ASCIZ  /A0     B0      A1      B1      A2      B2      B3/
11795  065264  030501  041011  004461
11796  065272  031101  041011  004462
11797  065300  031502     000
11798  065303     104  051125  047111  DH33:   .ASCIZ  /DURING SEEK CMD/
11799  065310  020107  042523  045505
11800  065316  041440  042115     000
11801  065323     123  041505  047524  DH34:   .ASCIZ  /SECTOR REG UNSTABLE/
11802  065330  020122  042522  020107
11803  065336  047125  052123  041101
11804  065344  042514     000
11805  065347     102  052105  042527  D 35:   .ASCIZ  /BETWEEN SECTOR COUNTS/
11806  065354  047105  051440  041505
11807  065362  047524  020122  047503
11808  065370  047125  051524     000
11809  065375     106  047522  020115  DH36:   .ASCIZ  /FROM SECT  TO SECT/
11810  065402  042523  052103  000040
11811  065410  047524  051440  041505
11812  065416  000124
11813  065420  050123  042505  020104  DH37:   .ASCIZ  /SPEED OK NOT =0 10 SEC AFTER UNLOAD/
11814  065426  045517  047040  052117
11815  065434  036440  020060  030061
```

```
 11816  065442  051440  041505  040440
 11817  065450  052106  051105  052440
 11818  065456  046116  040517  000104
 11819  065464  043101  042524  020122  DH38:   .ASCIZ  /AFTER LIM DET/
 11820  065472  044514  020115  042504
 11821  065500  000124
 11822  065502  043101  042524  020122  DH39:   .ASCIZ  /AFTER WRITE HEADER CMD/
 11823  065510  051127  052111  020105
 11824  065516  042510  042101  051105
 11825  065524  041440  042115     000
 11826  065531     127  051117  021504  DH40:   .ASCIZ  /WORD#  HEADER WAS  SHOULD BE/
 11827  065536  044011  040505  042504
 11828  065544  020122  040527  020123
 11829  065552  051440  047510  046125
 11830  065560  020104  042502     000
 11831  065565     104  051125  047111  DH41:   .ASCIZ  /DURING RECAL CMD/
 11832  065572  020107  042522  040503
 11833  065600  020114  046503  000104
 11834  065606  047117  051440  041505  DH42:   .ASCIZ  /ON SEC 0,2,4,6,8  CYL 410 TRK 2/
 11835  065614  030040  031054  032054
 11836  065622  033054  034054  020040
 11837  065630  054503  020114  030464
 11838  065636  020060  051124  020113
 11839  065644  000062
 11840  065646  047506  046522  052101  DH44:   .ASCIZ  /FORMAT & ALL R-W TESTS WILL BE BYPASSED/
 11841  065654  023040  040440  046114
 11842  065662  051040  053455  052040
 11843  065670  051505  051524  053440
 11844  065676  046111  020114  042502
 11845  065704  041040  050131  051501
 11846  065712  042523  000104
 11847  065716  042502  040503  051525  DH45:   .ASCIZ  /BECAUSE OF LIMIT DETECT ERROR ON PREVIOUS TEST/
 11848  065724  020105  043117  046040
 11849  065732  046511  052111  042040
 11850  065740  052105  041505  020124
 11851  065746  051105  047522  020122
 11852  065754  047117  050040  042522
 11853  065762  044526  052517  020123
 11854  065770  042524  052123     000
 11855  065775     103  052517  042114  DH46:   .ASCIZ  /COULD NOT READ BSE INFO ON PREV TEST/
 11856  066002  047040  052117  051040
 11857  066010  040505  020104  051502
 11858  066016  020105  047111  047506
 11859  066024  047440  020116  051120
 11860  066032  053105  052040  051505
 11861  066040  000124
 11862  066042  043101  042524  020122  DH48:   .ASCIZ  /AFTER SEEK CMD TO INV CYL/
 11863  066050  042523  045505  041440
 11864  066056  042115  052040  020117
 11865  066064  047111  020126  054503
 11866  066072  000114
 11867  066074  051515  020107  023101  DH49:   .ASCIZ  /MSG A&B IN RKMR2 & RKMR3 RESP., ARE INVALID/
 11868  066102  020102  047111  051040
 11869  066110  046513  031122  023040
 11870  066116  051040  046513  031522
 11871  066124  051040  051505  027120
```

```
11872  066132  020054  051101  020105
11873  066140  047111  040526  044514
11874  066146  000104
11875  066150  043101  042524  020122  DH51:   .ASCIZ  /AFTER SEEK TO SELF CMD/
11876  066156  042523  045505  052040
11877  066164  020117  042523  043114
11878  066172  041440  042115     000
11879  066177     105  050130  041440  DH52:   .ASCIZ  /EXP CYL#       CYL HEADER WAS/
11880  066204  046131  004443  054503
11881  066212  020114  042510  042101
11882  066220  051105  053440  051501
11883  066226     000
11884  066227     117  020116  042523  DH53:   .ASCIZ  /ON SEC 10,12,14,16,18,20 CYL 410 TRK 2/
11885  066234  020103  030061  030454
11886  066242  026062  032061  030454
11887  066250  026066  034061  031054
11888  066256  020060  054503  020114
11889  066264  030464  020060  051124
11890  066272  020113  000062
11891                          .SBTTL  ERR OUTPUT DATA
11892
11893                          .EVEN
11894  066276  001214  001116          DT1:    $TESTN,$ERRPC
11895  066302  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11896  066310  003336  003334  003322
11897  066316  003324
11898  066320  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11899  066326  003340  003342  003354
11900  066334  003356
11901  066336  001214  001116  001350  DT4:    $TESTN,$ERRPC,FRCYL,TOCYL,CALDIF
11902  066344  001352  001360
11903  066350  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11904  066356  003336  003334  003322
11905  066364  003324
11906  066366  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11907  066374  003340  003342  003354
11908  066402  003356
11909  066404  001214  001116  001402  DT6:    $TESTN,$ERRPC,PSEC,ESEC
11910  066412  001404
11911  066414  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11912  066422  003336  003334  003322
11913  066430  003324
11914  066432  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11915  066440  003340  003342  003354
11916  066446  003356
11917  066450  001214  001116  001442  DT7:    $TESTN,$ERRPC,WDCNT,HDWD,TEMP1
11918  066456  001454  003360
11919  066462  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11920  066470  003336  003334  003322
11921  066476  003324
11922  066500  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11923  066506  003340  003342  003354
11924  066514  003356
11925  066516  001214  001116  001352  DT8:    $TESTN,$ERRPC,TOCYL,FRCYL,CALDIF
11926  066524  001350  001360
11927  066530  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
```

```
11928  066536  003336  003334  003322
11929  066544  003324
11930  066546  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11931  066554  003340  003342  003354
11932  066562  003356
11933  066564  001214  001116  001352  DT9:    $TESTN,$ERRPC,TOCYL,RHTAB
11934  066572  001674
11935  066574  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11936  066602  003336  003334  003322
11937  066610  003324
11938  066612  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11939  066620  003340  003342  003354
11940  066626  003356
11941  066630  001214  001116  001350  DT10:   $TESTN,$ERRPC,FRCYL,RHTAB
11942  066636  001674
11943  066640  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11944  066646  003336  003334  003322
11945  066654  003324
11946  066656  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11947  066664  003340  003342  003354
11948  066672  003356
11949  066674  001214  001334          DT11:   $TESTN,TRAPPC
11950  066700  001214  001116  003412  DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
11951  066706  003414  003416  003420
11952  066714  003372  003374  003376
11953  066722  003400
11954  066724  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11955  066732  003336  003334  003322
11956  066740  003324
11957  066742  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11958  066750  003340  003342  003354
11959  066756  003356
11960
11961  066760  001214  001116  003412  DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
11962  066766  003414  003416  003420
11963  066774  003422  003424
11964  067000  003372  003374  003376          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
11965  067006  003400  003402  003404
11966  067014  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11967  067022  003336  003334  003322
11968  067030  003324
11969  067032  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11970  067040  003340  003342  003354
11971  067046  003356
11972
11973  067050  001214  001116  003412  DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
11974  067056  003414  003416  003420
11975  067064  003422  003424  003430
11976  067072  003372  003374  003376          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
11977  067100  003400  003402  003404
11978  067106  003410
11979  067110  003346  003350  003352          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11980  067116  003336  003334  003322
11981  067124  003324
11982  067126  003326  003330  003332          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11983  067134  003340  003342  003354
```

```
11984  067142  003356
11985
11986                                   .SBTTL  ERR DATA FORMATS
11987
11988  067144  000003                   DF1:    3
11989  067146     002      000                  .BYTE 2,0
11990  067150  064205                           DH2
11991  067152     007      000                  .BYTE 7,0
11992  067154  064255                           DH3
11993  067156     007      000                  .BYTE   7,0
11994
11995  067160  000005                   DF2:    5
11996  067162     000      000                  .BYTE 0,0
11997  067164  066074                           DH49
11998  067166     000      000                  .BYTE 0,0
11999  067170  064170                           DH1
12000  067172     002      000                  .BYTE 2,0
12001  067174  064205                           DH2
12002  067176     007      000                  .BYTE 7,0
12003  067200  064255                           DH3
12004  067202     007      000                  .BYTE   7,0
12005
12006  067204  000001                   DF3:    1
12007  067206     002      000                  .BYTE   2,0
12008  067210  000003                   DF4:    3
12009  067212     002      000                  .BYTE   2,0
12010  067214  064205                           DH2
12011  067216     007      000                  .BYTE   7,0
12012  067220  064255                           DH3
12013  067222     007      000                  .BYTE   7,0
12014
12015  067224  000004                   DF5:    4
12016  067226     000      000                  .BYTE 0,0
12017  067230  064170                           DH1
12018  067232     002      000                  .BYTE 2,0
12019  067234  064205                           DH2
12020  067236     007      000                  .BYTE   7,0
12021  067240  064255                           DH3
12022  067242     007      000                  .BYTE   7,0
12023
12024  067244  000005                   DF6:    5
12025  067246     000      000                  .BYTE 0,0
12026  067250  064170                           DH1
12027  067252     002      000                  .BYTE 2,0
12028  067254  064326                           DH6
12029  067256     003      000                  .BYTE 3,0
12030  067260  064205                           DH2
12031  067262     007      000                  .BYTE   7,0
12032  067264  064255                           DH3
12033  067266     007      000                  .BYTE   7,0
12034
12035
12036
12037  067270  000004                   DF10:   4
12038  067272     000      000                  .BYTE 0,0
12039  067274  064170                           DH1
```

```
12040  067276  002  000          .BYTE 2,0
12041  067300  064205            DH2
12042  067302  007  000          .BYTE 7,0
12043  067304  064255            DH3
12044  067306  007  000          .BYTE  7,0
12045
12046  067310  000005       DF12:  5
12047  067312  000  000          .BYTE 0,0
12048  067314  064170            DH1
12049  067316  002  000          .BYTE 2,0
12050  067320  065375            DH36
12051  067322  002  000          .BYTE 2,0
12052  067324  064205            DH2
12053  067326  007  000          .BYTE  7,0
12054  067330  064255            DH3
12055  067332  007  000          .BYTE  7,0
12056
12057  067334  000004       DF14:  4
12058  067336  002  000          .BYTE  2,0
12059  067340  065531            DH40
12060  067342  003  000          .BYTE  3,0
12061  067344  064205            DH2
12062  067346  007  000          .BYTE  7,0
12063  067350  064255            DH3
12064  067352  007  000          .BYTE  7,0
12065
12066
12067  067354  000004       DF15:  4
12068  067356  000  000          .BYTE  0,0
12069  067360  064170            DH1
12070  067362  002  000          .BYTE  2,0
12071  067364  064205            DH2
12072  067366  007  000          .BYTE  7,0
12073  067370  064255            DH3
12074  067372  007  000          .BYTE  7,0
12075
12076  067374  000004       DF16:  4
12077  067376  000  000          .BYTE 0,0
12078  067400  064170            DH1
12079  067402  002  000          .BYTE 2,0
12080  067404  064205            DH2
12081  067406  007  000          .BYTE  7,0
12082  067410  064255            DH3
12083  067412  007  000          .BYTE  7,0
12084
12085  067414  000005       DF17:  5
12086  067416  000  000          .BYTE  0,0
12087  067420  065646            DH44
12088  067422  000  000          .BYTE  0,0
12089  067424  064170            DH1
12090  067426  002  000          .BYTE  2,0
12091  067430  064205            DH2
12092  067432  007  000          .BYTE  7,0
12093  067434  064255            DH3
12094  067436  007  000          .BYTE  7,0
12095  067440  000007       DF20:  7
```

```
12096  067442     000     000                  .BYTE   0,0
12097  067444  064170                          DH1
12098  067446     002     000                  .BYTE   2,0
12099  067450  065210                          DH28
12100  067452     000     000                  .BYTE   0,0
12101  067454  065256                          DH32
12102  067456     004     000                  .BYTE   4,0
12103  067460  065220                          DH29
12104  067462     004     000                  .BYTE   4,0
12105  067464  064205                          DH2
12106  067466     007     000                  .BYTE   7,0
12107  067470  064255                          DH3
12108  067472     007     000                  .BYTE   7,0
12109  067474  000004                  DF21:   4
12110  067476     002     000                  .BYTE   2,0
12111  067500  066177                          DH52
12112  067502     002     000                  .BYTE   2,0
12113  067504  064205                          DH2
12114  067506     007     000                  .BYTE   7,0
12115  067510  064255                          DH3
12116  067512     007     000                  .BYTE   7,0
12117  067514  000007                  DF22:   7
12118  067516     000     000                  .BYTE   0,0
12119  067520  064170                          DH1
12120  067522     002     000                  .BYTE   2,0
12121  067524  065210                          DH28
12122  067526     000     000                  .BYTE   0,0
12123  067530  065256                          DH32
12124  067532     006     000                  .BYTE   6,0
12125  067534  065220                          DH29
12126  067536     006     000                  .BYTE   6,0
12127  067540  064205                          DH2
12128  067542     007     000                  .BYTE   7,0
12129  067544  064255                          DH3
12130  067546     007     000                  .BYTE   7,0
12131
12132  067550  000007                  DF23:   7
12133  067552     000     000                  .BYTE   0,0
12134  067554  064170                          DH1
12135  067556     002     000                  .BYTE   2,0
12136  067560  065210                          DH28
12137  067562     000     000                  .BYTE   0,0
12138  067564  065256                          DH32
12139  067566     007     000                  .BYTE   7,0
12140  067570  065220                          DH29
12141  067572     007     000                  .BYTE   7,0
12142  067574  064205                          DH2
12143  067576     007     000                  .BYTE   7,0
12144  067600  064255                          DH3
12145  067602     007     000                  .BYTE   7,0
```

```
12146                                   ;;*****************************************************************************
12147                                   .SBTTL  TYPE ERR ROUTINE
12148                                   ;*ENTRY JSR PC,TYP ERR
12149                                   ;*RETURN RTS PC
12150                                   ;*
12151                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
12152                                   ;*ERR IS TO BE REPORTED. IT THEN USES THE "ERR TABLE" ($ERRTB)
12153                                   ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
12154                                   ;*THE ERR.
12155                                   ;;*****************************************************************************
12156  067604  104413                  TYPERR: SAVREG
12157  067606  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERR NUMBER
12158  067612  042700  177400                  BIC     #177400,R0      ;CLEAR SIGN EXTENSION
12159  067616  005300                          DEC     R0              ;FORM INDEX FOR ERR TABLE
12160  067620  006300                          ASL     R0
12161  067622  006300                          ASL     R0
12162  067624  006300                          ASL     R0
12163  067626  062700  003474          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERR ENTRY
12164  067632  012037  067646                  MOV     (R0)+,2$        ;GET EM POINTER
12165  067636  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
12166  067640  104401  001205                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
12167  067644  104401                          TYPE                    ;TYPE ERR MSG (EM)
12168  067646  000000          2$:     .WORD   0                       ;EM POINTER GOES HERE
12169  067650  012037  067664          3$:     MOV     (R0)+,4$        ;GET DH POINTER
12170  067654  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
12171  067656  104401  001205                  TYPE    ,$CRLF          ;TYPE CR-LF
12172  067662  104401                          TYPE                    ;TYPE DATA HEADER
12173  067664  000000          4$:     .WORD   0                       ;DH POINTER GOES HERE
12174  067666  012001          5$:     MOV     (R0)+,R1        ;GET DT POINTER
12175  067670  001455                  BEQ     20$             ;BRANCH IF THERE ARE NONE
12176  067672  005004                  CLR     R4              ;SET INDENT SWITCH
12177  067674  012000                  MOV     (R0)+,R0        ;GET DF POINTER
12178  067676  012002                  MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
12179  067700  001446                  BEQ     17$             ;DH NUM IS 0-BRANCH
12180  067702  005104                  COM     R4              ;NO INDENT
12181  067704  104401  001205          TYPE    ,$CRLF
12182  067710  112003          10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
12183  067712  105720                  TSTB    (R0)+           ;BUMP PAST FORMAT WORD
12184  067714  005703                  TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
12185  067716  001407                  BEQ     14$             ;NO - SKIP DATA PRINT
12186  067720  013146          11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
12187  067722  104402                  TYPOC                   ;TYPE IT
12188  067724  005303                  DEC     R3              ;MORE DATA WORDS
12189  067726  001403                  BEQ     14$             ;NO-BRANCH
12190  067730  104401  070060          TYPE    ,SPACE2         ;TYPE SEPARATORS
12191  067734  000771                  BR      11$             ;LOOP
12192  067736  005302          14$:    DEC     R2              ;MORE DH'S?
12193  067740  003431                  BLE     20$             ;NO-BRANCH
12194  067742  104401  001205          TYPE    ,$CRLF
12195  067746  005760  000002          TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
12196  067752  001404                  BEQ     15$             ;YES-BRANCH BYPASS INDENT
12197  067754  005104                  COM     R4              ;INDENT?
12198  067756  001002                  BNE     15$             ;NO-BRANCH
12199  067760  104401  070060          TYPE    ,SPACE2         ;YES-TYPE SPACES
12200  067764  012037  067772  15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
12201  067770  104401                  TYPE                    ;TYPE DH
```

```
12202  067772  000000            16$:    .WORD   0               ;DH POINTER GOES HERE
12203  067774  105710            TSTB    (R0)            ;TYPE A DT?
12204  067776  001003            BNE     21$             ;YES-BRANCH
12205  070000  062700  000002    ADD     #2,R0           ;INCREMENT DF POINTER
12206  070004  000754            BR      14$             ;SEE IF END OF DF BLOCK
12207  070006  104401  001205    21$:    TYPE    ,$CRLF
12208  070012  005704            TST     R4              ;INDENT?
12209  070014  001335            BNE     10$             ;NO-BRANCH
12210  070016  104401  070060    17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES
12211  070022  000732            BR      10$             ;LOOP
12212  070024  104414            20$:    RESREG
12213
12214  070026  032777  010000  111104    BIT     #SW12,@SWR      ;ABORT DRV AFTER 20 ERRS?
12215  070034  001410            BEQ     25$             ;BR IF NO
12216  070036  023727  001103  000024    CMP     $ERFLG,#20.     ;ELSE SEE IF 20 ERRS
12217  070044  001004            BNE     25$             ;BR IF NO
12218  070046  012706  001100          MOV     #STACK,SP       ;ELSE RESTORE STK
12219  070052  000137  042706          JMP     $EOP            ;AND DROP DRIVE
12220  070056  000207            25$:    RTS     PC
12221  070060  020040     000    SPACE2: .ASCIZ/  '              ;2 SPACES
```

```
12222                                   ;  ODT-11 -- V005A
12223
12224                                   ; DEC-11-UODPA-A-LA
12225
12226                                   ; COPYRIGHT 1969,1970,1972
12227                                   ; DIGITAL EQUIPMENT CORPORATION
12228                                   ; MAYNARD, MASSACHUSETTS 01754
12229                                           .ENABL  ABS,AMA
12230   070064                                  .EVEN
12231   070144                                  .=.+60
12232   000000                          R0      =       %0      ; REGISTER
12233   000001                          R1      =       %1      ;  NAMING
12234   000002                          R2      =       %2      ;   CONVENTIONS
12235   000003                          R3      =       %3
12236   000004                          R4      =       %4
12237   000005                          R5      =       %5
12238   000006                          SP      =       %6
12239   000007                          PC      =       %7
12240   177776                          ST      =       177776          ;STATUS REGISTER
12241                                   ;
12242   000014                          0.TVEC  =       14              ;TRT VECTOR LOCATION
12243   000340                          0.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
12244   000020                          0.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
12245   000003                          TRT     =       000003          ;TRT INSTRUCTION
12246   000006                          RTT     =       000006          ;RTT INSTRUCTION
12247                                   ;
12248                                   ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
12249                                   ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
12250                                   ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
12251                                   ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
12252                                   ;
12253   177562                          0.RDB   =       177562  ;R DATA BUFFER
12254   177560                          0.RCSR  =       177560  ;R C/SR
12255   177566                          0.TDB   =       177566  ;T DATA BUFFER
12256   177564                          0.TCSR  =       177564  ;T C/SR
12257                                   ;
12258                                   ;
12259                                   ; INITIALIZE ODT
12260                                   ;  USE 0.ODT FOR A NORMAL ENTRY
12261                                   ;  USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
12262                                   ;  USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
12263                                   ;
12264 070144  000413                    0.ODT:  BR      0.STRT          ;NORMAL ENTRY
12265 070146  000417                            BR      0.RST           ;RESTART
12266 070150  013737  177776  070124    0.ENTR: MOV     ST,0.UST        ;RE-ENTER  --  SAVE STATUS
12267 070156  013737  000016  177776            MOV     0.TVEC+2,ST     ;SET UP LOCAL STATUS
12268 070164  010737  070122                    MOV     PC,0.UPC        ;FAKE THE PC
12269 070170  000137  071322                    JMP     0.BK1
12270                                   ;
12271 070174  012706  070104            0.STRT: MOV     #0.UR0,SP       ;SET UP STACK
12272 070200  010637  070120                    MOV     SP,0.USP        ;FAKE THE SAVED STACK
12273 070204  000414                            BR      0.RST1          ;CLEAR BREAKPOINT TABLES
12274 070206  004037  071530            0.RST:  JSR     0,0.SVR         ;SAVE REGISTERS
12275 070212  013777  070142  177716            MOV     0.UIN,@0.ADR1   ;REMOVE THE BREAKPOINT
12276 070220  113704  070126                    MOVB    0.PRI,R4        ;GET ODT PRIORITY
12277 070224  106004                            RORB    R4              ;SHIFT
```

```
12278  070226  106004                          RORB    R4              ; INTO
12279  070230  106004                          RORB    R4              ;  POSITION
12280  070232  110437  177776                 MOVB    R4,ST           ;STORE IN STATUS
12281  070236  000127                  O.RST1: JMP     (PC)+
12282  070240  000403                          BR      O.45
12283  070242  012737  000002  071232          MOV     #RTI,O.RTIT     ;SET TO RTI IF 11/20 OR /05
12284  070250  105037  072151          O.45:   CLRB    O.P             ;DISALLOW PROCEED
12285  070254  012737  000340  000016          MOV     #O.STM,O.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
12286  070262  012737  071312  000014          MOV     #O.BRK,O.TVEC   ;PC TO TRT VECTOR
12287  070270  000447                          BR      O.RALL          ;CLEAR BREAKPOINT TABLES
12288                                  ;
12289                                  ; SPECIAL NAME HANDLER
12290                                  ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URO
12291                                  ;
12292  070272  004537  071752          O.REGT: JSR     5,O.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
12293  070276  012704  072175                  MOV     #O.TL,R4        ;TABLE START ADDRESS
12294  070302  120024                  O.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
12295  070304  001413                          BEQ     O.SP            ;JUMP IF YES
12296  070306  022704  072203                  CMP     #O.TL+O.LG,R4   ;IS THE SEARCH DONE?
12297  070312  101373                          BHI     O.RSP           ;BRANCH IF NOT
12298  070314  042700  177770                  BIC     #177770,R0      ;MASK OFF OCTAL
12299  070320  010004                          MOV     R0,R4
12300  070322  006304                  O.SP1:  ASL     R4
12301  070324  062704  070104                  ADD     #O.URO,R4       ;GENERATE ADDRESS
12302  070330  005202                          INC     R2              ;SET FOUND FLAG
12303  070332  000444                          BR      O.SCAN          ;GO FIND NEXT CHARACTER
12304  070334  162704  072166          O.SP:   SUB     #O.TL-7,R4      ;CORRECT CONSTANT
12305  070340  000770                          BR      O.SP1
12306                                  ;
12307                                  ; _ HANDLER - OPEN INDEXED ON THE PC
12308                                  ;
12309  070342  004737  072076          O.ORPC: JSR     PC,O.TCLS
12310  070346  010502                          MOV     R5,R2           ;CURRENT ADDRESS IN R2
12311  070350  061202                          ADD     @R2,R2          ;COMPUTE
12312  070352  006202                          ASR     R2              ;MOVE ONE BIT TO CARRY
12313  070354  103421                          BCS     O.ERR           ;ERR IF ODD NUMBER
12314  070356  006302                          ASL     R2              ;RESTORE WORD
12315  070360  005722                          TST     (R2)+           ; AND INCREMENT BY TWO
12316  070362  010205                          MOV     R2,R5           ;UPDATE CAD
12317  070364  000137  070636                  JMP     O.OP2           ;GO FINISH UP
12318                                  ;
12319                                  ; B HANDLER - SET AND REMOVE BREAKPOINTS
12320                                  ;
12321  070370  005702                  O.BKPT: TST     R2              ;IF NO NUMBER TYPED
12322  070372  001406                          BEQ     O.RALL          ;  REMOVE BREAKPOINT
12323  070374  006204                          ASR     R4              ;CHECK IF ODD
12324  070376  103410                          BCS     O.ERR           ;JUMP IF ODD
12325  070400  006304                          ASL     R4              ;RESTORE ONE BIT
12326  070402  010437  070136                  MOV     R4,O.ADR1       ;SET A BREAKPOINT
12327  070406  000412                          BR      O.DCD
12328  070410  012737  072212  070136  O.RALL: MOV     #O.TRTC,O.ADR1  ;CLEAR BREAKPOINT
12329  070416  000406                          BR      O.DCD
12330                                  ;
12331                                  ; CMD DECODER - ODT11
12332                                  ;
12333                                  ;   REGISTERS R0-R4 MAY BE USED,
```

```
12334                                   ;    REGISTER R5 WILL BE CONSIDERED SAFE
12335                                   ;
12336  070420  052705  000001          O.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
12337  070424  012700  000077                  MOV     #'?,R0          ;  ? TO BE TYPED
12338  070430  004537  072030                  JSR     5,O.FTYP        ; OUTPUT ?
12339  070434  004537  072130          O.DCD:  JSR     5,O.CRLS        ;TYPE  <CR><LF>*
12340  070440  005004                  O.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
12341  070442  005002                          CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
12342  070444  004537  071752          O.SCAN: JSR     5,O.GET         ;GET A CHAR, RETURN IN R0
12343  070450  022700  000060                  CMP     #'0,R0          ;COMPARE WITH ASCII 0
12344  070454  101013                          BHI     O.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
12345  070456  022700  000067                  CMP     #'7,R0          ;COMPARE WITH ASCII 7
12346  070462  103410                          BLO     O.CLGL          ;CHECK LEGALITY IF NOT OCTAL
12347  070464  042700  177770                  BIC     #177770,R0      ;CONVERT TO BCD
12348  070470  006304                          ASL     R4              ; MAKE ROOM
12349  070472  006304                          ASL     R4              ;  IN
12350  070474  006304                          ASL     R4              ;    R4
12351  070476  060004                          ADD     R0,R4           ;PACK THREE BITS IN R4
12352  070500  005202                          INC     R2              ;R2 HAS NUMERIC FLAG
12353  070502  000760                          BR      O.SCAN          ;   AND TRY AGAIN
12354  070504  005001          O.CLGL: CLR     R1              ;CLEAR INDEX
12355  070506  120061  072161  O.LGL1: CMPB    R0,O.LGCH(R1)   ;DO THE CODES MATCH?
12356  070512  001405                          BEQ     O.LGL2          ;JUMP IF YES
12357  070514  005201                          INC     R1              ; SET INDEX FOR NEXT SEARCH
12358  070516  020127  000014                  CMP     R1,#O.CLGT      ;IS THE SEARCH DONE?
12359  070522  103336                          BHIS    O.ERR           ;    OOPS!
12360  070524  000770                          BR      O.LGL1          ;RE-LOOP
12361  070526  006301          O.LGL2: ASL     R1              ;MULTIPLY BY TWO
12362  070530  000171  070534                  JMP     @O.LGDR(R1)     ;GO TO PROPER ROUTINE
12363                                  ;
12364  070534  070564          O.LGDR: O.WRD   ;  /    OPEN WORD
12365  070536  070616                  O.CRET  ;  CARRIAGE RETURN    CLOSE
12366  070540  070272                  O.REGT  ;  $  REGISTER OPS
12367  070542  071126                  O.GO    ;  G  GO TO ADDRESS K
12368  070544  070630                  O.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
12369  070546  070342                  O.ORPC  ;     OPEN RELATED, INDEX - PC
12370  070550  070662                  O.BACK  ;  ^  OPEN PREVIOUS
12371  070552  070672                  O.OFST  ;  O  OFFSET
12372  070554  070750                  O.WSCH  ;  W  SEARCH WORD
12373  070556  070744                  O.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
12374  070560  070370                  O.BKPT  ;  B  BREAKPOINTS
12375  070562  071234                  O.PROC  ;  P  PROCEED
12376          000030          O.LGL   =       .-O.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS
12377                                  ;
12378                                  ; PROCESS / - OPEN WORD
12379                                  ;
12380  070564  005702          O.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
12381  070566  001410                  BEQ     O.WRDA          ;SKIP OTHERWISE
12382  070570  010405                  MOV     R4,R5           ;   PUT VALUE IN CAD
12383  070572  006205          O.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
12384  070574  103711          O.ERR2: BCS     O.ERR           ;JUMP IF ODD ADDRESS
12385  070576  006305                  ASL     R5              ;RESTORE THE CARRY BIT
12386  070600  011500                  MOV     @R5,R0          ;GET CONTENTS OF WORD
12387  070602  004537  071666          JSR     5,O.CADV        ;GO GET AND TYPE OUT @CAD
12388  070606  000714                  BR      O.DCD1          ;GO BACK TO DECODER
12389  070610  042705  000001  O.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
```

```
12390  070614  000766                      BR      0.WRD1          ;GO BACK TO MAIN-LINE
12391                                      ;
12392                                      ; PROCESS CARRIAGE RETURN
12393                                      ;
12394  070616  004737  072076              0.CRET: JSR     PC,0.TCLS       ;CLOSE LOCATION
12395  070622  052705  000001                      BIS     #1,R5           ;CLOSE EVERYTHING
12396  070626  000702                      BR      0.DCD           ;RETURN TO DECODER
12397                                      ;
12398                                      ; PROCESS <LF>, OPEN NEXT WORD
12399                                      ;
12400  070630  004737  072076              0.OP1:  JSR     PC,0.TCLS       ;CLOSE PRESENT CELL
12401  070634  005725                      TST     (R5)+           ;GENERATE NEW ADDRESS
12402  070636  004537  072122              0.OP2:  JSR     5,0.CRLF        ;<CR><LF>
12403  070642  010500                      MOV     R5,R0           ;NUMBER TO TYPE
12404  070644  004537  071666                      JSR     5,0.CADV        ; TYPE OUT ADDRESS
12405  070650  012700  000057                      MOV     #'/,R0          ;TYPE A /
12406  070654  004537  072030                      JSR     5,0.FTYP
12407  070660  000744                      BR      0.WRD1          ;GO PROCESS IT
12408                                      ;
12409                                      ; PROCESS ^, OPEN PREVIOUS WORD
12410                                      ;
12411  070662  004737  072076              0.BACK: JSR     PC,0.TCLS
12412  070666  005745                      TST     -(R5)           ;GENERATE NEW ADDRESS
12413  070670  000762                      BR      0.OP2           ;GO DO THE REST
12414                                      ;
12415                                      ; PROCESS 0, COMPUTE OFFSET
12416                                      ;
12417  070672  006205                      0.OFST: ASR     R5              ;GET LOW ORDER BIT
12418  070674  103737                      BCS     0.ERR2          ;ERR IF CLOSED
12419  070676  006305                      ASL     R5              ;RESTORE WORD
12420  070700  012700  000040                      MOV     #' ,R0          ;TYPE ONE BLANK
12421  070704  004537  072030                      JSR     5,0.FTYP        ; AS A SEPARATOR
12422  070710  160504                      SUB     R5,R4           ;COMPUTE
12423  070712  005304                      DEC     R4
12424  070714  005304                      DEC     R4              ;    16 BIT OFFSET
12425  070716  010400                      MOV     R4,R0           ;TYPE A
12426  070720  010402                      MOV     R4,R2           ;SAVE R4
12427  070722  004537  071666                      JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
12428  070726  010200                      MOV     R2,R0
12429  070730  006200                      ASR     R0              ;DIVIDE BY TWO
12430  070732  103402                      BCS     0.OF1           ;BRANCH IF ODD
12431  070734  004537  071666                      JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
12432  070740  000137  070440              0.OF1:  JMP     0.DCD1          ;ALL DONE
12433                                      ;
12434                                      ; SEARCHES - $MSK   HAS THE MASK
12435                                      ;            $MSK+2 HAS THE FWA
12436                                      ;            $MSK+4 HAS THE LWA
12437                                      ;
12438  070744  005201                      0.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
12439  070746  000401                      BR      0.WDS
12440  070750  005001                      0.WSCH: CLR     R1              ;SET WORD SEARCH
12441  070752  005702                      0.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
12442  070754  001621                      0.ERR1: BEQ     0.ERR           ;ERR IF NO OBJECT
12443  070756  013702  070132                      MOV     0.MSK+2,R2      ;SET ORIGIN
12444  070762  013705  070130                      MOV     0.MSK,R5        ;SET MASK
12445  070766  005105                      COM     R5              ;AND COMPLEMENT IT
```

```
12446  070770  020237  070134          O.WDS2: CMP     R2,O.MSK+4      ; IS THE SEARCH ALL DONE?
12447  070774  101217                          BHI     O.DCD           ;  YES
12448  070776  011200                          MOV     @R2,R0          ; GET OBJECT
12449  071000  005701                          TST     R1              ;NO
12450  071002  001027                          BNE     O.EFF1          ;BRANCH IF EFFECTIVE SEARCH
12451  071004  010046                          MOV     R0,-(SP)
12452  071006  010403                          MOV     R4,R3           ;EXCLUSIVE OR
12453  071010  040400                          BIC     R4,R0           ; IS DONE
12454  071012  042603                          BIC     (SP)+,R3                ;  IN A VERY
12455  071014  050003                          BIS     R0,R3           ;     FANCY MANNER HERE
12456  071016  040503                          BIC     R5,R3           ;AND RESULT WITH MASK
12457  071020  001016                  O.WDS3: BNE     O.WDS4          ;RE-LOOP IF NO MATCH
12458  071022  010446                          MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
12459  071024  004537  072122                  JSR     5,O.CRLF        ;TYPE <CR,LF>
12460  071030  010200                          MOV     R2,R0           ;GET READY TO TYPE
12461  071032  004537  071666                  JSR     5,O.CADV        ;  TYPE ADDRESS
12462  071036  012700  000057                  MOV     #'/,R0          ;SLASH TO R0
12463  071042  004537  072030                  JSR     5,O.FTYP        ;TYPE IT
12464  071046  011200                          MOV     @R2,R0          ;GET CONTENTS
12465  071050  004537  071666                  JSR     5,O.CADV        ;TYPE CONTENTS
12466  071054  012604                          MOV     (SP)+,R4        ; RESTORE R4
12467  071056  005722                  O.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
12468  071060  000743                          BR      O.WDS2          ;     RETURN
12469  071062  020004                  O.EFF1: CMP     R0,R4           ; IS (X)=K?
12470  071064  001755                          BEQ     O.WDS3          ;TYPE IF EQUAL
12471  071066  010003                          MOV     R0,R3           ;(X) TO R3
12472  071070  060203                          ADD     R2,R3           ;(X)+X
12473  071072  005203                          INC     R3
12474  071074  005203                          INC     R3              ;(X)+X+2
12475  071076  020304                          CMP     R3,R4           ;IS (X)+X+2=K?
12476  071100  001747                          BEQ     O.WDS3          ;BRANCH IF EQUAL
12477  071102  042700  177400                  BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
12478  071106  110000                          MOVB    R0,R0           ;EXTEND SIGN
12479  071110  000257                          CCC
12480  071112  006300                          ASL     R0              ;MULTIPLY BY TWO
12481  071114  005200                          INC     R0              ;ADD TWO
12482  071116  005200                          INC     R0
12483  071120  060200                          ADD     R2,R0           ;ADD PC
12484  071122  020004                          CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
12485  071124  000735                          BR      O.WDS3
12486                                  ;
12487                                  ; PROCESS G - GO
12488                                  ;
12489  071126  105037  072151          O.GO:   CLRB    O.P             ;DISALLOW PROCEED
12490  071132  006204                          ASR     R4              ;CHECK LOW ORDER BIT
12491  071134  103617                          BCS     O.ERR?          ;ERR IF ODD NUMBER
12492  071136  006304                          ASL     R4              ;RESTORE WORD
12493  071140  010437  0701<2                  MOV     R4,O.UPC        ;SET UP NEW PC
12494  071144  112737  000340  177776          MOVB    #O.STM,ST       ;SET HIGH PRIORITY
12495  071152  004537  071620                  JSR     5,O.RSTT        ;RESTORE TELETYPE
12496  071156  105037  072150          O.TBIT: CLRB    O.T             ;CLEAR BOTH
12497  071162  042737  000020  070124          BIC     #O.TBT,O.UST    ;  T-BIT FLAGS
12498  071170  017737  176742  070142          MOV     @O.ADR1,O.UIN   ;SAVE INSTRUCTION
12499  071176  013777  072212  176732          MOV     O.TRTC,@O.ADR1  ;REPLACE WITH TRAP
12500  071204  012600                  O.GO2:  MOV     (SP)+,R0        ;RESTORE
12501  071206  012601                          MOV     (SP)+,R1        ; R0
```

```
12502  071210  012602                        MOV     (SP)+,R2        ;   THRU
12503  071212  012603                        MOV     (SP)+,R3
12504  071214  012604                        MOV     (SP)+,R4        ;
12505  071216  012605                        MOV     (SP)+,R5        ;    R5
12506  071220  012606                        MOV     (SP)+,SP        ; AND SP
12507  071222  013746  070124                MOV     0.UST,-(SP)     ;    AND STATUS
12508  071226  013746  070122                MOV     0.UPC,-(SP)     ;    AND PC
12509  071232  000006                0.RTIT: RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
12510                                        ;
12511                                        ; PROCESS P - PROCEED
12512                                        ;   ONLY ALLOWED AFTER A BREAKPOINT
12513                                        ;
12514  071234  105737  072151        0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
12515  071240  001645                        BEQ     0.ERR1          ;NOT LEGAL
12516  071242  105037  072151                CLRB    0.P             ;CLEAR PROCEED FLAG
12517  071246  005702                        TST     R2              ;WAS COUNT SPECIFIED?
12518  071250  001402                        BEQ     0.PR1           ;NO
12519  071252  010437  070140                MOV     R4,0.CT         ;YES, PUT AWAY COUNT
12520  071256  112737  000340  177776 0.PR1: MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
12521  071264  004537  071620                JSR     5,0.RSTT        ;RESTORE TTY
12522  071270  112737  000340  177776 0.C1:  MOVB    #0.STM,ST       ;SET HIGH PRIORITY
12523  071276  105237  072150                INCB    0.T             ;SET T-BIT FLAG
12524  071302  052737  000020  070124        BIS     #0.TBT,0.UST    ;SET T-BIT
12525  071310  000735                        BR      0.GO2
12526                                        ;
12527                                        ; BREAKPOINT HANDLER
12528                                        ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
12529                                        ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
12530                                        ;  AND GIVES CONTROL TO THE CMD DECODER
12531                                        ;
12532  071312  012637  070122        0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
12533  071316  012637  070124                MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
12534  071322  004037  071530        0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
12535  071326  105737  072150                TSTB    0.T             ;CHECK FOR T-BIT SET
12536  071332  001311                        BNE     0.TBIT          ;JUMP IF SET
12537  071334  013777  070142  176574        MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
12538  071342  105737  070126                TSTB    0.PRI           ;CHECK IF PRIORITY
12539  071346  100003                        BPL     0.BK2           ; IS AS SAME AS USER PGM
```

```
 12540  071350  113705  070124                MOVB    0.UST,R5        ;PICK UP USER UST IF SO
 12541  071354  000407                        BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
```

```
 12542  071356  113705  070126          0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
```

```
12543  071362  000257                          CCC                     ;CLEAR CARRY
12544  071364  106005                          RORB    R5              ;SHIFT LOW ORDER BITS
12545  071366  106005                          RORB    R5              ;  INTO
12546  071370  106005                          RORB    R5              ;    HIGH ORDER
12547  071372  106005                          RORB    R5              ;      POSITION
12548  071374  110537  177776          0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
12549  071400  013705  070122                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
12550  071404  005745                          TST     -(R5)           ;SUBTRACT TWO
12551  071406  010537  070122                  MOV     R5,0.UPC        ;FROM THE USER'S PC
12552  071412  020537  070136                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
12553  071416  001417                          BEQ     0.B2            ;JUMP IF FOUND
12554  071420  004537  071566                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
12555  071424  004537  072122                  JSR     5,0.CRLF
12556  071430  012704  072154                  MOV     #0.BD,R4        ;ERR, NOTHING FOUND
12557  071434  012703  072155                  MOV     #0.BD+1,R3
12558  071440  004537  072014                  JSR     5,0.TYPE        ;OUTPUT 'BE'' FOR BAD ENTRY
12559  071444  010500                          MOV     R5,R0
12560  071446  042737  000020  070124          BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
12561  071454  000420                          BR      0.B3            ; AND CONTINUE
12562  071456  005337  070140          0.B2:   DEC     0.CT
12563  071462  003302                          BGT     0.C1            ;JUMP IF REPEAT
12564  071464  012737  000001  070140          MOV     #1,0.CT         ;RESET COUNT TO 1
12565  071472  105237  072151                  INCB    0.P             ;ALLOW PROCEED
12566  071476  004537  071566                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
12567  071502  012700  000102                  MOV     #'B,R0
12568  071506  004537  072030                  JSR     5,0.FTYP        ;TYPE 'B''
12569  071512  013700  070136                  MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
12570  071516  004537  071666          0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
12571  071522  005005                          CLR     R5              ;CLEAR CAD
12572  071524  000137  070434                  JMP     0.DCD           ;GO TO DECODER
12573                                  ;
12574                                  ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
12575                                  ;
12576  071530  012637  072146          0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
12577  071534  010637  070120                  MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
12578  071540  012706  070120                  MOV     #0.USP,SP       ;SET TO INTERNAL STACK
12579  071544  010546                          MOV     R5,-(SP)        ;SAVE
12580  071546  010446                          MOV     R4,-(SP)        ; REGISTERS
12581  071550  010346                          MOV     R3,-(SP)        ;1
12582  071552  010246                          MOV     R2,-(SP)        ; THRU
12583  071554  010146                          MOV     R1,-(SP)        ;    5
12584  071556  013746  072146                  MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
12585  071562  005746                          TST     -(SP)
12586  071564  000200                          RTS     R0
12587                                  ;
12588                                  ; SAVE TELETYPE STATUS
12589                                  ;
12590  071566  113737  177560  072152  0.SVTT: MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
12591  071574  113737  177564  072153          MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
12592  071602  105037  177560                  CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
12593  071606  105037  177564                  CLRB    0.TCSR          ;  BITS IN BOTH C/SR
12594  071612  004537  072122                  JSR     5,0.CRLF        ;TYPE <CR,LF>
12595  071616  000205                          RTS     R5
12596                                  ;
12597                                  ; RESTORE TELETYPE STATUS
12598                                  ;
```

```
12599  071620  004537  072122          O.RSTT: JSR     5,O.CRLF                ;<CR,LF> BEFORE RESTORING
12600  071624  105737  177564                  TSTB    O.TCSR          ;WAIT READY ON PRINTER
12601  071630  100375                          BPL     .-4
12602  071632  032737  004000  177560          BIT     #4000,O.RCSR    ;CHECK BUSY FLAG ON READER
12603  071640  001403                          BEQ     O.RSE1          ;SKIP READY LOOP IF NOT BUSY
12604  071642  105737  177560                  TSTB    O.RCSR          ;WAIT READY
12605  071646  100375                          BPL     .-4             ;    ON READER
12606  071650  113737  072152  177560  O.RSE1: MOVB    O.CSR1,O.RCSR   ;RESTORE
12607  071656  113737  072153  177564          MOVB    O.CSR2,O.TCSR   ;  THE STATUS REGISTERS
12608  071664  000205                          RTS     R5
12609                                  ;
12610                                  ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
12611                                  ; WORD IS IN R0
12612                                  ;
12613  071666  010246                  O.CADV: MOV     R2,-(SP)        ;SAVE R2
12614  071670  012704  072211                  MOV     #O.BUF+6,R4     ;BUFFER START ADDRESS
12615  071674  012746  000060                  MOV     #'0,-(SP)               ;CONSTANT ASCII 0
12616  071700  010002                  O.SPC:  MOV     R0,R2           ; GET
12617  071702  042702  177770                  BIC     #177770,R2      ;    OCTAL CHARACTER
12618  071706  061602                          ADD     @SP,R2          ;CONVERT TO ASCII
12619  071710  110244                          MOVB    R2,-(R4)        ;STORE IN BUFFER
12620  071712  006200                          ASR     R0              ;SHIFT THIS MESS
12621  071714  006200                          ASR     R0              ; RIGHT
12622  071716  006200                          ASR     R0              ;   THREE WHOLE PLACES
12623  071720  020427  072204                  CMP     R4,#O.BUF+1     ;DONE?
12624  071724  101365                          BHI     O.SPC           ;    NO
12625  071726  042700  177776                  BIC     #177776,R0      ;GET LAST BIT
12626  071732  062600                          ADD     (SP)+,R0                ;CONVERT TO ASCII
12627  071734  110044                          MOVB    R0,-(R4)        ;AND PUT IT AWAY
12628  071736  012703  072211                  MOV     #O.BUF+6,R3     ;LWA
12629  071742  004537  072014                  JSR     5,O.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
12630  071746  012602                          MOV     (SP)+,R2        ;RESTORE R2
12631  071750  000205                          RTS     R5
12632                                  ;
12633                                  ; GENERAL CHARACTER INPUT ROUTINE
12634                                  ; CHARACTER INPUT GOES TO R0
12635                                  ;
12636  071752  105737  177560          O.GET:  TSTB    O.RCSR          ;WAIT FOR
12637  071756  100375                          BPL     .-4             ;   INPUT FROM KEYBOARD
12638  071760  113700  177562                  MOVB    O.RDB,R0        ;GET A CHARACTER
12639  071764  004537  072030                  JSR     5,O.FTYP        ;ECHO CHARACTER
12640  071770  042700  177600                  BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
12641  071774  001766                          BEQ     O.GET           ;IGNORE NULLS
12642  071776  122700  000040                  CMPB    #40,R0          ;CHECK FOR SPACES
12643  072002  001763                          BEQ     O.GET           ;IGNORE NULLS
12644  072004  122700  000073                  CMPB    #';,R0          ;CHECK FOR SEMI-COLON
12645  072010  001760                          BEQ     O.GET           ;IGNORE THEM IF FOUND
12646  072012  000205                          RTS     R5
12647                                  ;
12648                                  ; GENERAL CHARACTER OUTPUT ROUTINE
12649                                  ;  ADDRESS OF FIRST BYTE IN R4,
12650                                  ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
12651                                  ;
12652  072014  020304                  O.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
12653  072016  103426                          BLO     O.TYP1          ; EXIT WHEN DONE
12654  072020  112400                          MOVB    (R4)+,R0        ;GET A CHARACTER
```

```
12655  072022  004537  072030          JSR     5,O.FTYP        ;TYPE ONE CHARACTER
12656  072026  000772                  BR      O.TYPE          ;LOOP UNTIL DONE
12657                          ;
12658                          ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
12659                          ;
12660  072030  105737  177564  O.FTYP: TSTB    O.TCSR          ;CHECK STATUS
12661  072034  100375                  BPL     .-4             ;WAIT UNTIL READY
12662  072036  110037  177566          MOVB    R0,O.TDB        ;TYPE ONE CHARACTER
12663  072042  120037  000045          CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
12664  072046  001012                  BNE     O.TYP1          ;NO
12665  072050  113746  000044          MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
12666  072054  105737  177564  O.TYP2: TSTB    O.TCSR
12667  072060  100375                  BPL     O.TYP2
12668  072062  105037  177566          CLRB    O.TDB           ;GENERATE NULL FILLER
12669  072066  105316                  DECB    @SP
12670  072070  003371                  BGT     O.TYP2
12671  072072  005726                  TST     (SP)+           ;POP STACK
12672  072074  000205          O.TYP1: RTS     R5
12673                          ;
12674                          ; CLOSE WORD OR BYTE AND EXIT,
12675                          ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
12676                          ;
12677  072076  006205          O.TCLS: ASR     R5              ;GET LOW ORDER BIT
12678  072100  103405                  BCS     O.TC            ;JUMP IF ALREADY CLOSED
12679  072102  006305                  ASL     R5
12680  072104  005702                  TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
12681  072106  001401                  BEQ     O.CLS1          ;NO CHANGE TO THE OPEN CELL
12682  072110  010415                  MOV     R4,@R5          ;STORE WORD
12683  072112  000207          O.CLS1: RTS     PC
12684  072114  005746          O.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
12685  072116  000137  070420          JMP     O.ERR           ;AND SCREAM BLOODY MURDER
12686                          ;
12687                          ; O.CRLF - TYPE <CR,LF>
12688                          ; O.CRLS - TYPE <CR,LF>*
12689                          ;
12690  072122  012703  072157  O.CRLF: MOV     #O.CR+1,R3      ;LWA <CR,LF>
12691  072126  000402                  BR      O.CRS
12692  072130  012703  072160  O.CRLS: MOV     #O.CR+2,R3      ;LWA <CR,LF>*
12693  072134  012704  072156  O.CRS:  MOV     #O.CR,R4        ;FWA
12694  072140  004537  072014          JSR     5,O.TYPE        ;TYPE SOMETHING
12695  072144  000205                  RTS     R5
12696                          ;
12697  072146  000000          O.XXX:  .WORD   0               ;TEMPORARY STORAGE
12698  072150     000          O.T:    .BYTE   0               ;  T-BIT FLAG
12699  072151     000          O.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
12700                                                          ;               = 1 IF PROCEED ALLOWED
12701  072152     000          O.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
12702  072153     000          O.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
12703                          ;
12704                                  .EVEN
12705  072154  042502          O.BD:   .WORD   'BE
12706                          ;
12707  072156     015          O.CR:   .BYTE   015     ;  <CR>
12708  072157     012                  .BYTE   012     ;  <LF>
12709  072160     052                  .BYTE   '*      ;  *
12710                          ;
```

```
12711  072161  057              O.LGCH: .BYTE   '/      ;  /
12712  072162  015                      .BYTE   015     ;  CARRIAGE RETURN
12713  072163  044                      .BYTE   '$      ;  $
12714  072164  107                      .BYTE   'G      ;  G
12715  072165  012                      .BYTE   012     ;  <LF>
12716  072166  137                      .BYTE   '       ;
12717  072167  136                      .BYTE   '^      ;  ^
12718  072170  117                      .BYTE   'O      ;  O
12719  072171  127                      .BYTE   'W      ;  W
12720  072172  105                      .BYTE   'E      ;  E
12721  072173  102                      .BYTE   'B      ;  B
12722  072174  120                      .BYTE   'P      ;  P
12723          000014           O.CLGT  =       .-O.LGCH                ;TABLE LENGTH
12724                                   ;
12725  072175  123              O.TL:   .BYTE   'S      ;DO       1
12726  072176  120                      .BYTE   'P      ;NOT      2
12727  072177  115                      .BYTE   'M      ;CHANGE   3
12728  072200  000                      .BYTE   0       ;THE      4
12729  072201  000                      .BYTE   0       ;ORDER    5
12730  072202  102                      .BYTE   'B      ;HERE     6
12731          000006           O.LG    =       .-O.TL
12732                                   ;
12733  072203                   O.BUF:                          ;6 CHAR. BUFFER WITH
12734          072211           .       =       .+6
12735  072211  040                      .BYTE                   ;TRAILING BLANK
12736                                   .EVEN
12737                                   ;
12738  072212  000003           O.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
12739                                   ;
12740                                   ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
12741                                   ;
12742          070104           .       =       O.ODT-40
12743  070104  000000           O.UR0:  0       ;USER R0
12744  070106  000000                   0       ;     R1
12745  070110  000000                   0       ;     R2
12746  070112  000000                   0       ;     R3
12747  070114  000000                   0       ;     R4
12748  070116  000000                   0       ;     R5
12749  070120  000000           O.USP:  0       ;USER SP
12750  070122  000000           O.UPC:  0       ;USER PC
12751  070124  000000           O.UST:  0       ;USER ST
12752  070126  000007           O.PRI:  7       ;ODT PRIORITY
12753  070130  000000           O.MSK:  0       ;MASK
12754  070132  000000                   0       ;LOW LIMIT
12755  070134  000000                   0       ;HIGH LIMIT
12756                                   ;
12757                                   ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
12758                                   ;   UIN = CONTENTS
12759                                   ;
12760  070136  000000           O.ADR1: 0
12761  070140  000000           O.CT:   0
12762  070142  000000           O.UIN:  0
12763          000001                   .END
```

```
ABASE = 177440           2323    2364    2378#
ACDW1 = 000000           2323    2366
ACDW2 = 000000           2323    2367
ACLO  = 000010           1383#
ACPUOP= 000000           2323    2338
ACT11   003436           2542#   3837*
ADDW0 = 000000           2323    2368
ADDW1 = 000000           2323    2369
ADDW10= 000000           2323
ADDW11= 000000           2323
ADDW12= 000000           2323
ADDW13= 000000           2323
ADDW14= 000000           2323
ADDW15= 000000           2323
ADDW2 = 000000           2323    2370
ADDW3 = 000000           2323    2371
ADDW4 = 000000           2323    2372
ADDW5 = 000000           2323    2373
ADDW6 = 000000           2323    2374
ADDW7 = 000000           2323    2375
ADDW8 = 000000           2323
ADDW9 = 000000           2323
ADEVCT= 000000           2323    2329
ADEVM = 000000           2323    2365
AENV  = 000000           2323    2334
AENVM = 000000           2323    2335
AFATAL= 000000           2323    2326
AMADR1= 000000           2323    2351
AMADR2= 000000           2323    2355
AMADR3= 000000           2323    2358
AMADR4= 000000           2323    2361
AMAMS1= 000000           2323    2345
AMAMS2= 000000           2323    2353
AMAMS3= 000000           2323    2356
AMAMS4= 000000           2323    2359
AMSGAD= 000000           2323    2331
AMSGLG= 000000           2323    2332
AMSGTY= 000000           2323    2325
AMTYP1= 000000           2323    2346
AMTYP2= 000000           2323    2354
AMTYP3= 000000           2323    2357
AMTYP4= 000000           2323    2360
APASS = 000000           2323    2328
APRIOR= 000000           2323
APTCSU= 000040          10161   10350#
APTENV= 000001           8918   10108   10154   10306   10348#
APTSIZ= 000200           3763   10347#
APTSPO= 000100          10156   10308   10349#
ASWREG= 000000           2323    2336
ATESTN= 000000           2323    2327
ATTN    003312           2474#   5333    9125    9144    9170
AUNIT = 000000           2323    2330
AUSWR = 000000           2323    2337
AVECT1= 000000           2323    2362
AVECT2= 000000           2323    2363
BADHDR  003306           2465#   3845*   5717*   7362*   9873    9921*
```

```
BADTMO  050064          3919    9936#
BAI    = 000020         1346#   5726    5877
BA16   = 000400         1332#
BA17   = 001000         1333#
BIT0   = 000001         1275#   1329    1361    1380    2532    3958    9796
BIT00  = 000001         1265#   1275
BIT01  = 000002         1264#   1274
BIT02  = 000004         1263#   1273
BIT03  = 000010         1262#   1272
BIT04  = 000020         1261#   1271
BIT05  = 000040         1260#   1270
BIT06  = 000100         1259#   1269
BIT07  = 000200         1258#   1268
BIT08  = 000400         1257#   1267    10043
BIT09  = 001000         1256#   1266    10051   10119
BIT1   = 000002         1274#   1362    2533
BIT10  = 002000         1255#   1334    1352    1371    1403    1417    1431    1444    1458    5307    10096
BIT11  = 004000         1254#   1335    1353    1372    1389    1404    1418    1432    1445    1459    5309    10058
BIT12  = 010000         1253#   1336    1354    1373    1405    1419    1433    1446    1460
BIT13  = 020000         1252#   1337    1355    1374    1390    1406    1420    1434    1447    1461    10103
BIT14  = 040000         1251#   1338    1356    1375    1391    1407    1421    1435    1448    1462    1472    9633    9665
                        10029
BIT15  = 100000         1250#   1339    1340    1357    1376    1392    1408    1475    4839    9633    9661
BIT2   = 000004         1273#   1363    1382    2534
BIT3   = 000010         1272#   1345    1364    1383
BIT4   = 000020         1271#   1346    1365    1384    1397    1425    1452    6592
BIT5   = 000040         1270#   1347    1366    1385    1398    1412    1426    1439    1453
BIT6   = 000100         1269#   1330    1348    1367    1386    1399    1413    1427    1440    1454
BIT7   = 000200         1268#   1331    1349    1368    1387    1400    1414    1428    1441    1455    3948
BIT8   = 000400         1267#   1332    1350    1369    1388    1401    1415    1429    1442    1456
BIT9   = 001000         1266#   1333    1351    1370    1402    1416    1430    1443    1457    5303
BPTVEC= 000014          1282#
BSE    = 000200         1368#
BSERR   001456          2446#   6869*   6875*   7144    8885*   8887*
BSE22H  002304          2460#   6821    6872    6891    9659
BSE22S  054474          6882    9663    10979#
BYP     044274          4035    4063    4067    4082    4143    4170    4174    4178    4182    9197#
BYPCER  001462          2449#   3946*   4194*   4918*   4953*   5521*   5602*   6917*   6998*   8813*   9314
BYPFMT  001464          2450#   6007*   6097*   6324*   6414*   9631
BYPT16  001340          2394#   3676*   3681*   3686*   3691*   3696*   3700*   4974
BYT16   007014          1489    3680#
BYT16A  007034          1493    3685#
CALADD  001366          2407#   6013*   6330*   7156*   7186*   7208    7292    7310    7347    7354*   7365*   9627    9637
                        9639    9687
CALDIF  001360          2404#   5075*   5090    5121*   5151    6123*   6439*   6635*   7190*   7752*   8088*   8438*   8792*
                        11901   11925
CCLR   = 100000         1340#   4632    4657    4684    4697    4770    4849    4886    4997    5034    5082    5096    5128
                        5157    5208    5245    5348    5354    5370    5434    5543    5573    5639    6178    6444    6500
                        6696    6939    6969    7035    7267    7583    7801    7926    8137    8262    8487    8612    8834
                        8840    8856    9354    9362    9370    9378    9877    9892
CCYL    001354          2402#   4987*   5069*   5070    5076    5115*   5116    5122    7878*   8214*   8564*
CDT    = 002000         1334#   4051    4153    4233    4235    4514
CERR   = 100000         1339#   4012    4127    4462    4537    4584    5409    5741    5795    5886    5930    6024    6060
                        6153    6223    6267    6341    6377    6470    6534    6576    6671    6754    6832    7080    7172
                        7242    7317    7380    7443    7456    7558    7628    7678    7776    7845    7901    7970    8018
                        8112    8181    8237    8306    8354    8416    8462    8531    8587    8656    8704    8774    9316
```

```
                    9411
CFMT  = 010000      1336#  5882   5893   5927   5949
CHKFLG  001466      2454#  9207*  9290   9294   9298
CHKMSG  044310      4345   4623   4646   4675   4719   4746   4872   4904   4938   5020   5052   5231   5263
                    5312   5422   5452   5536   5561   5589   5625   5657   5756   5822   5910   5966   6039
                    6135   6166   6196   6283   6356   6456   6483   6518   6594   6655   6684   6714   6844
                    6932   6957   6985   7021   7053   7221   7255   7285   7332   7540   7571   7601   7764
                    7789   7819   7888   7914   7944   8100   8125   8155   8224   8250   8280   8450   8475
                    8505   8574   8600   8630   8819   9207#
CKCERR  045050      9075   9079   9091   9096   9314#
CKSWR = 104407     10028  10092  10118  10968#
CLEAR = 000005      1317#  4659   4888   5036   5098   5159   5247   5356   5436   5545   5641   6180   6502
                    6698   6941   7037   7269   7585   7803   7928   8139   8264   8489   8614   8842   9894
CLKOF   047340      4696   4710   4737   4763   4811   4816   9767#
CLKON   047244      4687   4701   4728   4754   9745#
CLOCK   047304      3868   9756#
CLRFLG  043134      3786   8930#
COE   = 001000      1370#
COUNT   001372      2411#  4685*  4699*  4726*  4752*  9757*  9759*
CR    = 000015      1190#  10217  10227  10982  10986  10988  10992  11002  11011  11018  11026  11031  11036
                   11041  11047  11054  11058  11065  11070  11079  11084  11088  11092  11095  11100  11106
                   11110  11114  11120  11125  11129  11134  11138  11140  11143  11147  11152  11156  11166
                   11174  11183  11188  11194  11198  11201  11204  11211  11229  11236  11239  11457  11527
                   11659  11663  11669  11676
CRLF  = 000200      1191#  10171  10227
CTO   = 004000      1335#  9320
CYLADD  001364      2406#  4357   4652   4881   4945   5029   5085   5131   5240   5324   5392   5459   5634
                    5769   5835   5978   6203   6204   6289   6495   6600   6721   7030   7292   7608   7826
                    7951   8162   8287   8512   8637   8828   9491*  9492*  9493*  9494*  9495*  9496*  9514*
                    9515*  9516*  9517*  9518*  9519*
CYLDIF  001362      2405#  4354   4949   5090   5140   5146   5151   5321   5428   5766   5832   5975   6142
                    6172   6492   6662   6690   7228   7261   7547   7577   7795   7920   8131   8256   8481
                    8606   8825   9476*  9477*  9478*  9479*  9480*  9481*  9482   9484*  9501*  9502*  9503*
                    9504*  9505*  9506*  9507   9509*
CYLINV  042210      8071   8746#
DATA0   001444      2439#  5735   5848
DATA01  001446      2440#
DATA1   001450      2441#  5878   5985
DATCMD  043530      5738   5793   5883   5928   6021   6058   6221   6338   6375   6532   6829   7169   7314
                    7454   7626   7843   7968   8179   8304   8529   8654   9055#
DCK   = 100000      1376#
DCLO  = 000020      1384#
DCPAR = 020000      1337#  4562   4581
DDISP = 177570      1197#  2298   3751
DDPCH   003434      2541#  3833*  4020   4226
DDT   = 000400      1388#  4488   4496   4523   4531
DDUMP   003432      2540#  3799*  8930
DF1     067144      2582   2588   2594   2600   2605   2611   2617   2623   2691   2697   2708   2713   2718
                    2723   2728   2733   2738   2743   2748   2753   2758   2868   2893   2918   2943   3003
                    3018   3073   3078   3083   3118   3148  11988#
DF10    067270      2629   2635   2640   2645   2650   2686   2703   2813   2863   2978   2983   2988   2993
                    2998   3008   3013   3023   3028   3033   3038   3088   3108   3113   3128   3133   3138
                    3143   3173   3178   3183   3193   3268   3273   3308   3323   3328   3333   3338   3373
                    3383   3388   3409   3414   3479   3575   3580   3585   3600   3605   3615   3620   3625
                    3630   3635  12037#
DF12    067310      3093  12046#
```

```
DF14    067334  3238  12057#
DF15    067354  3203  3208  3228  3233  3343  12067#
DF16    067374  3188  3223  12076#
DF17    067414  3348  3363  12085#
DF2     067160  3660  3665  3671  11995#
DF20    067440  2655  2660  2665  2670  2793  2798  2823  2828  2833  2838  2843  2848  2853
                2858  2873  2878  2883  2888  2898  2903  2908  2913  2923  2928  2933  2938
                2958  2963  2968  2973  3043  3048  3053  3058  3153  3158  3163  3168  3243
                3248  3253  3258  3283  3288  3293  3298  3313  3318  3439  3444  3449  3454
                3494  3499  3504  3509  3514  3519  3524  3529  3534  3539  3544  3549  3554
                3559  3640  3645  12095#
DF21    067474  3590  3595  12109#
DF22    067514  2675  2680  2763  2768  2783  2788  2948  2953  3063  3068  3098  3103  3198
                3213  3218  3489  3564  3569  3650  3655  12117#
DF23    067550  2803  2808  2818  12132#
DF3     067204  3378  12006#
DF4     067210  3610  12008#
DF5     067224  3358  12015#
DF6     067244  2773  2778  3263  3278  3303  3393  3398  3404  3419  3424  3429  3434  3459
                3464  3484  12024#
DH1     064170  2580  2586  2592  2598  2603  2609  2615  2621  2689  2695  2706  2711  2716
                2721  2726  2731  2736  2741  2746  2751  2756  2866  2891  2916  2941  3001
                3016  3071  3076  3081  3116  3146  3236  3588  3593  3608  11688#  11999  12017
                12026  12039  12048  12069  12078  12089  12097  12119  12134
DH10    064451  2841  2846  2851  2856  2861  3211  3216  11721#
DH11    064500  2871  2876  2881  2886  3623  11725#
DH12    064537  2896  2901  2906  2911  3628  11731#
DH13    064600  2921  2926  2931  2936  3633  11737#
DH14    064647  2761  2766  2771  2776  2946  2951  2996  3011  3096  3101  3391  3396  3482
                11744#
DH16    064677  2806  2966  2971  3638  3643  3648  3653  11749#
DH17    064741  2781  2786  2801  2811  3006  3131  3306  3311  3316  3321  3326  3331  3386
                3532  3537  11755#
DH18    064761  2627  2633  2638  2643  2648  3613  11758#
DH19    065002  2701  2976  11761#
DH2     064205  11691#  11990  12001  12010  12019  12030  12041  12052  12061  12071  12080  12091  12105
                12113  12127  12142
DH20    065021  2981  11764#
DH21    065050  2684  2986  3021  3026  11768#
DH22    065076  3111  3126  3381  3477  3492  3497  3522  3527  11772#
DH23    065124  3376  11776#
DH25    065145  3031  3036  3041  3046  3051  3056  3061  3066  3261  3266  3271  3356  3402
                11779#
DH26    065164  2791  2796  2956  2961  3336  3341  11782#
DH28    065210  11786#  12099  12121  12136
DH29    065220  11788#  12103  12125  12140
DH3     064255  11698#  11992  12003  12012  12021  12032  12043  12054  12063  12073  12082  12093  12107
                12115  12129  12144
DH30    065230  3191  3196  3201  3206  3487  3512  3517  3552  3557  3603  11790#
DH32    065256  11794#  12101  12123  12138
DH33    065303  3241  3246  3251  3256  3276  11798#
DH34    065323  3086  11801#
DH35    065347  3091  11805#
DH36    065375  11809#  12050
DH37    065420  3106  11813#
DH38    065464  3151  3156  3161  3166  3171  3176  3181  3573  11819#
```

```
DH39     065502          3226   3231   3502   3507   3542   3547   3562   3567   3598  11822#
DH40     065531         11826#  12059
DH41     065565          3281   3286   3291   3296   3301   3407   3583  11831#
DH42     065606          3361  11834#
DH44     065646          3371  11840#  12087
DH45     065716          3186  11847#
DH46     065775          3221  11855#
DH48     066042          3412   3417   3422   3427   3432   3437   3442   3447   3452   3457   3462  11862#
DH49     066074         11867#  11997
DH51     066150          3136   3141  11875#
DH52     066177         11879#  12111
DH53     066227          3346  11884#
DH6      064326         11705#  12028
DH8      064361          2653   2658   2663   2668   2673   2678   2816  11710#
DH9      064405          2821   2826   2831   2836   2991   3578   3618  11714#
DI     = 040000          1338#
DISPLA   001142          2298#  3751*  3759*  10073*  10095*
DISPRE   000174          1484#  3759
DLT    = 100000          1357#  5806   5941   6070   6234   6387   6545   7467   7639   7856   7981   8192   8317
                         8542   8667
DLY      044256          4008   4123   9188#  9191
DMD    = 000040          1398#
DOCMD    043472          4298   4429   4606   4660   4788   4852   4889   4922   5000   5037   5078   5099   5124
                         5160   5211   5248   5287   5351   5357   5403   5437   5527   5546   5571   5642   5678
                         5894   5950   6128   6181   6261   6442   6503   6570   6648   6699   6748   6923   6942
                         6967   7038   7074   7107   7214   7270   7374   7437   7533   7586   7672   7757   7804
                         7881   7929   8012   8093   8140   8217   8265   8348   8410   8443   8490   8567   8615
                         8698   8768   8795   8837   8843   9044#  9343   9864   9880   9895
DOSEEK   036414          8070   8072#
DOTIM    003470          2560#  3867*  3876*  4629
DRA    = 000001          1380#  4057   4160
DRDY   = 000200          1387#
DRIVS    003442          2544#  3953*  3960*  3977   3992*  4024*  4043   4147*  4213   8883   8985*
DRIV0    003444          2549#  3849   3955   3994   4109   8968
DRIV1    003446          2550#
DRIV2    003450          2551#
DRIV3    003452          2552#
DRIV4    003454          2553#
DRIV5    003456          2554#
DRIV6    003460          2555#
DRIV7    003462          2556#
DROT   = 000040          1385#
DRPAR  = 000010          1364#
DRVMSK= 000007           1344#  4015   4130
DRVPTR   001346          2398#  3849*  4218   4236*
DSC    = 040000          1391#
DSWR   = 177570          1196#  2297   3750
DTE    = 010000          1373#
DTYE   = 000040          1366#  4049   4151   4499   4534
DT1      066276          2581   2587   2593   2599   2604   2610   2616   2622   2628   2634   2639   2644   2649
                         2685   2690   2696   2702   2707   2712   2717   2722   2727   2732   2737   2742   2747
                         2752   2757   2812   2862   2867   2892   2917   2942   2977   2982   2987   2992   2997
                         3002   3007   3012   3017   3022   3027   3032   3037   3072   3077   3082   3087   3107
                         3112   3117   3127   3132   3137   3142   3147   3172   3177   3182   3187   3192   3202
                         3207   3222   3227   3232   3267   3272   3307   3322   3327   3332   3337   3342   3347
                         3357   3362   3372   3382   3387   3408   3413   3478   3574   3579   3584   3599   3604
```

```
                    3609  3614  3619  3624  3629  3634  3659  3664  3670  11894#
DT10     066630     3594  11941#
DT11     066674     3377  11949#
DT13     066700     2654  2659  2664  2669  2792  2797  2822  2827  2832  2837  2842  2847  2852
                    2857  2872  2877  2882  2887  2897  2902  2907  2912  2922  2927  2932  2937
                    2957  2962  2967  2972  3042  3047  3052  3057  3152  3157  3162  3167  3242
                    3247  3252  3257  3282  3287  3292  3297  3312  3317  3438  3443  3448  3453
                    3493  3498  3503  3508  3513  3518  3523  3528  3533  3538  3543  3548  3553
                    3558  3639  3644  11950#
DT14     066760     2674  2679  2762  2767  2782  2787  2947  2952  3062  3067  3097  3102  3197
                    3212  3217  3488  3563  3568  3649  3654  11961#
DT15     067050     2802  2807  2817  11973#
DT4      066336     2772  2777  3262  3277  3302  3418  3423  3428  3433  3458  3463  11901#
DT6      066404     3092  11909#
DT7      066450     3237  11917#
DT8      066516     3392  3397  3403  3483  11925#
DT9      066564     3589  11933#
D.ACLO=  000100     1440#
D.BRHM=  000100     1427#  4339  4620  4640  4669  4716  4743  4866  4898  4928  5014  5046  5225
                    5257  5297  5416  5446  5533  5555  5586  5619  5651  5750  5816  5904  5960
                    6033  6132  6160  6190  6277  6350  6453  6477  6512  6587  6652  6678  6708
                    6838  6929  6951  6982  7015  7047  7218  7249  7279  7326  7537  7565  7595
                    7761  7783  7813  7885  7908  7938  8097  8119  8149  8221  8244  8274  8447
                    8469  8499  8571  8594  8624  8816
D.CART=  000400     1429#  4339  4620  4640  4669  4716  4743  4866  4898  4928  5014  5046  5225
                    5257  5297  5416  5446  5533  5555  5586  5619  5651  5750  5816  5904  5960
                    6033  6132  6160  6190  6277  6350  6453  6477  6512  6587  6652  6678  6708
                    6838  6929  6951  6982  7015  7047  7218  7249  7279  7326  7537  7565  7595
                    7761  7783  7813  7885  7908  7938  8097  8119  8149  8221  8244  8274  8447
                    8469  8499  8571  8594  8624  8816
D.DDT =  000400     1415#  4262  4485  4493  4520  4528
D.DOOR=  000200     1428#  4339  4620  4640  4669  4716  4743  4866  4898  4928  5014  5046  5225
                    5257  5297  5416  5446  5533  5555  5586  5619  5651  5750  5816  5904  5960
                    6033  6132  6160  6190  6277  6350  6453  6477  6512  6587  6652  6678  6708
                    6838  6929  6951  6982  7015  7047  7218  7249  7279  7326  7537  7565  7595
                    7761  7783  7813  7885  7908  7938  8097  8119  8149  8221  8244  8274  8447
                    8469  8499  8571  8594  8624  8816
D.DRA =  000040     1412#  4334  4337  4618  4638  4667  4714  4741  4864  4896  4926  5012  5044
                    5223  5255  5291  5414  5444  5531  5553  5584  5617  5649  5748  5814  5902
                    5958  6031  6130  6158  6188  6275  6348  6451  6475  6510  6585  6650  6676
                    6706  6836  6927  6949  6980  7013  7045  7216  7247  7277  7324  7535  7563
                    7593  7759  7781  7811  7883  7906  7936  8095  8117  8147  8219  8242  8272
                    8445  8467  8497  8569  8592  8622  8814
D.DRDY=  000200     1414#  4638  4667  4757  4864  4896  4926  5012  5044  5223  5255  5291  5414
                    5444  5531  5553  5617  5649  5748  5814  5902  5958  6031  6158  6188  6275
                    6348  6475  6510  6585  6676  6706  6836  6927  6949  7013  7045  7247  7277
                    7324  7563  7593  7781  7811  7906  7936  8117  8147  8242  8272  8467  8497
                    8592  8622  8814
D.DROT=  020000     1447#
D.DSC =  040000     1421#  4638  4864  4926  5012  5223  5291  5414  5531  5584  5617  6158  6275
                    6475  6676  6927  6980  7013  7247  7563  7781  7906  8117  8242  8467  8592
                    8809  8814
D.FLT =  000200     1441#  4575  4927  5296  5532  5585  6928  6981  8803  8815
D.FORM=  001000     1416#  5896  5952
D.FWD =  002000     1431#  4690  4704  4731  4743  6132  6652  7218  7761  8097  8571
D.HDFL=  000200     1455#
```

```
D.HDHM= 000040   1426#  4339  4620  9565
D.IDAE= 000040   1439#  5296  5366  8800  8815  8852
D.ILF = 000400   1442#
D.LIMD= 020000   1461#  5587  5595  6983  6991  9548
D.LOAD= 010000   1433#  9582
D.MHD = 000400   1456#
D.NMOV= 010000   1460#  5587  6983
D.OFF = 002000   1417#
D.PAR = 001000   1443#  4559  4578  4927  5532  6928
D.PIP = 020000   1420#  4714  4741  5584  6130  6451  6650  6980  7216  7535  7759  7883  8095
                 8219  8445  8569  8806
D.PLO = 004000   1459#
D.REV = 004000   1432#  4716  5586  6453  6982  7537  7885  8221  8447
D.RTZ = 020000   1434#  4716  4743  4857  5005  5216  6447  6453  9885
D.SECT= 000020   1452#
D.SKI = 002000   1444#  5296  5585  6981  8815
D.SPIN= 010000   1419#  4610  4618  4638  4667  4714  4741  4773  4864  4896  4926  5012  5044
                 5223  5255  5291  5414  5444  5531  5553  5584  5617  5649  5748  5814  5902
                 5958  6031  6130  6158  6188  6275  6348  6451  6475  6510  6585  6650  6676
                 6706  6836  6927  6949  6980  7013  7045  7216  7247  7277  7324  7535  7563
                 7593  7759  7781  7811  7883  7906  7936  8095  8117  8147  8219  8242  8272
                 8445  8467  8497  8569  8592  8622  8814
D.SPLS= 010000   1446#
D.SPOK= 001000   1430#  4640  4669  4716  4743  4866  4898  4928  5014  5046  5225  5257  5297
                 5416  5446  5533  5555  5586  5619  5651  5750  5816  5904  5960  6033  6132
                 6160  6190  6277  6350  6453  6477  6512  6587  6652  6678  6708  6838  6929
                 6951  6982  7015  7047  7218  7249  7279  7326  7537  7565  7595  7761  7783
                 7813  7885  7908  7938  8097  8119  8149  8221  8244  8274  8447  8469  8499
                 8571  8594  8624  8816  9598
D.SSP = 000020   1425#  4640  4669  4716  4743  4866  4898  4928  5014  5046  5225  5257  5297
                 5416  5446  5533  5555  5586  5619  5651  5750  5816  5904  5960  6033  6132
                 6160  6190  6277  6350  6453  6477  6512  6587  6652  6678  6708  6838  6929
                 6951  6982  7015  7047  7218  7249  7279  7326  7537  7565  7595  7761  7783
                 7813  7885  7908  7938  8097  8119  8149  8221  8244  8274  8447  8469  8'99
                 8571  8594  8624  8816
D.SUNS= 040000   1462#
D.TIB = 002000   1458#
D.UNLD= 040000   1435#  5578  5586  6974  6982
D.UNS = 040000   1448#
D.VV  = 000100   1413#  4332  4337  4419  4432  4618  4638  4667  4714  4741  4864  4896  4926
                 5012  5044  5223  5255  5291  5414  5444  5531  5553  5584  5617  5649  5748
                 5814  5902  5958  6031  6130  6158  6188  6275  6348  6451  6475  6510  6585
                 6650  6676  6706  6836  6927  6949  6980  7013  7045  7216  7247  7277  7324
                 7535  7563  7593  7759  7781  7811  7883  7906  7936  8095  8117  8147  8219
                 8242  8272  8445  8467  8497  8569  8592  8622  8814
D.WCUR= 000040   1453#
D.WGAT= 000100   1454#
D.WLE = 004000   1445#
D.WRL = 004000   1418#
D.XERR= 001000   1457#
ECCW  = 020000   1406#
ECH   = 000100   1367#
EMTVEC= 000030   1285#  3734*  3735*  3767*
EM1     057145   8994  11211#
EM10    057722   2614  11278#
EM11    057765   2620  11284#
```

EM12	060031	2626	2975	2980	2985	2990	2995	3005	3030	3110	3135	3190	3225	3335
		11291#												
EM13	060062	2632	2810	2860	3010	3035	3140	3170	3411	11296#				
EM14	060104	2637	11299#											
EM15	060131	2642	11303#											
EM16	060163	2647	3431	11308#										
EM17	060210	2652	2790	2820	2840	2870	2895	2920	2945	2965	3040	3150	3240	3280
		3310	3436	3521	3541	3551	11312#							
EM18	060225	2657	2795	2825	2845	2875	2900	2925	2950	2970	3045	3155	3245	3285
		3441	3491	3501	3511	3531	11315#							
EM19	060242	2662	2830	2850	2880	2905	2930	2955	3050	3095	3105	3160	3250	3290
		3315	3446	3526	3546	3556	3637	11318#						
EM2	057217	2579	11219#											
EM20	060257	2667	2835	2855	2885	2910	2935	2960	3055	3100	3165	3255	3295	3451
		3496	3506	3516	3536	3642	11321#							
EM21	060274	2683	3205	3230	3265	3305	3340	11324#						
EM22	060316	2688	11327#											
EM23	060355	2694	11333#											
EM24	060435	2700	11342#											
EM25	060461	2705	11346#											
EM26	060507	2710	11350#											
EM27	060527	2715	11353#											
EM28	060550	2720	11356#											
EM29	060626	2725	11364#											
EM3	057306	9940	11229#											
EM30	060675	2730	11371#											
EM31	060726	2735	11376#											
EM32	060753	2740	3421	11380#										
EM33	061000	2745	11384#											
EM34	061035	2750	11389#											
EM35	061066	2755	11394#											
EM36	061141	2760	2770	3260	3355	3390	3401	3461	11402#					
EM37	061201	2765	11408#											
EM38	061241	2775	3275	3300	3395	3456	11414#							
EM39	061312	2780	3020	3060	3195	3210	3320	3561	11421#					
EM4	057354	3185	11236#											
EM40	061347	2785	3025	3065	3215	11426#								
EM41	061375	3486	3566	11430#										
EM42	061436	5597	6993	11436#										
EM43	061466	2800	11440#											
EM44	061515	2805	11444#											
EM45	061551	2940	11449#											
EM46	061672	2672	3647	11463#										
EM47	061705	2677	3652	11465#										
EM48	061720	3085	3090	11467#										
EM49	061733	2915	11469#											
EM5	057467	2585	3668	11250#										
EM50	062021	2865	11479#											
EM51	062072	2890	11486#											
EM52	062172	3070	11497#											
EM53	062231	3075	11503#											
EM54	062262	3080	11508#											
EM55	062315	3125	11513#											
EM56	062350	3375	11518#											
EM57	062373	3607	11522#											
EM58	062427	3612	11527#											

```
EM59    062475            3145   11534#
EM6     057510            2591   11253#
EM60    062526            3175   11539#
EM61    062562            3180   11544#
EM63    062616            3200   11549#
EM64    062637            2815   11552#
EM65    062666            3235   11556#
EM66    062706            3325   11559#
EM67    062742            3270   11564#
EM68    063001            3330    3385   11570#
EM69    063040            3370   11576#
EM7     057531            2597   11256#
EM70    063060            3617    3622    3627    3632   11579#
EM71    063073            3380   11581#
EM72    063114            3220   11584#
EM73    063141            3657   11588#
EM74    063162            3406   11591#
EM75    063207            3416   11595#
EM76    063235            3426   11599#
EM77    063256            3476   11602#
EM78    063304            3481   11606#
EM79    063371            3662   11615#
EM8     057620            2602   11266#
EM80    063412            3572   11618#
EM81    063442            3577   11622#
EM82    063470            3582   11626#
EM83    063515            3587    3592   11630#
EM84    063555            3597    3602   11636#
EM85    063605            3130   11641#
EM86    063636            3658    3663    3669   11646#
EM87    063723            3345    3360   11655#
EM88    063750            3000   11659#
EM89    064041            3015   11669#
EM9     057652            2608   11271#
EM90    064144            3115   11681#
ERRVEC= 000004            1278#   3748    3749*   3760*   3769*   3770*   3783*   3784*   3852*   3856*   3863*   3901*   3919*
                          3920*  10034   10035*  10037*  10040*
ESEC    001404            2417#   5482*   5483*   5493   11909
E.A0    003412            2520#   4334*   4337*   4618*   4638*   4667*   4714*   4741*   4864*   4896*   4926*   5012*   5044*
                          5223*   5255*   5291*   5414*   5444*   5531*   5553*   5584*   5617*   5649*   5748*   5814*   5902*
                          5958*   6031*   6130*   6158*   6188*   6275*   6348*   6451*   6475*   6510*   6585*   6650*   6676*
                          6706*   6836*   6927*   6949*   6980*   7013*   7045*   7216*   7247*   7277*   7324*   7535*   7563*
                          7593*   7759*   7781*   7811*   7883*   7906*   7936*   8095*   8117*   8147*   8219*   8242*   8272*
                          8445*   8467*   8497*   8569*   8592*   8622*   8814*   9211*   9215*   9219    9221*   9250   11950
                         11961   11973
E.A1    003416            2522#   4339*   4620*   4640*   4669*   4716*   4743*   4866*   4898*   4928*   5014*   5046*   5225*
                          5257*   5297*   5416*   5446*   5533*   5555*   5586*   5619*   5651*   5750*   5816*   5904*   5960*
                          6033*   6132*   6160*   6190*   6277*   6350*   6453*   6477*   6512*   6587*   6652*   6678*   6708*
                          6838*   6929*   6951*   6982*   7015*   7047*   7218*   7249*   7279*   7326*   7537*   7565*   7595*
                          7761*   7783*   7813*   7885*   7908*   7938*   8097*   8119*   8149*   8221*   8244*   8274*   8447*
                          8469*   8499*   8571*   8594*   8624*   8816*   9212*   9223    9225*   9270   11950   11961   11973
E.A2    003422            2524#   4341*   4642*   4671*   4868*   4900*   4930*   4931*   4932*   4933*   4934*   4935    5016*
                          5048*   5227*   5259*   5299*   5418*   5448*   5557*   5621*   5653*   5752*   5818*   5906*   5962*
                          6035*   6162*   6192*   6279*   6352*   6479*   6514*   6589*   6680*   6710*   6840*   6953*   7017*
                          7049*   7251*   7281*   7328*   7567*   7597*   7785*   7815*   7910*   7940*   8121*   8151*   8246*
                          8276*   8471*   8501*   8596*   8626*   9213*   9227    9229*  11961   11973
```

```
E.A3    003426  2526# 9214*
E.B0    003414  2521# 4338* 4619* 4639* 4668* 4715* 4742* 4865* 4897* 4927* 5013* 5045* 5224*
                5256* 5294* 5296* 5415* 5445* 5532* 5554* 5585* 5618* 5650* 5749* 5815* 5903*
                5959* 6032* 6131* 6159* 6189* 6276* 6349* 6452* 6476* 6511* 6586* 6651* 6677*
                6707* 6837* 6928* 6950* 6981* 7014* 7046* 7217* 7248* 7278* 7325* 7536* 7564*
                7594* 7760* 7782* 7812* 7884* 7907* 7937* 8096* 8118* 8148* 8220* 8243* 8273*
                8446* 8468* 8498* 8570* 8593* 8623* 8815* 9231  9233* 9260  11950 11961 11973
E.B1    003420  2523# 4340* 4621* 4641* 4670* 4717* 4744* 4867* 4899* 4929* 5015* 5047* 5226*
                5258* 5298* 5417* 5447* 5534* 5556* 5587* 5620* 5652* 5751* 5817* 5905* 5961*
                6034* 6133* 6161* 6191* 6278* 6351* 6454* 6478* 6513* 6588* 6653* 6679* 6709*
                6839* 6930* 6952* 6983* 7016* 7048* 7219* 7250* 7280* 7327* 7538* 7566* 7596*
                7762* 7784* 7814* 7886* 7909* 7939* 8098* 8120* 8150* 8222* 8245* 8275* 8448*
                8470* 8500* 8572* 8595* 8625* 8817* 9235  9237* 9280  11950 11961 11973
E.B2    003424  2525# 4342* 4643* 4672* 4869* 4901* 4935* 4936* 5017* 5049* 5228* 5260* 5300*
                5419* 5449* 5558* 5622* 5654* 5753* 5819* 5907* 5963* 6036* 6163* 6193* 6280*
                6353* 6480* 6515* 6590* 6592* 6681* 6711* 6841* 6954* 7018* 7050* 7252* 7282*
                7329* 7568* 7598* 7786* 7816* 7911* 7941* 8122* 8152* 8247* 8277* 8472* 8502*
                8597* 8627* 9239  9241* 11961 11973
E.B3    003430  2527# 4343* 4644* 4673* 4870* 4902* 5018* 5050* 5229* 5261* 5303* 5307* 5309*
                5420* 5450* 5559* 5623* 5655* 5754* 5820* 5908* 5964* 6037* 6164* 6194* 6281*
                6354* 6481* 6516* 6591* 6682* 6712* 6842* 6955* 7019* 7051* 7253* 7283* 7330*
                7569* 7599* 7787* 7817* 7912* 7942* 8123* 8153* 8248* 8278* 8473* 8503* 8598*
                8628* 9243  9245* 11973
E.DDT   012114  4252* 4262* 4276# 9215
FATT1   044102  4634  4861  5009  5220  5372  5612  5682  7008  7111  8858  9141# 9868  9889
FATT2   044176  5407  6151  6265  6468  6574  6669  6752  7078  7240  7378  7441  7556  7676
                7774  7899  8016  8110  8235  8352  8414  8460  8585  8702  8772  9168#
FC      012116  4277# 5569  7738  8025  8074  8361  8681
FCP1    012120  4278# 5525  7739  8075
FHDHM   046374  5607  7003  9562# 9570
FHDTAB  046572  6016  6333  7159  7183  7304  9613#
FLGTST  047036  9660  9664  9680#
FLIM    046316  5575  6971  9544#
FLOAD   046450  5609  7005  9579#
FMTE  = 000020  1365#
FMT1    001440  2436# 9623* 9624* 9625* 9630
FNS22   045676  5480  9456#
FORM    031366  7121# 9903
FORMAT  001436  2435# 6015* 6332* 7158* 7188* 9623  9657
FRCYL   001350  2400# 4985* 5114  5115  5129  5131  5135  5173* 5174  5176  6121* 6437* 6633*
                6727* 7191* 7355* 7366* 7519* 7652* 7738* 7749  7877  7878  7951  7963  7989
                8074* 8085  8213  8214  8287  8299  8325  8424* 8435  8563  8564  8637  8649
                8675  8782* 11901 11925 11941
FRDY    043566  3987  4001  4010  4116  4125  4516  9047  9058  9070# 9073  9406
FRDY1   043634  9085# 9088  9393
FSPOK   046516  4307  4797  9595# 9603
FS022   045612  5476  9435#
FTITLE  001344  2396# 8941  8943*
GBA     043374  3819  9008#
GDRVS   043234  3817  8963#
GINT    043422  3821  9021#
GNS   = ****** U  1483  10960 10961 10962 10963 10964 10966 10968 10969 10970 10971 10972 10973
                10974
GO    = 000001  1329#
GSTAT   045132  4331  4353  4385  4418  4460  4574  4609  4689  4703  4730  4756  4772  4856
                4880  5004  5028  5215  5239  5320  5365  5574  5633  5740  5765  5808  5831
```

```
                    5885   5943   5974   6023   6072   6236   6340   6389   6446   6491   6547   6831   6970
                    7029   7171   7316   7469   7550   7641   7858   7983   8194   8319   8544   8669   8851
                    9151   9155   9175   9179   9340#  9410   9421   9475   9490   9547   9564   9581   9597
                    9884
GSTAT1  045166      9209   9352#
GSTAT2  045416      9357   9365   9373   9380   9389#
GTSWR = 104406      8952   10966#
HASOF   003340      2492#  9109*  9125   11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
HBA     003330      2488#  9105*  11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
HCS1    003322      2485#  4004*  4005*  4006   4012   4119*  4120*  4121   4127   4297*  4428*  4462   4537
                    4562   4581   4584   4605*  4659*  4787*  4851*  4888*  4921*  4999*  5036*  5077*  5098*
                    5123*  5159*  5210*  5247*  5286*  5350*  5356*  5402*  5409   5436*  5526*  5545*  5570*
                    5641*  5677*  5737*  5741   5792*  5795   5882*  5886   5893*  5927*  5930   5949*  6020*
                    6024   6057*  6060   6127*  6153   6180*  6220*  6223   6260*  6267   6337*  6341   6374*
                    6377   6441*  6470   6502*  6531*  6534   6569*  6576   6647*  6671   6698*  6747*  6754
                    6828*  6832   6922*  6941*  6966*  7037*  7073*  7080   7106*  7168*  7172   7213*  7242
                    7269*  7313*  7317   7373*  7380   7436*  7443   7453*  7456   7532*  7558   7585*  7625*
                    7628   7671*  7678   7756*  7776   7803*  7842*  7845   7880*  7901   7928*  7967*  7970
                    8011*  8018   8092*  8112   8139*  8178*  8181   8216*  8237   8264*  8303*  8306   8347*
                    8354   8409*  8416   8442*  8462   8489*  8528*  8531   8566*  8587   8614*  8653*  8656
                    8697*  8704   8767*  8774   8794*  8836*  8842*  9044*  9045   9055*  9056   9102*  9316
                    9320   9342*  9389*  9390*  9391   9411   9863*  9879*  9894*  11895  11903  11911  11919
                    11927  11935  11943  11954  11966  11979
HCS2    003324      2486#  4053   4055   4059   4071   4156   4158   4162   4165   4448   4465   9103*  9325
                    9329   11895  11903  11911  11919  11927  11935  11943  11954  11966  11979
HDA     003332      2489#  9106*  11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
HDB     003344      2494#
HDC     003342      2493#  9110*  11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
HDS     003334      2490#  4057   4160   4488   4496   4523   4531   9107*  11895  11903  11911  11919  11927
                    11935  11943  11954  11966  11979
HDTAB   001470      2456#  5727   5735*  6009   6084   6326   6401   7162   7308   9615
HDWD    001454      2444#  5849*  5850   5986*  5987   6086*  6088   6247*  6249   6403*  6405   11917
HD1     001434      2434#  9617*  9618*  9619*  9620*  9621*  9622*  9629
HEAD    001430      2432#  5718*  5729*  5730   5731*  5733   5774*  5775   5778*  5787*  5788   5789*  5939*
                    5840   5841*  5846   5857*  5858   6014*  6331*  7157*  7164*  7165   7166*  7179*  7180
                    7187*  7209*  7210   7211*  7305*  7306   7307*  7344*  7345   7351   7353*  7428*  7429
                    9617   9692
HEADA   001432      2433#  4361   5271   5334   9534*  9535*  9536*  9537*
HER     003336      2491#  4049   4151   4499   4534   9108*  11895  11903  11911  11919  11927  11935  11943
                    11954  11966  11979
HMR1    003346      2495#  9111*  11895  11903  11911  11919  11927  11935  11943  11954  11966  11979
HMR2    003350      2496#  4014   4129   4332   4386   4419   4432   4485   4493   4520   4528   4610   4690
                    4704   4731   4757   4773   4857   5005   5216   5578   5896   5952   6447   6974   8806
                    8809   9089*  9094*  9112*  9359   9367   9375   9382   9476   9565   9582   9598   9885
                    11895  11903  11911  11919  11927  11935  11943  11954  11966  11979
HMR3    003352      2497#  4559   4575   4578   5366   8800   8803   8852   9090*  9095*  9113*  9360   9368
                    9376   9383   9422   9491   9548   11895  11903  11911  11919  11927  11935  11943  11954
                    11966  11979
HOLD    043722      9074   9078   9102#  9353
HPAT    003356      2499#  9115*  11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
HPEN    050044      7402   7480   9921#
HPEND   003310      2467#  3846*  7357   7361*  9875*  9922
HPOS    003354      2498#  9114*  11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
HT    = 000011      1188#  10169  10227
HVRC  = 000400      1369#
HWC     003326      2487#  9104*  11898  11906  11914  11922  11930  11938  11946  11957  11969  11982
```

```
HZ      001370          2409#   4685    4699    4726    4752    9759
H.A0    003372          2509#   9250    9382*   11950   11964   11976
H.A1    003376          2511#   9270    9375*   11950   11964   11976
H.A2    003402          2513#   9367*   9501    11964   11976
H.A3    003406          2515#   9359*
H.B0    003374          2510#   9260    9383*   11950   11964   11976
H.B1    003400          2512#   5595    6991    9280    9376*   11950   11964   11976
H.B2    003404          2514#   9368*   9514    11964   11976
H.B3    003410          2516#   9360*   9524    9534    11976
IDAE  = 002000          1371#
IE    = 000100          1330#
ILF   = 000001          1361#
INTER   050214          9036    9980#
IOTVEC= 000020          1283#   3732*   3733*
IR    = 000100          1348#
LC      012100          4250*   4260*   4270#   6725    6826    6965    7087    7347    7519    7685    8331    8407    8424
                        8711
LCLKF   003464          2558#   3854*
LCM1    012076          4249*   4259*   4269#   6921    7520    8425
LCP1    012102          4251*   4261*   4271#   5169    5174    8783
LCVEC   001330          2386#   3855
LF    = 000012          1189#   10221   10227   10982   10986   10988   10992   11002   11011   11018   11026   11031   11036
                        11041   11047   11054   11058   11065   11070   11079   11084   11088   11092   11095   11100   11106
                        11110   11114   11120   11125   11129   11134   11138   11140   11143   11147   11152   11156   11166
                        11174   11183   11188   11194   11198   11201   11204   11211   11229   11236   11239   11457   11527
                        11659   11663   11669   11676
LIMERR  001460          2447#   5711    7140    9544*   9552*
LKS     001326          2384#   3853    9748*   9769*
LPFLG   001410          2420#   4804    4812*   4817*   5520*   5686*   6295    6301*   6306*   6605    6610*   6615*   6770
                        6778*   6784*   6916*   7096    7115*   7396    7410    7416*   7422*   7488*   7493*   7694    7702*
                        7708*   8034    8040*   8045*   8050*   8055*   8370    8376*   8381*   8386*   8391*   8720    8726*
                        8731*   8736*   8741*   9828*
MASK    012110          4255*   4265*   4274#   8863    9482    9507
MASK1   012112          4256*   4266*   4275#   9477    9492    9502    9515
MC      012104          4253*   4263*   4272#   7995
MCLK  = 000400          1401#
MC1     012106          4254*   4264*   4273#   4956
MDS   = 001000          1351#   4053    4156    9329
MDTST   007056          1497    3690#
MDTSTA  007076          1499    3695#
MEMBAS= 172100          2390#   3772    3781
MEMERR  050152          3775    9966#
MEMPAR  007444          3769#
MEMVEC= 000114          2389#   3775*   3776*
MERD  = 001000          1402#
MESMSK= 000017          1396#
MEWD  = 002000          1403#
MIND  = 000200          1400#
MODTST  001342          2395#   3677*   3682*   3687*   3692*   3697*   3701*   7136    7203    7232    7340    7515    8069
MSG1    054640          8944    10982#
MSG10   056133          3952    3991    11110#
MSG11   056162          10007   11114#
MSG12   056222          7152    11120#
MSG13   056253          3877    11125#
MSG14   056331          9197    11134#
MSG15   056354          4073    4237    11138#
```

```
MSG16   056366            4383   11140#
MSG17   056407            6890   11143#
MSG18   056436            5744    5798    5810    5889    5933    5945    6027    6063    6074    6226    6238    6344    6380
                          6391    6537    6549    7175    7320    7459    7471    7631    7643    7848    7860    7973    7985
                          8184    8196    8309    8321    8534    8546    8659    8671   11147#
MSG19   056472            4215   11152#
MSG2    055503            3800   11058#
MSG20   056522            7138   11156#
MSG21   056671            7517   11174#
MSG22   056751            7514   11183#
MSG23   057001            4981   11188#
MSG3    055667            3816   11079#
MSG4    055721            3818   11084#
MSG5    055747            3820   11088#
MSG6    055774            9982   11092#
MSG7    056012            3835   11095#
MSG74   057041            9855   11194#
MSG75   057070            9857   11198#
MSG76   057112            9907   11201#
MSG77   057130            9912   11204#
MSG8    056047            4983   11100#
MSG9    056105            4976   11106#
MSP   = 000100            1399#
MULT6   047222            9710    9736#
M.ALGN= 040000            1472#
M.DRV = 000007            1466#
M.HEAD= 007000            1474#   9535
M.ID  = 000003            1471#
M.PAR = 100000            1475#   9798
M.SECT= 000760            1473#   9423    9525
M.SER = 077770            1467#
NED   = 010000            1354#   4059    4071    4162    4165    9325
NEM   = 004000            1353#
NUDRV   011544            4085    4194#   8886
NXF   = 000004            1363#
OFFSET= 000015            1321#
OFST  = 000004            1382#
OPI   = 020000            1374#
OR    = 000200            1349#
O.ADR1  070136           12275*  12326*  12328*  12498   12499*  12537*  12552   12569   12760#
O.BACK  070662           12370   12411#
O.BD    072154           12556   12557   12705#
O.BKPT  070370           12321#  12374
O.BK1   071322           12269   12534#
O.BK2   071356           12539   12542#
O.BK3   071374           12541   12548#
O.BRK   071312           12286   12532#
O.BUF   072203           12614   12623   12628   12733#
O.B2    071456           12553   12562#
O.B3    071516           12561   12570#
O.CADV  071666           12387   12404   12427   12431   12461   12465   12570   12613#
O.CLGL  070504           12344   12346   12354#
O.CLGT= 000014           12358   12723#
O.CLS1  072112           12681   12683#
O.CR    072156           12690   12692   12693   12707#
O.CRET  070616           12365   12394#
```

```
O.CRLF  072122          12402   12459   12555   12594   12599   12690#
O.CRLS  072130          12339   12692#
O.CRS   072134          12691   12693#
O.CSR1  072152          12590*  12606   12701#
O.CSR2  072153          12591*  12607   12702#
O.CT    070140          12519*  12562*  12564*  12761#
O.C1    071270          12522#  12563
O.DCD   070434          12327   12329   12339#  12396   12447   12572
O.DCD1  070440          12340#  12388   12432
O.EFF   070744          12373   12438#
O.EFF1  071062          12450   12469#
O.ENTR  070150          12266#
O.ERR   070420          12313   12324   12336#  12359   12384   12442   12685
O.ERR1  070754          12442#  12515
O.ERR2  070574          12384#  12418   12491
O.FTYP  072030          12338   12406   12421   12463   12568   12639   12655   12660#
O.GET   071752          12292   12342   12636#  12641   12643   12645
O.GO    071126          12367   12489#
O.GO2   071204          12500#  12525
O.LG  = 000006          12296   12731#
O.LGCH  072161          12355   12711#  12723
O.LGDR  070534          12362   12364#  12376
O.LGL = 000030          12376#
O.LGL1  070506          12355#  12360
O.LGL2  070526          12356   12361#
O.MSK   070130          12443   12444   12446   12753#
O.ODT   070144           1495   12264#  12742
O.OFST  070672          12371   12417#
O.OF1   070740          12430   12432#
O.OP1   070630          12368   12400#
O.OP2   070636          12317   12402#  12413
O.ORPC  070342          12309#  12369
O.P     072151          12284*  12489*  12514    2516*  12565*  12699#
O.PRI   070126          12276   12538   12542   12752#
O.PROC  071234          12375   12514#
O.PR1   071256          12518   12520#
O.RALL  070410          12287   12322   12328#
O.RCSR= 177560          12254#  12590   12592*  12602   12604   12606*  12636
O.RDB = 177562          12253#  12638
O.REGT  070272          12292#  12366
O.RSE1  071650          12603   12606#
O.RSP   070302          12294#  12297
O.RST   070206          12265   12274#
O.RSTT  071620          12495   12521   12599#
O.RST1  070236          12273   12281#
O.RTIT  071232          12283*  12509#
O.SCAN  070444          12303   12342#  12353
O.SP    070334          12295   12304#
O.SPC   071700          12616#  12624
O.SP1   070322          12300#  12305
O.STM = 000340          12243#  12285   12494   12520   12522
O.STRT  070174          12264   12271#
O.SVR   071530          12274   12534   12576#
O.SVTT  071566          12554   12566   12590#
O.T     072150          12496*  12523*  12535   12698#
O.TBIT  071156          12496#  12536
```

```
O.TBT = 000020          12244#  12497   12524   12560
O.TC    072114          12678   12684#
O.TCLS  072076          12309   12394   12400   12411   12677#
O.TCSR= 177564          12256#  12591   12593*  12600   12607*  12660   12666
O.TDB = 177566          12255#  12662*  12668*
O.TL    072175          12293   12296   12304   12725#  12731
O.TRTC  072212          12328   12499   12738#
O.TVEC= 000014          12242#  12267   12285*  12286*
O.TYPE  072014          12558   12629   12652#  12656   12694
O.TYP1  072074          12653   12664   12672#
O.TYP2  072054          12666#  12667   12670
O.UIN   070142          12275   12498*  12537   12762#
O.UPC   070122          12268*  12493*  12508   12532*  12549   12551*  12750#
O.URO   070104          12271   12301   12743#
O.USP   070120          12272*  12577*  12578   12749#
O.UST   070124          12266*  12497*  12507   12524*  12533*  12540   12560*  12751#
O.WDS   070752          12439   12441#
O.WDS2  070770          12446#  12468
O.WDS3  071020          12457#  12470   12476   12485
O.WDS4  071056          12457   12467#
O.WRD   070564          12364   12380#
O.WRDA  070610          12381   12389#
O.WRD1  070572          12383#  12390   12407
O.WSCH  070750          12372   12440#
O.XXX   072146          12576*  12584   12697#
O.45    070250          12282   12284#
PACK  = 000003           1316#   4428
PARAM   001336           2393#   3675*   3680*   3685*   3690*   3695*   3699*   3813
PARSRT  006774           1491    3675#
PAT   = 000020           1397#   4573    4919    5067    5113    5524    6920
PCA   = 004000           1404#
PCD   = 010000           1405#
PCLKF   003466           2559#   3858*   3865*   9746    9767
PCVEC   001332           2387#   3859    3866
PCYL    001356           2403#   4988*   5071    5076*   5117    5122*
PFSRT   012214           4314#  10009
PGE   = 002000           1352#
PIP   = 020000           1390#
PIRQ  = 177772           1195#
PIRQVE= 000240           1289#
PKRB    001324           2383#
PKS     001320           2381#   3857    3864    9751*   9771*
PKSB    001322           2382#   9750*
PPTP    003440           2543#   3804*
PRGSRT  007134           3678    3683    3688    3693    3698    3702#   8995
PR0   = 000000           1212#   3704    3767    3768    3896
PR1   = 000040           1213#
PR2   = 000100           1214#
PR3   = 000140           1215#
PR4   = 000200           1216#
PR5   = 000240           1217#   2380
PR6   = 000300           1218#   3869
PR7   = 000340           1219#   3770    3776    3920    10004   10005
PS    = 177776           1192#   1193
PSEC    001402           2416#   5479*   5482    5484*   9459    9462    11909
PSW   = 177776           1193#
```

```
PWRVEC= 000024  1284#  3738*  3739*  9994*  10003*  10004*
RCYLA   046174  9296   9514#
RCYLD   046122  9292   9501#
RDCHR = 104410  10634  10969#
RDCYLA  046052  5084   5130   5391   9489#
RDCYLD  045766  5089   5139   5145   515C   9474#
RDDATA= 000021  1323#  6828
RDGATE= 100000  1408#
RDHEAD= 000025  1325#  5792   5927   6057   6220   6374   6531   7453   7625   7842   7967   8178   8303
                8528   8653
RDLIN = 104411  8963   10708  10970#
RDOCT = 104412  9008   9021   10971#
RDSEC   045542  9420#  9437   9440   9458   9461   9708
RDY   = 000200  1331#  9070   9085
RECAL = 000013  1320#  4851   4999   5210   5350   6441   8836   9879
RESREG= 104414  10768  10838  10973#  12212
RESVEC= 000010  1279#
RHEAD   046270  9301   9534#
RHTAB   001674  2457#  5791   5843   5926   5982   6055   6077   6219   6243   6247   6372   6394   6530
                6553   6557   7452   7475   7624   7647   7841   7864   7966   7989   8177   8200   8302
                8325   8527   8550   8652   8675   9715   9717   9721   9724   11933  11941
RKASOF= 000016  1302#  3910   9109   9144   9170
RKBA  = 000004  1297#  3906   5727*  5878*  6009*  6326*  6822*  6864*  6883*  7162*  7308*  9105
RKCS1 = 000000  1295#  3903   4006*  4121*  4512*  4514*  4632*  4657*  4684*  4697*  4770*  4849*  4886*
                4997*  5034*  5082*  5096*  5128*  5157*  5208*  5245*  5348*  5354*  5370*  5404*  5543*
                5573*  5639*  6178*  6444*  6500*  6696*  6939*  6969*  7035*  7267*  7583*  7801*  7926*
                8137*  8262*  8487*  8612*  8834*  8840*  8856*  9045*  9056*  9070   9085   9102   9354*
                9362*  9370*  9378*  9391*  9877*  9892*
RKCS2 = 000010  1299#  3904   3985*  3999*  4003*  4114*  4118*  4459*  4658*  4850*  4887*  4998*  5035*
                5097*  5158*  5209*  5246*  5349*  5355*  5435*  5544*  5640*  5726*  5806   5877*  5941
                6070   6179*  6234   6387   6501*  6545   6697*  6940*  7036*  7268*  7467   7584*  7639
                7802*  7856   7927*  7981   8138*  8192   8263*  8317   8488*  8542   8613*  8667   8835*
                8841*  9103   9341*  9355*  9363*  9371*  9379*  9404*  9408*  9878*  9893*
RKDA  = 000006  1298#  3907   5283*  5730*  5788*  5840*  6824*  6866*  6885*  7165*  7210*  7306*  7'33*
                9106
RKDB  = 000024  1304#  3912   5801   5802   5803   5936   5937   5938   6066   6067   6068   6229   6230
                6231   6383   6384   6385   6540   6541   6542   7462   7463   7464   7634   7635   7636
                7851   7852   7853   7976   7977   7978   8187   8188   8189   8312   8313   8314   8537
                8538   8539   8662   8663   8664
RKDC  = 000020  1303#  3911   4920*  5068*  5083*  5114*  5129*  5525*  5569*  6018*  6053*  6124*  6217*
                6335*  6370*  6567*  6645*  6745*  6826*  6921*  6965*  7071*  7208*  7310*  7371*  7530*
                7621*  7669*  7754*  7838*  7877*  7963*  8009*  8090*  8174*  8213*  8299*  8345*  8407*
                8440*  8524*  8563*  8649*  8695*  8793*  9110
RKDS  = 000012  1300#  3908   9107
RKECPS= 000030  1308#  3916   9114
RKECPT= 000032  1309#  3917   9115
RKER  = 000014  1301#  3909   9108
RKMR1 = 000026  1305#  3913   4352*  4384*  4417*  4573*  4688*  4702*  4729*  4755*  4771*  4855*  4879*
                4919*  5003*  5027*  5067*  5113*  5214*  5238*  5319*  5524*  5632*  5764*  5830*  5973*
                6445*  6490*  6920*  7028*  7549*  9111   9150*  9154*  9174*  9178*  9356*  9364*  9372*
                9409*  9420*  9474*  9489*  9546*  9563*  9580*  9596*  9883*
RKMR2 = 000034  1306#  3914   9089   9094   9112
RKMR3 = 000036  1307#  3915   9090   9095   9113
RKPRI   001316  2380#  9037
RKVEC   001314  2379#  3840*  9025*  9028*  9035
RKWC  = 000002  1296#  3905   5728*  5879*  6010*  6327*  6827*  7163*  7309*  9104
```

```
RLS    = 000010          1345#   4459
RSEC     046232          9300    9524#
RTT    = 000006         12246#
SAVREG= 104413          10753   10812   10972#  12156
SBPAR    047362          9220    9224    9228    9232    9236    9240    9244    9784#
SCLR   = 000040          1347#   3985    3999    4114    9404
SCOP1  = 104415          3996    4111    4422    4453    4504    4567    4842    4912    4990    5060    5107    5200    5276
                         5396    5720    5779    5919    6047    6210    6364    6638    7197    7297    7523    7613    7742
                         7831    7870    7956    8078    8167    8206    8292    8428    8517    8556    8642    8786   10974#
SCOP1$   047434          9808#  10974
SDC    = ****** U        1175    1494   12222
SEC      001374          2413#   4686*   4700*   4727*   4753*   9760*
SECFLG   046766          9635    9656#
SECNT    001400          2415#   5474*   5496*   5917*   6046*   6363*
SECTOR   001406          2418#   5484    5493    9422*   9423*   9424*   9425*   9426*   9427*   9438    9441    9459    9462
                         9524*   9525*   9526*   9527*   9528*   9529*   9709*   9713    9714*   9715*   9726    9736*   9737
                         9738*   9739*
SEEK   = 000017          1322#   4921    5077    5123    5286    5402    5526    5570    6127    6260    6569    6647    6747
                         6922    6966    7073    7213    7373    7436    7532    7671    7756    7880    8011    8092    8216
                         8347    8409    8442    8566    8697    8767    8794
SELDRV= 000001          1315#   4004    4119    4512    4514    5893    5949    9342    9389
SETINT   043454          3841    9026    9035#
SIZFLG   003472          2561#   3842*   3989    8981*   9001*
SKI    = 000002          1362#
SORT     047114          6080    6397    9706#
SPACE2   070060         12190   12199   12210   12221#
SRTSPL= 000011          1319#   4605    5677    7106    9863
SRTTAB   002100          2458#   6083    6400    9718
ST     = 177776         12240#  12266   12267*  12280*  12494*  12520*  12522*  12548*
STACK  = 001100          1183#   3703    3730    3894    3945    3997    4107    4112    4209    4290    4326    4376    4415
                         4423    4444    4454    4479    4505    4555    4568    4600    4836    4843    4913    4972    4991
                         5061    5108    5196    5201    5277    5387    5397    5468    5515    5710    5721    5780    5873
                         5920    6002    6048    6118    6211    6319    6365    6433    6629    6639    6813    6911    7135
                         7198    7298    7512    7524    7614    7736    7743    7832    7871    7957    8067    8079    8168
                         8207    8293    8401    8429    8518    8557    8643    8762    8787    8881    9306    9975   10006
                        12218
START    007120          1487    3699#
START1   007552          3794#
STKLMT= 177774          1194#
STOP     047536          7363    9846#  10475   10558
ST2      007622          3802    3813#
ST3      007666          3803    3831#
ST4      007720          3815    3836    3839#
ST5      007744          3822    3844#   3983    4047    8924    9914
ST5XY    043102          8913    8918#
SUBCLR   045462          4294    4303    4328    4380    4425    4446    4456    4481    4507    4557    4570    4602    4784
                         4793    4800    4845    4915    4993    5063    5110    5203    5279    5389    5399    5470    5517
                         5604    5674    5723    5782    5875    5922    6004    6050    6119    6213    6257    6271    6321
                         6367    6435    6525    6565    6580    6631    6641    6742    6758    6766    6815    6851    6913
                         7000    7068    7084    7092    7103    7149    7200    7300    7368    7384    7392    7447    7526
                         7616    7666    7682    7690    7745    7834    7873    7959    8006    8022    8030    8081    8170
                         8209    8295    8342    8358    8366    8404    8420    8431    8520    8559    8645    8692    8708
                         8716    8764    8778    8789    9404#   9848    9860
SVAL   = 100000          1392#
SWR      001140          2297#   3728    3750*   3752    3758*   3765*   4813    4818    5687    6302    6307    6611    6616
                         6779    6785    7116    7417    7423    7489    7494    7703    7709    8041    8046    8051    8056
```

```
                        8377   8382   8387   8392   8727   8732   8737   8742   8950   9256   9266   9276   9286
                        9808   9829   9834   9836   9938   9946   9951   9971  10029  10043  10045  10051  10058
                       10096  10103  10115  10119  10478  10517  10572* 12214
SWREG   000176          1485#  3758   8950  10478  10517  10540
SWTST   047466          4778   5664   6253   6561   6733   7060   7405   7483   7658   8000   8336   8686   9827#
SW0   = 000001          1247#
SW00  = 000001          1237#  1247
SW01  = 000002          1236#  1246
SW02  = 000004          1235#  1245
SW03  = 000010          1234#  1244
SW04  = 000020          1233#  1243
SW05  = 000040          1232#  1242
SW06  = 000100          1231#  1241
SW07  = 000200          1230#  1240
SW08  = 000400          1229#  1239
SW09  = 001000          1228#  1238
SW1   = 000002          1246#
SW10  = 002000          1227#
SW11  = 004000          1226#
SW12  = 010000          1225#  12214
SW13  = 020000          1224#   9938
SW14  = 040000          1223#   9829   9951
SW15  = 100000          1222#
SW2   = 000004          1245#
SW3   = 000010          1244#
SW4   = 000020          1243#
SW5   = 000040          1242#
SW6   = 000100          1241#
SW7   = 000200          1240#
SW8   = 000400          1239#  9834
SW9   = 001000          1238#  4813   4818   5687   6302   6307   6611   6616   6779   6785   7116   7417   7423
                        7489   7494   7703   7709   8041   8046   8051   8056   8377   8382   8387   8392   8727
                        8732   8737   8742   9256   9266   9276   9286   9808   9946   9971
TBITVE= 000014          1280#
TEMP1   003360          2501#  3986*  4000*  4007*  4009*  4014*  4015*  4016   4115*  4122*  4124*  4129*  4130*
                        4131   4515*  5332*  5333*  5334   5405*  5848*  5850   5985*  5987   6087*  6088   6148*
                        6248*  6249   6264*  6404*  6405   6467*  6573*  6667*  6751*  7077*  7238*  7377*  7440*
                        7554*  7675*  7772*  7897*  8015*  8108*  8233*  8351*  8413*  8458*  8583*  8701*  8771*
                        9046*  9057*  9072*  9087*  9142*  9146*  9172*  9188   9190*  9217   9219*  9221   9223*
                        9225   9227*  9229   9231*  9233   9235*  9237   9239*  9241   9243*  9245   9247*  9340
                        9345*  9352   9385*  9392*  9405*  9435   9436*  9443*  9445*  9447*  9456   9457*  9464*
                        9466*  9468*  9545*  9550*  9562*  9567*  9579*  9584*  9595*  9600*  9637*  9640*  9641
                        9790*  9798*  11917
TEMP2   003362          2502#  4306*  4633*  4796*  4860*  5008*  5219*  5371*  5606*  5611*  5681*  6817*  6854*
                        6855   6881*  7002*  7007*  7110*  8857*  9148*  9569*  9602*  9638*  9639*  9641*  9642
                        9867*  9888*
TEMP3   003364          2503#  5070*  5072   5116*  5118   6818*  6857   6878*  6879   7749*  7751   8085*  8087
                        8435*  8437*  8438
TEMP4   003366          2504#  5071*  5072*  5074*  5075   5117*  5118*  5120*  5121   6821*  6822   6864   6882*
                        6883   7750*  7751*  7752   8086*  8087*  8088   8436*  8437
TEMP5   003370          2505#  4839*  4954*  6823*  6824   6865*  6866   6884*  6885
TIMUP   001376          2414#  4692   4706   4733   4759   9745*  9756*  9762*
TITLE   043154          3796   3834   8941#
TKVEC = 000060          1287#  10450* 10451*
TOCYL   001352          2401#  4986*  5068   5069   5083   5085   5135   5137   5168*  5169   5176*  6011*  6013
                        6122*  6328*  6330   6438*  6634*  6645   6721   6725   6728*  6745   6761   6763*  7071
```

```
                        7087    7089*   7154*   7156    7192*   7356*   7371    7387    7389*   7434*   7475    7520*   7530
                        7608    7621    7647    7654*   7669    7685    7687*   7739*   7750    7754    7826    7838    7864
                        7995    7997*   8009    8025    8027*   8075*   8086    8090    8162    8174    8200    8331    8333*
                        8345    8361    8363*   8425*   8436    8440    8512    8524    8550    8681    8683*   8695    8711
                        8713*   8783*   8792    8793    8825    8828    8862*   8863    11901   11925   11933
TPVEC = 000064           1288#
TRAPPC  001334           2391#   9967*   11949
TRAPVE= 000034           1286#   3736*   3737*   3768*   10005*
TRT   = 000003          12245#  12738
TRTVEC= 000014           1281#
TSTATN  044050           4300    4615    4662    4711    4738    4764    4790    4891    4924    5039    5080    5101    5126
                         5162    5250    5289    5359    5439    5529    5548    5600    5644    6183    6505    6701    6925
                         6944    6996    7040    7272    7588    7806    7931    8142    8267    8492    8617    8797    8845
                         9123#   9897
TST1    010134           3870    3892#
TST10   012550           4379    4413#
TST11   012646           4420    4433    4442#
TST12   012754           4466    4477#
TS113   013250           4538    4553#
TST14   013414           4585    4598#
TST15   014736           4779    4834#
TST16   015550           4958    4970#
TST17   017012           5175    5194#
TST2    010316           3921    3943#
TST20   020032           5385#
TST21   020376           5460    5466#
TST22   020522           5513#
TST23   021602           5665    5708#
TST24   022522           5859    5871#
TST25   023302           6000#
TST26   023756           6116#
TST27   025052           6254    6317#
TST3    011152           4105#
TST30   025526           6431#
TST31   026626           6562    6627#
TST32   027526           6734    6811#
TST33   030164           6870    6889    6909#
TST34   031366           7061    7131#
TST35   033436           7406    7484    7510#
TST36   034514           7659    7734#
TST37   036366           8001    8065#
TST4    011554           4141    4207#
TST40   040252           8337    8399#
TST41   042210           8687    8760#
TST5    012122           4257    4267    4288#
TST6    012214           4324#
TST7    012412           4362    4374#
TYPDS = 104405           8902   10964#
TYPE  = 104401           3800    3816    3818    3820    3835    3877    3952    3962    3991    4027    4073    4215    4237
                         4383    4404    4405    4976    4981    4983    5597    5744    5798    5810    5889    5933    5945
                         6027    6063    6074    6226    6238    6344    6380    6391    6537    6549    6890    6894    6895
                         6993    7138    7152    7175    7320    7459    7471    7514    7517    7631    7643    7848    7860
                         7973    7985    8184    8196    8309    8321    8534    8546    8659    8671    8900    8903    8944
                         8994    9197    9855    9857    9907    9912    9940    9982   10007   10098   10106   10174   10285
                        10411   10472   10484   10538   10539   10542   10553   10563   10574   10593   10641   10647   10652
                        10656   10661   10662   10664   10667   10671   10737   10739   10879   10960#  12166   12167   12171
```

```
                    12172  12181  12190  12194  12199  12201  12207  12210
TYPERR  067604      10105  12156#
TYPOC = 104402      10541  10961#  12187
TYPON = 104404      10963#
TYPOS = 104403       3965   4030   4076   4241   9200   9943   9985  10962#
T.A2  = 000001       2532#  4346   4647   4676   4873   4905   4939   5021   5053   5232   5264   5313   5423
                     5453   5562   5626   5658   5757   5823   5967   6040   6136   6167   6197   6284   6357
                     6457   6484   6519   6595   6656   6685   6715   6958   7022   7054   7222   7256   7286
                     7333   7541   7572   7602   7765   7790   7820   7889   7915   7945   8101   8126   8156
                     8225   8251   8281   8451   8476   8506   8575   8601   8631   8820   9290
T.B2  = 000002       2533#  4346   4647   4676   4873   4905   4939   5021   5053   5232   5264   5313   5423
                     5453   5562   5626   5658   5757   5823   5967   6040   6136   6167   6197   6284   6357
                     6484   6519   6595   6656   6685   6715   6958   7022   7054   7222   7256   7286   7333
                     7541   7572   7602   7765   7790   7820   7889   7915   7945   8101   8126   8156   8225
                     8251   8281   8451   8476   8506   8575   8601   8631   8820   9294
T.B3  = 000004       2534#  4346   4873   5021   5232   5313   9298
T1      001412       2424#  5371   8857
T10     001414       2425#  3986   4000   4009   4115   4124   4306   4515   4796   4860   5008   5219   5606
                     7002   9046   9392   9405   9888
T100    001416       2426#  4633   5611   7007
T2500   001422       2428#  6148   6667
T500    001420       2427#  5681   7110   9457   9867
T5000   001424       2429#  9436
T50000  001426       2430#  4007   4122   6264   6573   6751   7077   7377   7440   7675   7772   7897   8015
                     8108   8233   8351   8413   8458   8583   8701   8771   9057
UFE   = 000400       1350#  4055   4158   4448   4465
UNLD    003304       2463#  3844*  4292*  4777*  4782*  5522*  5615*  5685*  6918*  7011*  7114*  9851   9871*
UNLOAD= 000007       1318#  4297   4787
UNS   = 040000       1375#
UPE   = 020000       1355#
VERIFY  011152       3979   3984   4087#
VV    = 000100       1386#
WCE   = 040000       1356#
WDCNT   001442       2437#  5845*  5853*  5854   5984*  5990*  5991   6082*  6092*  6093   6399*  6409*  6410
                    11917
WLE   = 004000       1372#
WORD    001452       2443#
WRDATA= 000023       1324#
WRHEAD= 000027       1326#  5737   5882   6020   6337   7168   7313
WRL   = 004000       1389#
WRTCHK= 000031       1327#
WRTGAT= 040000       1407#
$APTHD  001000       1521   1527#
$ASTAT= ****** U    10328  10343
$ATYC   051606      10299  10301#
$ATY1   051562      10297#
$ATY3   051570      10159  10298#
$ATY4   051600      10111  10300#
$AUTOB  001134       2294#  8954*  10531  10688
$BASE   001264       2364#  3839*   3902   9012*  9014*
$BDADR  001122       2289#
$BDDAT  001126       2291#
$BELL   001200       2315#  10098  10131  10484  10681
$CDW1   001270       2366#
$CDW2   001272       2367#
$CHARC  051332      10176*  10186*  10193  10219*  10224#
```

```
$CKSWR  052566          10517#  10968
$CMTAG  001100           2277#   3725    3726    3734    3740    3741    3742
$CM3  = 000000           2307#
$CM4  = 000006           2307#   2308#   2309#   2310#   2311#   2312#   2313#
$CNTLC  053476          10472   10553   10681#
$CNTLG  053510          10538   10683#
$CNTLU  053503          10563   10656   10682#
$CPUOP  001236           2338#
$CRLF   001205           2317#   3962    4027    4404    4405    6894    6895   10106   10131   10175   10227   10574   10661
                        10681   10742   12166   12171   12181   12194   12207
$DBLK   051552          10251   10285   10293#
$DB2D   054054          10794   10812#
$DB2O   053700           6892   10753#
$DDW0   001274           2368#
$DDW1   001276           2369#
$DDW2   001300           2370#
$DDW3   001302           2371#
$DDW4   001304           2372#
$DDW5   001306           2373#
$DDW6   001310           2374#
$DDW7   001312           2375#
$DECVL  054234          10814   10860#
$DEVCT  001220           2329#   3787*   3850*   4219    8882*   8883
$DEVM   001266           2365#   3956
$DOAGN  043056           8896    8905    8911#
$DTBL   051542          10254   10289#
$ENDAD  043046           1507    8907#  10126
$ENDCT  043014           3740    8898#
$ENDMG  043065           8900    8915#
$ENULL  043062           8903    8914#
$ENV    001230           2334#   8918    8948   10108   10154   10306   10330
$ENVM   001231           2335#   3763    3948   10156   10161   10308
$EOP    042706           4776    5714    5745    5799    5811    5890    5934    5946    6028    6064    6075    6227    6239
                         6345    6381    6392    6538    6550    7143    7147    7176    7321    7460    7472    7632    7644
                         7849    7861    7974    7986    8185    8197    8310    8322    8535    8547    8660    8672    8864
                         8876#  12219
$EOPCT  043006           3740*   8895#   8899    9908*
$EOP1   042760           3925    4216    8888    8889#   9910    9962
$ERFLG  001103           2280#   9810   10018   10047   10049   10055*  10077   10093*  10131   12216
$ERMAX  001115           2286#   3743*  10049   10072*  10077
$ERROR  050602           3734   10091#
$ERRPC  001116           2287#  10100*  10101*  10102   10131   11894   11901   11909   11917   11925   11933   11941   11950
                        11961   11973
$ERRTB  003474           2576#  12163
$ERTTL  001112           2284#  10099*  10131
$ESCAP  001176           2314#   3742*   3847*   4614*   4631*   4766*   4769*   4803*   5472*   5500*   5583*   5614*   5670*
                         6125*   6147*   6216*   6294*   6440*   6466*   6528*   6604*   6644*   6666*   6769*   6979*   7010*
                         7095*   7205*   7207*   7234*   7237*   7303*   7395*   7409*   7529*   7553*   7619*   7693*   7748*
                         7771*   7837*   7876*   7896*   7962*   8033*   8084*   8107*   8173*   8212*   8232*   8298*   8369*
                         8434*   8457*   8523*   8562*   8582*   8648*   8719*   8879*   9248    9252*   9262*   9272*   9282*
                         9303*   9307*   9827*   9909*   9966*   9970*  10071*  10122   10124   10131
$ETABL  001230           2333#
$ETEND  001314           1533    2376#
$FATAL  001212           2326#  10334*
$FFLG   052026          10297*  10300*  10328   10337*  10345#
$FILLC  001156           2305#  10179   10227
```

```
$FILLS  001155          2304#  10227
$GDADR  001120          2288#
$GDDAT  001124          2290#
$GET42  043036          8904#
$GTSWR  052656         10539#  10966
$HD    = 000000          1152
$HIBTS  001000          1528#
$HIOCT  053676         10730*  10741#
$ICNT   001104          2281#  10062*  10063   10065*  10076
$INTAG  001135          2295#  10536*  10555   10575   10688
$ITEMB  001114          2285#  10102*  10110   10131   12157
$LF     001206          2318#  10131   10227   10671   10681   10742
$LFLG   052025         10338*  10344#
$LPADR  001106          2282#   3744*   4807    5671    6298    6608    6773    7099    7399    7413    7697    8037    8373
                        8723    9949   10053*  10069*  10074   10076
$LPERR  001110          2283#   3745*   4806    6297    6607    6772    7098    7398    7412    7696    8036    8372    8722
                        9308    9812    9815*   9976   10053   10070*  10076   10121
$MADR1  001242          2351#
$MADR2  001246          2355#
$MADR3  001252          2358#
$MADR4  001256          2361#
$MAIL   001210          1529    1533    2324#   3762    8948   10068   10108   10154
$MAMS1  001240          2345#
$MAMS2  001244          2353#
$MAMS3  001250          2356#
$MAMS4  001254          2359#
$MBADR  001002          1529#
$MFLG   052024         10298*  10304   10339*  10343#
$MNEW   053526         10542   10686#
$MSGAD  001224          2331#  10314*  10317
$MSGLG  001226          2332#  10319*
$MSGTY  001210          2325#  10312   10320*  10332   10336*
$MSWR   053515         10539   10684#
$MTYP1  001241          2346#
$MTYP2  001245          2354#
$MTYP3  001251          2357#
$MTYP4  001255          2360#
$MXCNT  050600         10066   10076#
$NULL   001154          2303#  10181   10227
$NWTST= 000001          3882#   3884    3928#   3930    4090#   4092    4198#   4200    4279#   4281    4317#   4319    4367#
                        4369    4407#   4409    4436#   4438    4469#   4471    4541#   4543    4588#   4590    4823#   4825
                        4961#   4963    5181#   5183    5376#   5378    5463#   5502#   5504    5696#   5698    5862#   5864
                        5997#   6102#   6104    6314#   6418#   6420    6619#   6621    6789#   6791    6897#   6899    7122#
                        7124    7503#   7505    7716#   7718    8062#   8396#   8749#   8751
$OCNT   052252         10383*  10412*  10425#
$OCTVL  054002          4387   10755   10780#
$OMODE  052254         10378*  10382*  10387   10390*  10401*  10427#
$OVER   050564         10030   10046   10054   10064   10073#
$PASS   001216          2328#   3762*   4378    6888    8892*   8893*   8901    8914    8920   10060   10077
$PASTM  001006          1531#
$PWRCT  050320         10000*  10001*  10011#
$PWRDN  050234          3738    9994#  10003
$PWRUP  050246          9994   10000#
$QUES   001204          2316#  10131   10227   10593   10664   10681   10739   10742
$RDCHR  053130         10606#  10969
$RDDEC= ****** U       10972
```

```
$RDLIN  053220          10629#  10970
$RDOCT  053540          10703#  10971
$RDSZ = 000022          10622#
$RESRE  054346          10917#  10973
$RTNAD  043060           8913#
$R2A  = ****** U        10974
$SAVRE  054310          10901#  10972
$SB2D   054020          10792#
$SCOPE  050322           3732   10027#
$SETUP= 000137           3672#   3731    3732    3734    3736    3738    3740    3741    3742    3744    8890    8945    10028
                        10092   10118   10126   10476   10481   10482   10512   10688
$STUP = 177777           3672#
$SUPRS  054250           4403    6893   10871#
$SVLAD  050530          10038   10067#
$SVPC = 000274           1505#   1510
$SWR  = 167400           1137#   1152    1158    1159    1160    1161    1162    1163    1164    1165    2313    2314    2315
                         3741    3742    3744    3745    3893    3944    4106    4208    4289    4325    4375    4414    4443
                         4478    4554    4599    4835    4971    5195    5386    5467    5514    5709    5872    6001    6117
                         6318    6432    6628    6812    6910    7132    7511    7735    8066    8400    8761    8873    8891
                         8906    8912    8914   10019   10020   10021   10022   10023   10029   10041   10043   10044   10047
                        10048   10049   10056   10057   10058   10070   10073   10076   10083   10084   10085   10086   10087
                        10096   10103   10115   10119   10131
$SWREG  001232           2336#   3765
$SWRMK= 000000           1165    1166   10023   10024   10045
$TESTN  001214           2327#   4210*  10068*  11894   11901   11909   11917   11925   11933   11941   11949   11950   11961
                        11973
$TIMES  001174           2313#   3741*   3893*   3944*   4106*   4208*   4289*   4325*   4375*   4414*   4443*   4478*   4554*
                         4599*   4835*   4971*   5195*   5386*   5467*   5514*   5709*   5872*   6001*   6117*   6318*   6432*
                         6628*   6812*   6910*   7132*   7511*   7735*   8066*   8400*   8761*   8880*   8891*  10056*  10063
                        10066*  10076
$TKB    001146           2300#  10200   10207   10227   10431   10452   10463   10493   10521   10548
$TKCNT  052256          10432#  10447*  10482   10499*  10613   10615*
$TKINT  052266           3708   10447#  10473   10534
$TKQEN= 052265          10436#  10507   10618
$TKQIN  052260          10433#  10448*  10449   10505*  10506*  10507   10509*
$TKQOU  052262          10434#  10449*  10616   10617*  10618   10620*
$TKQSR  052264          10435#  10448   10509   10620
$TKS    001144           2299#  10198   10205   10227   10431   10453*  10489*  10491   10497*  10519   10535*  10545   10557*
                        10577*
$TKSRV  052336          10450   10463#
$TMP0   001160           2307#
$TMP1   001162           2308#
$TMP2   001164           2309#
$TMP3   001166           2310#
$TMP4   001170           2311#   3848*   4005    4026    4040*   4051*   4120    4135    4138*   4153*   4196*   4235*   4247
                         4483    4510    4518    4979    9044    9055    9390
$TMP5   001172           2312#   9248*   9303    9307
$TN   = 000042           1138#   1152    3870    3882    3893#   3921    3928    3944#   4090    4106#   4141    4198    4208#
                         4210    4211    4257    4267    4279    4289#   4317    4325#   4362    4367    4375#   4379    4407
                         4414#   4420    4433    4436    4443#   4466    4469    4478#   4538    4541    4554#   4585    4588
                         4599#   4779    4823    4835#   4958    4961    4971#   5175    5181    5195#   5376    5386#   5460
                         5463    5467#   5502    5514#   5665    5696    5709#   5859    5862    5872#   5997    6001#   6102
                         6117#   6254    6314    6318#   6418    6432#   6562    6619    6628#   6734    6789    6812#   6870
                         6889    6897    6910#   7061    7122    7132#   7406    7484    7503    7511#   7659    7716    7735#
                         8001    8062    8066#   8337    8396    8400#   8687    8749    8761#
$TNPWR  054164          10819   10820   10840#
```

```
$TPB    001152          2302#  10216*  10227
$TPFLG  001157          2306#  10148   10227
$TPS    001150          2301#  10214   10227
$TRAP   054404          3736   10937#
$TRAP2  054426         10948#  10959
$TRP  = 000016         10952#  10961#  10962#  10963#  10964#  10965#  10966   10967#  10968   10969#  10970#  10971#  10972#
                       10973#  10974#  10975#
$TRPAD  054440         10942   10959#
$TSTM   001004          1530#
$TSTNM  001102          2279#   4211*   8890*   8922*   9836   10018   10045   10067*  10068   10073   10077   10095   10131
$TTYIN  053454         10631   10632   10644   10662   10676   10680#
$TYPBN= ****** U       10965
$TYPDS  051336         10239#  10964
$TYPE   051002         10148#  10325   10952   10960
$TYPEC  051214         10178   10185   10192   10197#  10579
$TYPEX  051334         10220   10222   10225#
$TYPOC  052054         10381#  10961
$TYPON  052070         10380   10383#  10963
$TYPOS  052030         10376#  10962
$UNIT   001222          2330#   3788*   3851*   4221*   4228   4238   4658   4850   4887   4998   5035   5097   5158
                        5209    5246    5349    5355    5435   5544   5640   6179   6501   6697   6940   7036   7268
                        7584    7802    7927    8138    8263   8488   8613   8835   8841   9124   9143   9169   9211
                        9212    9213    9214    9341    9355   9363   9371   9379   9408   9878   9893
$UNITM  001010          1532#
$USWR   001234          2337#
$VECT1  001260          2362#
$VECT2  001262          2363#
$XOFF = 000023         10202   10227
$XON  = 000021         10209   10227   10465
$XTSTR  050334         10032#
$$GET4= 000000          8906#
$OFILL  052253         10377*  10381*  10391   10426#
$40CAT= ****** U       10029   10105
.     = 070144          1479#   1483#   1488#   1490#   1492#   1494#   1496#   1498#   1505    1506#   1508#   1510#   1511#
                        1517    1518#   1520#   1522#   2276#   2319    2456#   2457#   2458#   2460#   3729    3744    3745
                        8914    8918    9996   10076   10077   10131   10227   10293#  10346#  10431   10435#  10436   10437#
                       10680#  10681   10688#  10742   10780#  10860#  10979#  12230#  12231#  12376   12601   12605   12637
                       12661   12723   12731   12734#  12742#
.$ASTA= ****** U       10298   10301
.$X   = 001000          1517#   1522
```

```
CALIB   1663#  4848  4996  5207  9876
CHECK   1576#  4344  4622  4645  4674  4718  4745  4871  4903  4937  5019  5051  5230  5262  5310
        5421   5451  5535  5560  5624  5656  5755  5821  5909  5965  6038  6134  6165  6195  6282
        6355   6455  6482  6517  6593  6654  6683  6713  6843  6931  6956  7020  7052  7220  7254
        7284   7331  7539  7570  7600  7763  7788  7818  7887  7913  7943  8099  8124  8154  8223
        8249   8279  8449  8474  8504  8573  8599  8629  8818
CIDAE   1697#  5346  8832
COMMEN     1#  1290#
CWD2    1590#  4351  5318  5763  5829  5972  6489
DRCLR   1633#  4655  4885  5033  5094  5155  5244  5353  5432  5542  5637  6176  6499  6694  6938
        7033   7265  7581  7799  7924  8135  8260  8485  8610  8839  9891
ENDCOM     1#  1290#
EOPGM   2253#  8877
ERROR   1184#  3924  3981  3988  4002  4011  4036  4045  4064  4068  4079  4083  4117  4126  4144
        4171   4175  4179  4183  4295  4299  4301  4304  4308  4329  4347  4348  4349  4350  4356
        4359   4363  4381  4426  4430  4434  4447  4450  4457  4464  4467  4482  4487  4490  4495
        4498   4501  4508  4517  4522  4525  4530  4533  4536  4539  4558  4561  4564  4571  4577
        4580   4583  4586  4603  4607  4612  4617  4625  4626  4627  4628  4635  4648  4649  4650
        4651   4654  4661  4664  4677  4678  4679  4680  4694  4708  4713  4721  4722  4723  4724
        4735   4740  4748  4749  4750  4751  4761  4765  4775  4785  4789  4791  4794  4798  4801
        4846   4853  4859  4862  4874  4875  4876  4877  4883  4890  4893  4906  4907  4908  4909
        4916   4923  4925  4940  4941  4942  4943  4947  4951  4994  5001  5007  5010  5022  5023
        5024   5025  5031  5038  5041  5054  5055  5056  5057  5064  5079  5081  5087  5092  5100
        5103   5111  5125  5127  5133  5142  5148  5153  5161  5164  5204  5212  5218  5221  5233
        5234   5235  5236  5242  5249  5252  5265  5266  5267  5268  5273  5280  5288  5290  5314
        5315   5316  5317  5323  5326  5338  5352  5358  5361  5368  5373  5390  5394  5400  5404
        5408   5411  5424  5425  5426  5427  5430  5438  5441  5454  5455  5456  5457  5461  5471
        5477   5481  5495  5518  5528  5530  5538  5539  5540  5541  5547  5550  5563  5564  5565
        5566   5572  5576  5580  5591  5592  5593  5598  5601  5605  5608  5610  5613  5627  5628
        5629   5630  5636  5643  5646  5659  5660  5661  5662  5675  5679  5683  5713  5724  5739
        5743   5758  5759  5760  5761  5768  5771  5783  5794  5797  5809  5824  5825  5826  5827
        5834   5837  5852  5876  5884  5888  5895  5898  5912  5913  5914  5915  5923  5929  5932
        5944   5951  5954  5968  5969  5970  5971  5977  5980  5989  6005  6022  6026  6041  6042
        6043   6044  6051  6059  6062  6073  6090  6120  6129  6137  6138  6139  6140  6144  6152
        6155   6168  6169  6170  6171  6174  6182  6185  6198  6199  6200  6201  6206  6214  6222
        6225   6237  6245  6251  6258  6262  6266  6269  6272  6285  6286  6287  6288  6291  6322
        6339   6343  6358  6359  6360  6361  6368  6376  6379  6390  6407  6436  6443  6449  6458
        6459   6460  6461  6469  6472  6485  6486  6487  6488  6494  6497  6504  6507  6520  6521
        6522   6523  6526  6533  6536  6548  6555  6559  6566  6571  6575  6578  6581  6596  6597
        6598   6599  6602  6632  6642  6649  6657  6658  6659  6660  6664  6670  6673  6686  6687
        6688   6689  6692  6700  6703  6716  6717  6718  6719  6723  6743  6749  6753  6756  6759
        6767   6816  6830  6834  6846  6847  6848  6849  6852  6859  6861  6874  6914  6924  6926
        6934   6935  6936  6937  6943  6946  6959  6960  6961  6962  6968  6972  6976  6987  6988
        6989   6994  6997  7001  7004  7006  7009  7023  7024  7025  7026  7032  7039  7042  7055
        7056   7057  7058  7069  7075  7079  7082  7085  7093  7104  7108  7112  7142  7146  7150
        7170   7174  7201  7215  7223  7224  7225  7226  7230  7241  7244  7257  7258  7259  7260
        7263   7271  7274  7287  7288  7289  7290  7294  7301  7315  7319  7334  7335  7336  7337
        7369   7375  7379  7382  7385  7393  7438  7442  7445  7448  7455  7458  7470  7477  7527
        7534   7542  7543  7544  7545  7551  7557  7560  7573  7574  7575  7576  7579  7587  7590
        7603   7604  7605  7606  7610  7617  7627  7630  7642  7649  7667  7673  7677  7680  7683
        7691   7746  7758  7766  7767  7768  7769  7775  7778  7791  7792  7793  7794  7797  7805
        7808   7821  7822  7823  7824  7828  7835  7844  7847  7859  7866  7874  7882  7890  7891
        7892   7893  7900  7903  7916  7917  7918  7919  7922  7930  7933  7946  7947  7948  7949
        7953   7960  7969  7972  7984  7991  8007  8013  8017  8020  8023  8031  8082  8094  8102
        8103   8104  8105  8111  8114  8127  8128  8129  8130  8133  8141  8144  8157  8158  8159
        8160   8164  8171  8180  8183  8195  8202  8210  8218  8226  8227  8228  8229  8236  8239
```

```
        8252    8253    8254    8255    8258    8266    8269    8282    8283    8284    8285    8289    8296    8305    8308
        8320    8327    8343    8349    8353    8356    8359    8367    8405    8411    8415    8418    8421    8432    8444
        8452    8453    8454    8455    8461    8464    8477    8478    8479    8480    8483    8491    8494    8507    8508
        8509    8510    8514    8521    8530    8533    8545    8552    8560    8568    8576    8577    8578    8579    8586
        8589    8602    8603    8604    8605    8608    8616    8619    8632    8633    8634    8635    8639    8646    8655
        8658    8670    8677    8693    8699    8703    8706    8709    8717    8765    8769    8773    8776    8779    8790
        8796    8798    8802    8805    8808    8811    8821    8822    8823    8824    8827    8830    8838    8844    8847
        8854    8859    9322    9327    9331    9344    9358    9366    9374    9381    9407    9849    9861    9865    9869
        9881    9887    9890    9896    9899    9968
ESCAPE    1#    1290#
FSECA   1986#   9430
FSECB   2012#   9450
F.EAB   1557#   4636    4666    4863    4895    5011    5043    5222    5254    5412    5443    5552    5616    5648    5747
        5813    5900    5956    6030    6156    6187    6274    6347    6473    6509    6584    6674    6705    6835    6948
        7012    7044    7245    7276    7323    7561    7592    7779    7810    7904    7935    8115    8146    8240    8271
        8465    8496    8590    8621
GETPRI    1#    1290#
GETSWR    1#    1290#   8945
HDCHK3  1914#   7450    7622    7839    7964    8175    8300    8525    8650
HDTBL   1974#   6012    6329    7155
HEADER  2165#   5996    6313
LIMIT   2077#   5516    6912
LOOP    1541#   3995    4110    4422    4452    4503    4566    4841    4912    4990    5060    5107    5199    5275    5395
        5720    5779    5918    6047    6209    6364    6637    7196    7296    7522    7612    7741    7830    7870    7955
        8077    8166    8206    8291    8427    8516    8556    8641    8785
LPCHK   1604#   4803    6293    6603    6769    7095    7395    7409    7693    8033    8369    8719
MSG     3882#   3884    3928#   3930    4090#   4092    4198#   4200    4279#   4281    4317#   4319    4367#   4369    4407#
        4409    4436#   4438    4469#   4471    4541#   4543    4588#   4590    4823#   4825    4961#   4963    5181#   5183
        5376#   5378    5502#   5504    5695#   5698    5862#   5864    6101#   6104    6417#   6420    6619#   6621    6789#
        6791    6897#   6899    7122#   7124    7503#   7505    7716#   7718    8749#   8751
MULT      1#    1290#
NEWTST    1#    1290#   3882    3928    4090    4198    4279    4317    4367    4407    4436    4469    4541    4588    4823
        4961    5181    5376    5463    5502    5696    5862    5997    6102    6314    6418    6619    6789    6897    7122
        7503    7716    8062    8396    8749
OWNTAG  1535#   2378
POP       1#    1290#   10280   10340   10341   10731   10922
PUSH      1#    1290#   10239   10301   10303   10324   10705   10902
QKRPSK  1798#   6740    7066    7367    7664    8004    8340    8690
QKSEEK  1775#   6259    6568    6746    7072    7372    7435    7670    8010    8346    8408    8696    8766
QKSRT   1753#   5672    7101    9858
QKUNLD  1822#   4293    4783
RALLHD  1924#   6054    6371
RDHDR   1878#   5790    5925    6218    6529    7451    7623    7840    7965    8176    8301    8526    8651
REPORT    1#    1290#
SCOPE   1185#   3892    3943    4105    4207    4288    4324    4374    4413    4442    4477    4553    4598    4834    4970
        5194    5385    5466    5513    5708    5871    6000    6116    6317    6431    6627    6811    6909    7131    7510
        7734    8065    8399    8760    8878    8889
SECTST  2041#   5475
SETPRI    1#    1290#   10609
SETTRA 10952#   10961   10962   10963   10964   10966   10968   10969   10970   10971   10972   10973   10974
SETUP     1#    1290#   3724
SKATN   1734#   5406    6150    6668    7239    7555    7773    7898    8109    8234    8459    8584
SKIP      1#    1290#   3870    3921    4141    4257    4267    4362    4379    4420    4433    4466    4538    4585    4779
        4958    5175    5460    5665    5859    6254    6562    6734    6870    6889    7061    7406    7484    7659    8001
        8337    8687
SKOSC   2197#   7737    8073    8423
```

```
SKRDY    1721#  6126   6646   7212   7531   7755   7879   8091   8215   8441   8565
SLASH       1#  1290#
SPACE    1290#
STARS       1#  1290#  1503   1514   1516   1523   2272   2319   2322   3882   3891   3928   3942   4090   4104
         4198   4206   4279   4287   4317   4323   4367   4373   4407   4412   4436   4441   4469   4476   4541
         4552   4588   4597   4823   4833   4961   4969   5181   5193   5376   5384   5463   5465   5502   5512
         5696   5707   5862   5870   5997   5999   6102   6115   6314   6316   6418   6430   6619   6626   6789
         6810   6897   6908   7122   7130   7503   7509   7716   7733   8062   8064   8396   8398   8749   8759
         8870  10015  10079  10133  10229  10296  10353  10430  10512  10527  10598  10622  10691  10744  10783
        10801  10863  10886  10931  12146  12155
SWRSU       1#  1290#  3746#
SW814    1612#  4778   5664   6252   6560   6732   7060   7405   7483   7657   7999   8335   8685
TRMTRP  10952#
TSTSW9   1622#  4812   4817   5686   6300   6305   6609   6614   6777   6783   7115   7415   7421   7487   7492
         7701   7707   8039   8044   8049   8054   8375   8380   8385   8390   8725   8730   8735   8740
TYPBIN      1#  1290#
TYPDEC      1#  1290#  8901
TYPNAM      1#  1290#
TYPNUM      1#  1290#
TYPOCS      1#  1290#  3963   4028   4074   4239   9198   9941   9983
TYPOCT      1#  1290#  10540
TYPTXT      1#  1290#
WRHDR    1855#  5736   5881   6019   6336   7167   7312
$$CMRE   2270#
$$CMTM   2270#  2307   2308   2309   2310   2311   2312
$$ESCA      1#  1290#
$$NEWT      1#  1290#  3882   3928   4090   4198   4279   4317   4367   4407   4436   4469   4541   4588   4823
         4961   5181   5376   5463   5502   5696   5862   5997   6102   6314   6418   6619   6789   6897   7122
         7503   7716   8062   8396   8749
$$SET   10952#  10961  10962  10963  10964  10966  10968  10969  10970  10971  10972  10973  10974
$$SETM   3762#
$$SKIP      1#  1290#  3870   3921   4141   4257   4267   4362   4379   4420   4433   4466   4538   4585   4779
         4958   5175   5460   5665   5859   6254   6562   6734   6870   6889   7061   7406   7484   7659   8001
         8337   8687
.EQUAT      1#  1137#  1180
.HEADE      1#  1137#  1142
.KT11       1#
.SETUP      1#  1137#  3672
.SWRHI      1#  1137#  1154
.SWRLO   1137#  1166#
.$ACT1      1#  1137#  1501
.$APTB      1#  2320#
.$APTH      1#  1137#  1512
.$APTY      1#  1137#  10294
.$ASTA      1#
.$CATC      1#  1137#  1477
.$CMTA      1#  1137#  2270
.$DB2D      1#  1137#  10799
.$DB2O      1#  1137#  10742
.$DIV       1#
.$EOP       1#  1137#  8868
.$ERRO      1#  1137#  10077
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
```

```
.$RDDE      1#
.$RDOC      1#   1137#  10689
.$READ      1#   1137#  10428
.$R2AZ      1#
.$SAVE      1#   1137#  10884
.$SB2D      1#   1137#  10781
.$SB2O      1#
.$SCOP      1#   1137#  10013
.$SIZE      1#
.$SUPR      1#   1137#  10861
.$TRAP      1#   1137#  10929
.$TYPB      1#
.$TYPD      1#   1137#  10227
.$TYPE      1#   1137#  10131
.$TYPO      1#   1137#  10351
.$40CA      1#
.1170       1#


. ABS.  072214     000


 ERRORS DETECTED:  0

 CZR6HF,CZR6HF.LST/SOL/CRF/NL:TOC=SYSMAC.SML,CZR6HF.P11
 RUN-TIME: 24 31 3 SECONDS
 RUN-TIME RATIO: 175/59=2.9
 CORE USED:  43K  (86 PAGES)
```