RK06

DISK DRIVE DIAG. PART 1

MD-11-DZR6H-C

EP-DZR6H-C-DL-B

FICHE 1 OF 2

DEC 1976

digital

MADE IN USA

RK06

DISK DRIVE DIAG. PART 1

MD-11-DZR6H-C

EP-DZR6H-C-DL-B

FICHE 2 OF 2

DEC 1976

digital

MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6H-C-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 1
DATE:	DECEMBER 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 1 OF THE DRIVE DIAGNOSTICS
TO INSURE THAT THE
DISK IS CAPABLE OF PERFORMING ALL STATIC & CYCLE UP TESTS.
IT INSURES THAT THE DRIVE CAN WRITE AND READ HEADERS
IN BOTH 20 & 22 SECTOR FORMATS.
FINALLY, IT INSURES THAT THE DISK CAN PERFORM SEEK
OPERATIONS BY DOING SEVERAL SEEK PATTERNS.
ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.

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

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

*****CAUTION*****

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

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

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

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

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

2.2 PRELIMINARY TESTING & PROGRAMS

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

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

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

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

3.2 XXDP

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

CHAIN MODE OPERATION (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

DUMP MODE OPERATION (MANUAL)

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

3.3 ACT/APT

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

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

AUTOMATIC MODE (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRORUPT 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. INTERRORUPT 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. INTERRORUPT 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 36 FORMAT PACK                          : 1 MIN
TEST 41 SEEK FROM CYL 0 TO ALL CYLS          : 40 SEC
```

TEST 42 SEEK FROM CYL 410 TO ALL CYLS : 40 SEC

3.9 FAULT ISOLATION

TO BE DETERMINED.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OR FAILURE RATE ANALYSIS.

3.11 DEFAULT UNIBUS ADDRESSES & VECTORS

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

	LOCATION	DEFAULT CONTENTS
RK06 BUSS ADDRESS	1264	177440
CONTROLLER INTERRORUPT 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 INTERRORUPT VECTOR	1330	100
P-CLOCK INTERRORUPT 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 INTERRORUPT 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. INTERRORUPT VECTOR INPUTTED ONLY ON 1ST PASS.

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

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

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

THE PURPOSE OF BYPASSING IS TO PROVIDE
A QUICK MODULE TEST

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

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

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

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

4.3.1 SW<15>

THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT THE ERROR MSG AND PERTINENT INFORMATION, IF SW13=0.
PRESSING "CONTINUE" RESTORES NORMAL 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).

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 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 INTERRORUPT.
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 INTERRORUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRORUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

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

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

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

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

                                TYPE IN CONTROLLER INTERRORUPT VECTOR IF NOT 210 <CR>
                                                                             ----

                                WILL TEST DRIVES:
                                1
                                3

                                DRIVE 1

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


                        4.6     PROGRAM EXAMPLE

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

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

                                        WILL TEST DRIVES:
                                        0
                                        1

                                        DRIVE 0

                                        DRIVE SERIAL NO. AAA
                                        CARTRIDGE SERIAL NO. BBB

                                        DRIVE 1

                                        DRIVE SERIAL NO. CCC
                                        CARTRIDGE SERIAL NO. DDD

                                        END PASS #1


                                        WILL TEST DRIVES:
                                        0
                                        1

                                        DRIVE 0

                                        DRIVE SERIAL NO. AAA
                                        CARTRIDGE SERIAL NO. BBB

                                        DRIVE 1
```

```
                                DRIVE SERIAL NO. CCC
                                CARTRIDGE SERIAL NO. DDD

                                END PASS # 2

                                   (ETC)

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

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

                4.7     HALTING THE PROGRAM

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

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

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

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

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

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


                        NOTES:

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

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

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


        5.0     DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION
```

5.1 GENERAL

A. BASIC CONTROLLER TESTS, SIZING & SETUP

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

B. STATIC & CYCLE UP TESTS

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

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

C. SEEK, WRITE HEADER, READ HEADER TESTS

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

5.2 TEST DESCRIPTIONS

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

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

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

TEST 2 SIZE THE BUSS

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

```
                TEST 3  VERIFY OPERATOR DRIVE SELECTIONS

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

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


                TEST 4  FIND NEXT DRIVE TO BE TESTED

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

                TEST 5  UNLOAD DRIVE TO BE TESTED

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


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


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

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


                TEST 7  PRINT DRIVE SERIAL NUMBER

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


                TEST 10 SET VV WITH PACK CMD

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


                TEST 11 RELEASE DRIVE
```

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


TEST 12 DRIVE TYPE TEST

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


TEST 13 C-D PARITY ERROR DETECTION

        TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
        THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
        THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
        THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
        CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
        AN ERROR AND SET SPAR.
        THE ERROR CONDITION IS RESET WITH THE CLEAR CMD


TEST 14 VERIFY START SPINDLE CMD

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


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


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

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


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

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


TEST 17 HEAD REGISTER TEST

        THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
        VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
```

HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.

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

TEST 20 SEEK TO CYL 0

TESTS THE ABILITY TO DO A SEEK CMD.
VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
STATUS BITS. VERIFIES CMD COMPLETION BETWEEN 10-15USEC.
READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.

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

TEST 22 DETECT OUTER LIMIT

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

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

TEST 23 READ HEADERS, ALL TRACKS, CYL 0

THIS TEST DOES NOT CHECK THE HEADERS, BUT ONLY THE FACT
THAT HEADERS CAN BE READ CORRECTLY BY CHECKING CERR & DTE.

THIS TEST IS VALID ONLY IF THE PACK HAS NOT BEEN ZERO'D OUT.

TEST 24 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 25 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 26 WRITE & READ HEADERS CYL 0, HEAD 0


                TEST 27 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 30 WRITE & READ HEADERS CYL 1, HEAD 0


                TEST 31 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 32 SINGLE INCREMENT SEEKS TO CYL 410

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


                TEST 33 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

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

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

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

```
                        TEST 34 DETECT INNER LIMIT

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


                        TEST 35 FORMAT PACK

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


                        TEST 36 DECREMENT FROM CYL 410 TO 0 & READ HEADERS

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


                        TEST 37 SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS

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

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

                        TEST 40 SEEK TO ALL CYLS FROM 0 & READ HEADERS

                        TEST 41 SEEK TO ALL CYLS FROM CYL 410 & READ HEADERS

                        TEST 42 SEEK TO ALL KEY INVALID CYLS

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

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

                                THIS TEST CHECKS KEY INVALID CYLS 411,412,416,448 & 480
```

FOR A FULL LOGIC TEST

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

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

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION

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

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

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLES:

EXAMPLE #1:

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

  TEST NO.  PC
  000014  016530
EXPECT  EXPECT  EXPECT  EXPECT  EXPECT  EXPECT  EXPECT
A0      B0      A1      B1      A2      B2      B3
030144  100000  013704  000001
ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL
A0      B0      A1      B1      A2      B2      B3
```

```
                                140144  100000  101744  000001
                                RKCS1   RKCS2   RKASOF  RKER    RKDS    RKDC
                                040200  000100  010000  000000  000000  000000

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

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

                                EXAMPLE #2:

                                NO ATTN IN RKASOF
                                AFTER UNLOAD CMD

                                  TEST NO.  PC
                                  000003  014330
                                RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2   RKASOF
                                000144  100000  000000  100101  000206  000104  000000


                                        [ END OF DOCUMENT ]

                        %
```

```
                        .NLIST  CND,MC,MD
                        .LIST   ME
                        .ENABL  ABS,AMA

                        ;DEFINE SYSMAC MACROS

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


                        .TITLE  UNIBUS RK06 DRIVE DIAGNOSTIC  PART 1
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY GARY PAPAZIAN
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                        ;*


                        .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
                        ;*              260     RUN MODULE TEST VERSION-DEFAULT MODE ONLY BYPASS
                        ;*                      TESTS 35,36,40 & 41
                        ;*              270     SAME AS 260 START BUT BYPASS TEST 16 ALSO
                        ;*
```

```
                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

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

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

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

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

                        .SBTTL  RK06 CONTROLLER REGISTER DEFINITION
```

```
                        ;       $BASE=177440
                        
000000                  RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
000002                  RKWC=   2               ;WORD COUNT REGISTER
000004                  RKBA=   4               ;BUS ADDRESS REGISTER
000006                  RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
000010                  RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
000012                  RKDS=   12              ;DRIVE STATUS REGISTER
000014                  RKER=   14              ;ERROR REGISTER
000016                  RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
000020                  RKDC=   20              ;DESIRED CYL REGISTER
000024                  RKDB=   24              ;DATA BUFFER
000026                  RKMR1=  26              ;MAINTENANCE REGISTER 1
000034                  RKMR2=  34              ;MAINTENANCE REGISTER 2 (MSG LINE A)
000036                  RKMR3=  36              ;MAINTENANCE REGISTER 3 (MSG LINE B)
000030                  RKECPS= 30              ;ECC POSITION INFORMATION
000032                  RKECPT= 32              ;ECC PATTERN INFORMATION

                        .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)

                        ;       DRIVE CMDS

000001                  SELDRV= 1               ;SELECT DRIVE (GET STATUS)
000003                  PACK=   3               ;PACK ACKNOWLEDGE
000005                  CLEAR=  5               ;DRIVE CLEAR
000007                  UNLOAD= 7               ;UNLOAD
000011                  SRTSPL= 11              ;START SPINDLE
000013                  RECAL=  13              ;RECALIBRATE
000015                  OFFSET= 15              ;OFFSET
000017                  SEEK=   17              ;SEEK
000021                  RDDATA= 21              ;READ DATA
000023                  WRDATA= 23              ;WRITE DATA
000025                  RDHEAD= 25              ;READ HEADER
000027                  WRHEAD= 27              ;WRITE HEADER AND DATA
000031                  WRTCHK= 31              ;WRITE CHECK

000001                  GO=     BIT0            ;GO BIT
000100                  IE=     BIT6            ;INTERRORUPT ENABLE
000200                  RDY=    BIT7            ;CONTROLLER READY
000400                  BA16=   BIT8            ;BUS ADDRESS BIT 16
001000                  BA17=   BIT9            ;BUS ADDRESS BIT 17
002000                  CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06)
004000                  CTO=    BIT11           ;CONTROLLER TIMEOUT
010000                  CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
020000                  DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
040000                  DI=     BIT14           ;DRIVE INTERRORUPT
100000                  CERR=   BIT15           ;CONTROLLER ERROR
100000                  CCLR=   BIT15           ;CONTROLLER CLEAR

                        .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)

000007                  DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
000010                  RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
000020                  BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
000040                  SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
1345            000100          IR=     BIT6                    ;INPUT READY
1346            000200          OR=     BIT7                    ;OUTPUT READY
1347            000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1348            001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1349            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1350            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1351            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1352            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1353            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1354            100000          DLT=    BIT15                   ;DATA LATE ERROR
1355
1356                            .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1357
1358            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1359            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
1360            000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
1361            000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
1362            000020          FMTE=   BIT4                    ;FORMAT ERROR
1363            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1364            000100          ECH=    BIT6                    ;ECC HARD
1365            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
1366            000400          HVRC=   BIT8                    ;HEADER VRC ERROR
1367            001000          COE=    BIT9                    ;CYL ADDRESS OVERFLOW ERROR
1368            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1369            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1370            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
1371            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1372            040000          UNS=    BIT14                   ;DRIVE UNSAFE
1373            100000          DCK=    BIT15                   ;DATA CHECK
1374
1375                            .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1376
1377            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1378                                                            ;   THIS BIT IS RESET)
1379            000004          OFST=   BIT2                    ;DRIVE OFFSET
1380            000010          ACLO=   BIT3                    ;AC LOW
1381            000020          DCLO=   BIT4                    ;DC LOW
1382            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
1383            000100          VV=     BIT6                    ;VOLUME VALID
1384            000200          DRDY=   BIT7                    ;DRIVE READY
1385            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
1386            004000          WRL=    BIT11                   ;WRITE LOCK
1387            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
1388            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
1389            100000          SVAL=   BIT15                   ;STATUS VALID
1390
1391                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1392
1393            000017          MESMSK= 17                      ;MSG MASK
1394            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MSG LINES
1395            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
1396            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1397            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
1398            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
1399            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1400            002000          MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
```

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

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

                        D.DRA=  BIT5                    ;DRIVE AVAILABLE
                        D.VV=   BIT6                    ;VOLUME VALID
                        D.DRDY= BIT7                    ;DRIVE READY
                        D.DDT=  BIT8                    ;DRIVE TYPE (0=RK06)
                        D.FORM= BIT9                    ;DRIVE FORMAT
                        D.OFF=  BIT10                   ;OFFSET ON
                        D.WRL=  BIT11                   ;WRITE LOCK
                        D.SPIN= BIT12                   ;SPINDLE ON
                        D.PIP=  BIT13                   ;POSITIONING IN PROGRESS
                        D.DSC=  BIT14                   ;DRIVE STATUS CHANGE

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

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

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

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

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

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

```
1457            010000                  D.NMOV= BIT12                   ;SEEK AND NO MOTION
1458            020000                  D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1459            040000                  D.SUNS= BIT14                   ;SERVO UNSAFE
1460
1461                                    .SBTTL  COMMON MASKS AND OTHER BITS: MSG A (RKMR2:34)
1462
1463            000007                  M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1464            017760                  M.CDIF= 17760                   ;CYL DIFF, BYTE 10
1465            017760                  M.OFST= 17760                   ;OFFSET VALUE, BYTE 10
1466            077770                  M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1467
1468                                    .SBTTL  COMMON MASKS AND OTHER BITS: MSG B (RKMR3:36)
1469
1470            000003                  M.ID=   3                       ;BYTE ID, ALL BYTES
1471            017760                  M.CADD= 17760                   ;CYL ADDRESS, BYTE 10
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 CONTRIN 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  007132          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                        000204          .=204
000204  000137  007026          JMP     BYT16           ;BYPASS N-SQUARE TEST IN DEFAULT MODE
                        000220          .=220
000220  000137  007006          JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING
                        000230          .=230
000230  000137  007046          JMP     BYT16A  ;BYPASS N-SQUARE TEST IN PARAM MODE
                        000260          .=260
000260  000137  007070          JMP     MDTST           ;MODULE TESTS DEFAULT MODE ONLY
                        000270          .=270           ;BYPASS SEVERAL TESTS
000270  000137  007110          JMP     MDTSTA          ;SAME AS 260 & BYPASS N-SQUARE TEST ALSO

                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000274          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  043604                  $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)
001010  001130          $UNITM: .WORD   600.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000042                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                        .LIST   MD
                ;
                ;USE LOOP X TO OMIT JSR PC, SUBCLR
                ;


                ;
                ; THIS MACRO FILLS EXPECTED MSG A0,B0,A1,B1,A2,B2 & B3 WITH STANDARD BITS SET
                ; A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
                ; NOTE: A CAN BE ANY BIT COMBINATION DESIRED

                ;
                ; THIS MACRO ASSUMES DRIVE MSG A0,B0,A1,B1 WILL ALWAYS BE TESTED
                ; USE   A,C,D,E FOR MSG A0,B0,A1,B1 ERROR NUMBERS RESP.
                ; USE   G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
                ;       H=T.B2  TO READ MSG B2 & PUT INFOR INTO 'CYLADD'
                ;       I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
                ;
                ;       F=      < ERROR DESCRIPTION>
                ;
                ;
                ; A=CYL DIFF/OFFSET ERROR #
                ; B=CYL ADDR ERROR #
                ; C= <ERROR DESCRIPTION>
                ;


                ;
                ;SWR9 (LOOP ON ERROR) TEST       A=BRANCH POINT TO RECONDITION DRIVE
                ;                               B=JMP POINT TO RE-ENTER MAIN LINE
                ;

                ;
                ; USE DRCLR     X    TO OMIT CHECKING MSG A0,B0,A1 & B1
                ;


                ;
                ;A=BLANK TO CHECK A0 THRU B2
                ;A=NON BLANK TO OMIT CHECKING A0 THRU B2
                ;D=BLANK TO CHECK A0 THRU B2 IN DRCLR
                ;D=NON-BLANK TO OMIT CHECKING A0 THRU B2 IN DRCLR
                ;

                ;
                ;IDAE IS CLEARED ONLY BY RECAL & DRIVE CLEAR
                ;

                ;
                ; A=D.FWD/D.REV
                ;
```

```
1588
1589                                    ;
1590                                    ;QUICK START SPINDLE.
1591                                    ;
1592
1593                                    ;
1594                                    ;QUICK SEEK.    ENTER WITH CYL # IN RKDC
1595                                    ;
1596
1597                                    ;
1598                                    ;QUICK REPETITIVE SEEKS
1599                                    ;       A=INC/DEC    CYL#
1600                                    ;       B=FIND VALUE OF CYL# BEFORE EXITING
1601
1602
1603                                    ;
1604                                    ;QUICK UNLOAD
1605                                    ;D=BLANK TO DO SUBCLR & LPCHK
1606                                    ;D=NON-BLANK TO BYPASS
1607                                    ;
1608
1609                                    ;
1610                                    ; A=WRHEAD/<CFMT!WRHEAD>
1611                                    ; USE WRHDR     <A>,X   TO OMIT CHECKING A0,B0,A1,B1
1612                                    ;
1613
1614                                    ;
1615                                    ; A=RDHEAD/<CFMT!RDHEAD>
1616                                    ; USE RDHDR <A>,X TO OMIT CHECKING A0,B0,A1,B1
1617                                    ;
1618
1619                                    ;
1620                                    ;A=TOCYL/FRCYL
1621                                    ;B=310 FOR TOCYL/311 FOR FRCYL
1622                                    ;
1623
1624
1625
1626
1627
1628                                    ;
1629                                    ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1630                                    ;
1631                                    ;
1632                                    ;USE FSECA FS022,RDSEC,22 FOR 22 SECTOR FORMAT
1633                                    ;USE FSECA FS020,R20SEC,20 FOR 20 SECTOR FORMAT.
1634                                    ;
1635                                    ;
1636                                    ;USE FSECB FNS22,RDSEC,22 FOR 22 SECTOR FORMAT
1637                                    ;USE FSECB FNS20,R20SEC,20 FOR 20 SECTOR FORMAT
1638                                    ;
1639
1640                                    ;
1641                                    ;USE SECTST FS022,FNS22,RDSEC FOR 22 SECTOR FORMAT
1642                                    ;USE SECTST FS020,FNS20,R20SEC FOR 20 SECTOR FORMAT
1643                                    ;
```

```
1644
1645
1646                            ;
1647                            ;DETECT OUTER LIMIT: 1,0,D.REV,OUTER
1648                            ;DETECT INNER LIMIT: 409.,410.,D.FWD,INNER
1649                            ;
```

```
1650
1651
1652                            ;
1653                            ; A=CYL#, B=HEAD#
1654                            ;
1655
1656
1657
1658                            ;
1659                            ;SEEK TO MAJOR CYL: 0,1,TEMP3,TEMP4,D.FWD,D.REV,ASL,400,DEC,0
1660                            ;SEEK 0 TO ALL CYL: 0,1,TEMP3,TEMP4,D.FWD,D.REV,INC,410.,DEC,0
1661                            ;SEEK 410 TO ALL CYL: 410.,409.,TEMP4,TEMP3,D.REV,D.FWD,DEC,0,INC,410.
1662                            ;
1663
1664
1665                                    .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
```

```
1722  001212  000000                $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
1723  001214  000000                $TESTN: .WORD   ATESTN  ;;TEST NUMBER
1724  001216  000000                $PASS:  .WORD   APASS   ;;PASS COUNT
1725  001220  000000                $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
1726  001222  000000                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
1727  001224  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
1728  001226  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1729  001230                        $ETABLE:                ;;APT ENVIRONMENT TABLE
1730  001230     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1731  001231     000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1732  001232  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1733  001234  000000                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1734  001236  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1735                                ;*                      BITS 15-11=CPU TYPE
1736                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1737                                ;*                              11/70=06,PDQ=07,Q=10
1738                                ;*                      BIT 10=REAL TIME CLOCK
1739                                ;*                      BIT  9=FLOATING POINT PROCESSOR
1740                                ;*                      BIT  8=MEMORY MANAGEMENT
1741  001240     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1742  001241     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1743                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1744                                ;*                              900 NSEC CORE=001
1745                                ;*                              300 NSEC BIPOLAR=002
1746                                ;*                              500 NSEC MOS=003
1747  001242  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1748                                ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1749  001244     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1750  001245     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1751  001246  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1752  001250     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1753  001251     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1754  001252  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1755  001254     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1756  001255     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1757  001256  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1758  001260  000000                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1759  001262  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1760  001264  177440                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1761  001266  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1762  001270  000000                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1763  001272  000000                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1764  001274  000000                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
1765  001276  000000                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
1766  001300  000000                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
1767  001302  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1768  001304  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1769  001306  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1770  001310  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1771  001312  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1772  001314                        $ETEND:
1773                                .MEXIT
1774          177440                ABASE=  177440                  ;DEFAULT BUSS ADDRESS
1775  001314  000210                RKVEC:  210                     ;DEFAULT CONTROLLER INTERRORUPT VECTOR
1776  001316  000240                RKPRI:  PR5                     ;PRIORITY
1777  001320  172540                PKS:    172540                  ;P-CLOCK STATUS REG
```

```
1778  001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
1779  001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
1780  001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.
1781
1782  001330  000100          LCVEC:  100             ;L-CLOCK INTERRORUPT VECTOR
1783  001332  000104          PCVEC:  104             ;P-CLOCK INTERRORUPT VECTOR.
1784
1785          000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
1786          172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
1787  001334  000000          TRAPPC: 0               ;PC FOR MEMORY CHECK ENABLE TRAP
1788
1789  001336  000000          PARAM:  0               ;1 FOR 220 OR 230 START, NO DEFAULT
1790  001340  000000          BYPT16: 0               ;1 FOR 210, 230, 270 START
1791  001342  000000          MODTST: 0               ;1 FOR 260 OR 270 START
1792  001344  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1793
1794  001346  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
1795                                                  ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1796  001350  000000          FRCYL:  0               ;FROM CYL
1797  001352  000000          TOCYL:  0               ;TO CYL
1798  001354  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
1799  001356  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
1800  001360  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
1801  001362  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1802  001364  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
1803  001366  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
1804
1805  001370  000074          HZ:     60.             ;60 FOR 60 CPS
1806                                                  ;50 FOR 50 CPS
1807  001372  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1808                                                  ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1809  001374  000000          SEC:    0               ;SECOND COUNTER
1810  001376  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
1811  001400  000000          SECNT:  0               ;SECTOR COUNT
1812  001402  000000          PSEC:   0               ;PREVIOUS SECTOR
1813  001404  000000          ESEC:   0               ;EXPECTED SECTOR
1814  001406  000000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1815
1816  001410  000000          LPFLG:  0               ;SET TO 0 TO RETURN TO $LPADR
1817                                                  ;IF SW14 OR SW8 SET
1818                                                  ;SET TO 1 TO RETURN TO $LPERR
1819                                                  ;IF SW9 SET
1820  001412  000001          T1:     1               ;TIMEOUT CONSTANTS
1821  001414  000012          T10:    10.
1822  001416  000144          T100:   100.
1823  001420  000764          T500:   500.
1824  001422  004704          T2500:  2500.
1825  001424  011610          T5000:  5000.
1826  001426  141520          T50000: 50000.
1827
1828  001430  000000          HEAD:   0               ;HEAD NUMBER
1829  001432  000000          HEADA:  0               ;HEAD # FROM H.B3 RIGHT JUSTIFIED
1830  001434  000000          HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
1831  001436  000000          FORMAT: 0               ;FORMAT TYPE
1832  001440  000000          FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
1833  001442  000000          WDCNT:  0               ;WORD COUNT
```

```
1834
1835  001444  000000                DATA0:  0                       ;ALL 0'S
1836  001446  052525                DATA01: 52525                   ;0101 PATT
1837  001450  177777                DATA1:  177777                  ;ALL 1'S
1838
1839  001452  000000                WORD:   0                       ;HEADER/DATA WORD
1840  001454  000000                HDWD:   0                       ;HEADER WORD FROM RKDB
1841
1842  001456  000000                BSERR:  0                       ;CANNOT READ BSE INFO WHEN SET
1843  001460  000000                LIMERROR:       0                       ;LIMIT DETECT ERROR FLAG
1844
1845  001462  000000                BYPCERR:0                       ;SET TO 1 TO BYPASS CKCERR IN GSTAT1 ROUTINE
1846  001464  000000                BYPFMT: 0                       ;BYPASS FORMAL FORMATTING OF HEADERS
1847                                                                ;UNTIL BSE INFO HAS BEEN STORED.
1848                                                                ;IF SET, BIT 14,15 = 1
1849
1850  001466  000000                CHKFLG: 0                       ;WORDS TO BE CHECKED
1851
1852  001470  000102                HDTAB:  .BLKW 66.               ;CALCULATED HEADER WORD TABLE
1853  001674  000102                RHTAB:  .BLKW 66.               ;FILLED AFTER READ HEADER CMD
1854  002100  000102                SRTTAB: .BLKW 66.               ;ABOVE RHTAB SORTED STARTING FORM
1855                                                                ;SECTOR 0 BY SORT ROUTINE
1856  002304  000400                BSE22H: .BLKW 256.              ;22 SECTOR HARDWARE BSE INFO.
1857                                                                ;22 SECTOR SOFTWARE BSE INFO
1858                                                                ;OVERLAYS MSG1
1859
1860  003304  000633                INVCYL: 411.                    ;INVALID CYL ADDR
1861  003306  000634                        412.
1862  003310  000640                        416.
1863  003312  000700                        448.
1864  003314  000740                        480.
1865
1866  003316  000000                UNLD:   0                       ;SET TO 0 IF HEADS ARE LOADED
1867                                                                ;SET TO 1 IF HEADS UNLOADED
1868  003320  000000                BADHDR: 0                       ;SET TO 0 IF FORMATTING OK
1869                                                                ;SET TO 1 IF FORMATTING ALTERED
1870  003322  000000                HPEND:  0                       ;SET TO 0 IF HALT NOT PENDING
1871                                                                ;SET TO 1 IF HALT PENDING
1872
1873                                                                ;THE ABOVE 3 FLAGS ARE USED
1874                                                                ;BY 'STOP' ROUTINE TO BRING
1875                                                                ;THE CPU TO A VALID HALT.
1876
1877  003324    001   002   004   ATTN:   .BYTE 1,2,4,10,20,40,100,200     ;ATN 0-7 RESP.
1878  003327    010   020   040
1879  003332    100   200
1880                                .EVEN
1881
1882
1883                                ;
1884                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
1885                                ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
1886                                ;
1887
1888  003334  000000                HCS1:   0                       ;HOLD RKCS1
1889  003336  000000                HCS2:   0                       ;HOLD RKCS2
```

```
  1890  003340  000000                HWC:    0                       ;HOLD RKWC
  1891  003342  000000                HBA:    0                       ;ETC.
  1892  003344  000000                HDA:    0
  1893  003346  000000                HDS:    0
  1894  003350  000000                HER:    0
  1895  003352  000000                HASOF:  0
  1896  003354  000000                HDC:    0
  1897  003356  000000                HDB:    0
  1898  003360  000000                HMR1:   0
  1899  003362  000000                HMR2:   0
  1900  003364  000000                HMR3:   0
  1901  003366  000000                HPOS:   0
  1902  003370  000000                HPAT:   0
  1903
  1904  003372  000000                TEMP1:  0                       ;TEMPORARY STORAGE.
  1905  003374  000000                TEMP2:  0
  1906  003376  000000                TEMP3:  0
  1907  003400  000000                TEMP4:  0
  1908  003402  000000                TEMP5:  0
  1909                                ;
  1910                                ; THE FOLLOWING ARE HOLDING REGISTERS FOR MSG A (0-3) & MSG B (0-3)
  1911                                ;
  1912  003404  000000                H.A0:   0
  1913  003406  000000                H.B0:   0
  1914  003410  000000                H.A1:   0
  1915  003412  000000                H.B1:   0
  1916  003414  000000                H.A2:   0
  1917  003416  000000                H.B2:   0
  1918  003420  000000                H.A3:   0
  1919  003422  000000                H.B3:   0
  1920                                ;
  1921                                ; THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE
  1922                                ;
  1923  003424  000000                E.A0:   0
  1924  003426  000000                E.B0:   0
  1925  003430  000000                E.A1:   0
  1926  003432  000000                E.B1:   0
  1927  003434  000000                E.A2:   0
  1928  003436  000000                E.B2:   0
  1929  003440  000000                E.A3:   0
  1930  003442  000000                E.B3:   0
  1931
  1932                                ;
  1933                                ; THE FOLLOWING ARE IDENTITIES FOR DRIVE MSG WORDS TO BE TESTED
  1934                                ;
  1935          000001                T.A2=BIT0                       ;TEST MSG A2 IF SET
  1936          000002                T.B2=BIT1
  1937          000004                T.B3=BIT2
  1938
  1939                                ;
  1940                                ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
  1941                                ;
  1942
  1943  003444  000000                DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
  1944  003446  000000                DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
  1945  003450  000000                ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
```

```
1946  003452  000000                  PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
1947  003454  000000                  DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
1948
1949                                  ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
1950                                  ;IS PRESENT AND IS TO BE TESTED.
1951
1952  003456  000000                  DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
1953  003460  000000                  DRIV1:  0                       ;FOR DRIVE 1
1954  003462  000000                  DRIV2:  0                       ;FOR DRIVE 2
1955  003464  000000                  DRIV3:  0                       ;FOR DRIVE 3
1956  003466  000000                  DRIV4:  0                       ;FOR DRIVE 4
1957  003470  000000                  DRIV5:  0                       ;FOR DRIVE 5
1958  003472  000000                  DRIV6:  0                       ;FOR DRIVE 6
1959  003474  000000                  DRIV7:  0                       ;FOR DRIVE 7
1960
1961  003476  000000                  LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
1962  003500  000000                  PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
1963  003502  000000                  DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
1964  003504  000000                  SIZFLG: 0                       ;SET IF DEFAULT DO SIZING IN TEST 1
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


003506                          $ERRTB:

                                ;ERROR 1
003506  057327                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
003510  064066                          DH1
003512  066606                          DT1
003514  067240                          DF1

                                ;ERROR 2
003516  057464                          EM5                     ;DETECTED MDS
003520  064066                          DH1
003522  066606                          DT1
003524  067240                          DF1

                                ;ERROR 3
003526  057505                          EM6                     ;DETECTED UFE
003530  064066                          DH1
003532  066606                          DT1
003534  067240                          DF1

                                ;ERROR 4
003536  057526                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
003540  064066                          DH1
003542  066606                          DT1
003544  067240                          DF1
                                ;ERROR 5
003546  057615                          EM8                     ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
003550  064066                          DH1
003552  066606                          DT1
003554  067240                          DF1

                                ;ERROR 6
003556  057665                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
003560  064066                          DH1
003562  066606                          DT1
003564  067240                          DF1

                                ;ERROR 7
003566  057735                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
003570  064066                          DH1
003572  066606                          DT1
003574  067240                          DF1
```

```
                    2021
                    2022
                                        ;ERROR 10
003576  060000                                EM11            ;DRA & NED BOTH SET
003600  064066                                DH1
003602  066606                                DT1
003604  067240                                DF1
                    2027
                    2028
                                        ;ERROR 11
003606  060044                                EM12            ;CONTROLLER NOT READY
003610  064754                                DH18            ;AFTER UNLOAD CMD.
003612  066606                                DT1
003614  067334                                DF10
                    2033
                    2034
                                        ;ERROR 12
003616  060075                                EM13            ;NO ATTN
003620  064754                                DH18            ;AFTER UNLOAD CMD
003622  066606                                DT1
003624  067334                                DF10
                    2039
                                        ;ERROR 13
003626  060117                                EM14            ;WRONG ATTN
003630  064754                                DH18
003632  066606                                DT1
003634  067334                                DF10
                    2044
                                        ;ERROR 14
003636  060144                                EM15            ;DRDY NOT CLEARED
003640  064754                                DH18
003642  066606                                DT1
003644  067334                                DF10
                    2049
                                        ;ERROR 15
003646  060176                                EM16            ;DSC NOT SET
003650  064754                                DH18
003652  066606                                DT1
003654  067334                                DF10
                    2054
                                        ;ERROR 16
003656  060223                                EM17            ;MSG A0 ERROR
003660  064354                                DH8             ;IN UNLD
003662  067054                                DT13
003664  067454                                DF20
                    2059
                                        ;ERROR 17
003666  060236                                EM18            ;MSG B0 ERROR
003670  064354                                DH8             ;IN UNLD
003672  067054                                DT13
003674  067454                                DF20
                    2064
                                        ;ERROR 20
003676  060251                                EM19            ;MSG A1 ERROR
003700  064354                                DH8             ;IN UNLD
003702  067054                                DT13
003704  067454                                DF20
                    2069
                                        ;ERROR 21
003706  060264                                EM20            ;MSG B1 ERROR
003710  064354                                DH8             ;IN UNLD
003712  067054                                DT13
003714  067454                                DF20
                    2074
                                        ;ERROR 22
003716  061614                                EM46            ;MSG A2 ERROR
003720  064354                                DH8             ;IN UNLD
```

```
2077  003722  067114                DT14
2078  003724  067520                DF22
2079                          ;ERROR 23
2080  003726  061627                EM47          ;MSG B2 ERROR
2081  003730  064354                DH8           ;IN UNLD
2082  003732  067114                DT14
2083  003734  067520                DF22
2084
2085                          ;ERROR 24
2086  003736  060277                EM21          ;CERR SET
2087  003740  065043                DH21          ;AFTER SCLR
2088  003742  066606                DT1
2089  003744  067334                DF10
2090                          ;ERROR 25
2091  003746  060321                EM22          ;RLS DID NOT SET CERR
2092  003750  064066                DH1
2093  003752  066606                DT1
2094  003754  067240                DF1
2095
2096                          ;ERROR 26
2097  003756  060360                EM23          ;SACK SET AFTER RLS SENT
2098  003760  064066                DH1
2099  003762  066606                DT1
2100  003764  067240                DF1
2101
2102                          ;ERROR 27
2103  003766  060440                EM24          ;VOL VALID NOT SET
2104  003770  064775                DH19          ;AFTER PACK CMD
2105  003772  066606                DT1
2106  003774  067334                DF10
2107                          ;ERROR 30
2108  003776  060464                EM25          ;DRIVE TYPE SET IN MR2
2109  004000  064066                DH1
2110  004002  066606                DT1
2111  004004  067240                DF1
2112                          ;ERROR 31
2113  004006  060512                EM26          ;DDT SET IN RKDS
2114  004010  064066                DH1
2115  004012  066606                DT1
2116  004014  067240                DF1
2117                          ;ERROR  32
2118  004016  060532                EM27          ;DTYE SET IN RKER
2119  004020  064066                DH1
2120  004022  066606                DT1
2121  004024  067240                DF1
2122                          ;ERROR 33
2123  004026  060553                EM28          ;DTYE NOT SET IN RKER
2124  004030  064066                DH1
2125  004032  066606                DT1
2126  004034  067240                DF1
2127                          ;ERROR 34
2128  004036  060600                EM29          ;DTYE DID NOT SET CERR
2129  004040  064066                DH1
2130  004042  066606                DT1
2131  004044  067240                DF1
2132                          ;ERROR 35
```

ERROR POINTER TABLE

```
2133  004046  060647          EM30                    ;C-D PARITY ERROR SET IN MR3
2134  004050  064066          DH1
2135  004052  066606          DT1
2136  004054  067240          DF1
2137                    ;ERROR 36
2138  004056  060700          EM31                    ;D-C PARITY  SET IN CS1
2139  004060  064066          DH1
2140  004062  066606          DT1
2141  004064  067240          DF1
2142                    ;ERROR 37
2143  004066  060725          EM32                    ;FAULT NOT SET IN MR3
2144  004070  064066          DH1
2145  004072  066606          DT1
2146  004074  067240          DF1
2147                    ;ERROR 40
2148  004076  060752          EM33                    ;C-D PARITY ERROR NOT SET IN MR3
2149  004100  064066          DH1
2150  004102  066606          DT1
2151  004104  067240          DF1
2152                    ;ERROR 41
2153  004106  061007          EM34                    ;D-C PARITY  NOT SET IN CS1
2154  004110  064066          DH1
2155  004112  066606          DT1
2156  004114  067240          DF1
2157                    ;ERROR 42
2158  004116  061040          EM35                    ;DCPAR DID NOT SET CERR
2159  004120  064066          DH1
2160  004122  066606          DT1
2161  004124  067240          DF1
2162                    ;ERROR 43
2163  004126  061113          EM36                    ;CYL ADDR IN B2 NOT = RKDC
2164  004130  064642          DH14                    ;AFTER SEEK WITH BAD PARITY
2165  004132  067114          DT14
2166  004134  067520          DF22
2167                    ;ERROR 44
2168  004136  061153          EM37                    ;CYL DIFF IN A2 NOT=RKDC
2169  004140  064642          DH14
2170  004142  067114          DT14
2171  004144  067520          DF22
2172                    ;ERROR 45
2173  004146  061113          EM36                    ;CYL ADDR IN RKMR3 NOT=RKDC
2174  004150  064642          DH14
2175  004152  066652          DT4
2176  004154  067320          DF6
2177                    ;ERROR 46
2178  004156  061213          EM38                    ;CYL DIFF IN RKMR2 NOT=CALDIF
2179  004160  064642          DH14
2180  004162  066652          DT4
2181  004164  067320          DF6
2182                    ;ERROR 47
2183  004166  061264          EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2184  004170  064734          DH17                    ;AFTER RECAL CMD
2185  004172  067114          DT14
2186  004174  067520          DF22
2187                    ;ERROR 50
2188  004176  061321          EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
```

ERROR POINTER TABLE

```
2189  004200  064734                DH17                    ;AFTER RECAL COMD
2190  004202  067114                DT14
2191  004204  067520                DF22
2192                          ;ERROR 51
2193  004206  060223                EM17                    ;AD ERROR
2194  004210  065157                DH26                    ;AFTER READ DATA CMD
2195  004212  067054                DT13
2196  004214  067454                DF20
2197                          ;ERROR 52
2198  004216  060236                EM18                    ;BD ERROR
2199  004220  065157                DH26
2200  004222  067054                DT13
2201  004224  067454                DF20
2202                          ;ERROR 53
2203  004226  061410                EM43                    ;HEAD DECODE IN B3 NOT CLEARED
2204  004230  064734                DH17                    ;AFTER RECAL CMD
2205  004232  067164                DT15
2206  004234  067554                DF23
2207                          ;ERROR 54
2208  004236  061437                EM44                    ;B3 HEAD DECODE INCORRECT
2209  004240  064672                DH16
2210  004242  067164                DT15
2211  004244  067554                DF23
2212                          ;ERROR 55
2213  004246  060075                EM13                    ;NO ATTN
2214  004250  064734                DH17                    ;AFTER RECAL CMD
2215  004252  066606                DT1
2216  004254  067334                DF10
2217                          ;ERROR 56
2218  004256  062561                EM64                    ;MSG B3 HEAD REG NOT CLEARED
2219  004260  064354                DH8                     ;IN UNLOAD
2220  004262  067164                DT15
2221  004264  067554                DF23
2222                          ;ERROR 57
2223  004266  060223                EM17                    ;MSG AD ERROR
2224  004270  064400                DH9                     ;AFTER START SPINDLE CMD REC'D BY DRIVE
2225  004272  067054                DT13
2226  004274  067454                DF20
2227                          ;ERROR 60
2228  004276  060236                EM18                    ;MSG BD ERROR
2229  004300  064400                DH9
2230  004302  067054                DT13
2231  004304  067454                DF20
2232                          ;ERROR 61
2233  004306  060251                EM19                    ;MSG A1 ERROR
2234  004310  064400                DH9
2235  004312  067054                DT13
2236  004314  067454                DF20
2237                          ;ERROR 62
2238  004316  060264                EM20                    ;MSG B1 ERROR
2239  004320  064400                DH9
2240  004322  067054                DT13
2241  004324  067454                DF20
2242                          ;ERROR 63
2243  004326  060223                EM17
2244  004330  064444                DH10                    ;AT END OF HEAD LOADING
```

```
2245  004332  067054                  DT13
2246  004334  067454                  DF20
2247                          ;ERROR 64
2248  004336  060236                  EM18
2249  004340  064444                  DH10
2250  004342  067054                  DT13
2251  004344  067454                  DF20
2252                          ;ERROR 65
2253  004346  060251                  EM19
2254  004350  064444                  DH10
2255  004352  067054                  DT13
2256  004354  067454                  DF20
2257                          ;ERROR 66
2258  004356  060264                  EM20
2259  004360  064444                  DH10
2260  004362  067054                  DT13
2261  004364  067454                  DF20
2262                          ;ERROR 67
2263  004366  060075                  EM13                    ;NO ATTN
2264  004370  064444                  DH10                    ;AT END OF HEAD LOADING.
2265  004372  066606                  DT1
2266  004374  067334                  DF10
2267                          ;ERROR 70
2268  004376  061743                  EM50                    ;FWD NOT SET WITHIN 60 SEC FROM
2269  004400  064066                  DH1                     ;START SPINDLE CMD.
2270  004402  066606                  DT1
2271  004404  067240                  DF1
2272                          ;ERROR 71
2273  004406  060223                  EM17
2274  004410  064473                  DH11                    ;AFTER START SPINDLE CMD & FWD SET.
2275  004412  067054                  DT13
2276  004414  067454                  DF20
2277                          ;ERROR 72
2278  004416  060236                  EM18
2279  004420  064473                  DH11
2280  004422  067054                  DT13
2281  004424  067454                  DF20
2282                          ;ERROR 73
2283  004426  060251                  EM19
2284  004430  064473                  DH11
2285  004432  067054                  DT13
2286  004434  067454                  DF20
2287                          ;ERROR 74
2288  004436  060264                  EM20
2289  004440  064473                  DH11
2290  004442  067054                  DT13
2291  004444  067454                  DF20
2292                          ;ERROR 75
2293  004446  062014                  EM51                    ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
2294  004450  064066                  DH1                     ;FROM START SPINDLE CMD.
2295  004452  066606                  DT1
2296  004454  067240                  DF1
2297                          ;ERROR 76
2298  004456  060223                  EM17
2299  004460  064532                  DH12                    ;AT INNER LIMIT FROM START SPINDLE CMD.
2300  004462  067054                  DT13
```

```
2301   004464   067454                          DF20
2302                                    ;ERROR 77
2303   004466   060236                          EM18
2304   004470   064532                          DH12
2305   004472   067054                          DT13
2306   004474   067454                          DF20
2307                                    ;ERROR 100
2308   004476   060251                          EM19
2309   004500   064532                          DH12
2310   004502   067054                          DT13
2311   004504   067454                          DF20
2312                                    ;ERROR 101
2313   004506   060264                          EM20
2314   004510   064532                          DH12
2315   004512   067054                          DT13
2316   004514   067454                          DF20
2317                                    ;ERROR 102
2318   004516   061655                          EM49            ;FWD NOT SET WITHIN 4 SEC IN RTZ PORTION
2319   004520   064066                          DH1             ;OF START SPIN CMD.
2320   004522   066606                          DT1
2321   004524   067240                          DF1
2322                                    ;ERROR 103
2323   004526   060223                          EM17
2324   004530   064573                          DH13            ;FROM OUTER LIMIT TO CYL 0 DURING LOADING
2325   004532   067054                          DT13
2326   004534   067454                          DF20
2327                                    ;ERROR 104
2328   004536   060236                          EM18
2329   004540   064573                          DH13
2330   004542   067054                          DT13
2331   004544   067454                          DF20
2332                                    ;ERROR 105
2333   004546   060251                          EM19
2334   004550   064573                          DH13
2335   004552   067054                          DT13
2336   004554   067454                          DF20
2337                                    ;ERROR 106
2338   004556   060264                          EM20
2339   004560   064573                          DH13
2340   004562   067054                          DT13
2341   004564   067454                          DF20
2342                                    ;ERROR 107
2343   004566   061473                          EM45            ;DRIVE READY NOT SET WITHIN 1 SEC
2344   004570   064066                          DH1             ;FROM FWD IN RTZ PORTION OF START SPIN CMD.
2345   004572   066606                          DT1
2346   004574   067240                          DF1
2347                                    ;ERROR 110
2348   004576   060223                          EM17            ;MSG A0 ERROR
2349   004600   064642                          DH14            ;AFTER SEEK WITH BAD PARITY
2350   004602   067114                          DT14
2351   004604   067520                          DF22
2352                                    ;ERROR 111
2353   004606   060236                          EM18            ;MSG B0 ERROR
2354   004610   064642                          DH14            ;AFTER SEEK WITH BAD PARITY
2355   004612   067114                          DT14
2356   004614   067520                          DF22
```

```
2357
2358  004616  060251          ;ERROR 112
2359  004620  065157                EM19                    ;A1 ERROR
2360  004622  067054                DH26                    ;AFTER READ DATA CMD
2361  004624  067454                DT13
2362                           DF20
2363  004626  060264          ;ERROR 113
2364  004630  065157                EM20                    ;B1 ERROR
2365  004632  067054                DH26
2366  004634  067454                DT13
2367                           DF20
2368  004636  060223          ;ERROR 114
2369  004640  064672                EM17
2370  004642  067054                DH16                    ;AFTER LOADING HEAD REGISTER & SEEK CMD
2371  004644  067454                DT13
2372                           DF20
2373  004646  060236          ;ERROR 115
2374  004650  064672                EM18
2375  004652  067054                DH16
2376  004654  067454                DT13
2377                           DF20
2378  004656  060044          ;ERROR 116
2379  004660  064775                EM12                    ;CONT NOT RDY
2380  004662  066606                DH19                    ;AFTER PACK CMD
2381  004664  067334                DT1
2382                           DF10
2383  004666  060044          ;ERROR 117
2384  004670  065014                EM12                    ;CONT NOT RDY
2385  004672  066606                DH20                    ;AFTER SEL DR CMD
2386  004674  067334                DT1
2387                           DF10
2388  004676  060044          ;ERROR 120
2389  004700  065043                EM12
2390  004702  066606                DH21                    ;AFTER SUBSYS CLEAR
2391  004704  067334                DT1
2392                           DF10
2393  004706  060044          ;ERROR 121
2394  004710  064400                EM12
2395  004712  066606                DH9                     ;AFTER START SPINDLE CMD
2396  004714  067334                DT1
2397                           DF10
2398  004716  060044          ;ERROR 122
2399  004720  064642                EM12
2400  004722  066606                DH14                    ;AFTER SEEK WITH BAD PARITY
2401  004724  067334                DT1
2402                           DF10
2403  004726  063672          ;ERROR 123
2404  004730  064066                EM88                    ;NO DRIVES FOUND
2405  004732  066606                DH1
2406  004734  067240                DT1
2407                           DF1
2408  004736  060044          ;ERROR 124
2409  004740  064734                EM12
2410  004742  066606                DH17                    ;AFTER RECAL CMD
2411  004744  067334                DT1
2412                           DF10
                               ;ERROR 125
```

```
2413  004746  060075                  EM13            ;NO ATTN
2414  004750  064642                  DH14            ;FROM SEEK WITH BAD PARITY
2415  004752  066606                  DT1
2416  004754  067334                  DF10
2417                          ;ERROR 126
2418  004756  063763                  EM89            ;NO DRVS FOUND IN DEVICE MAP
2419  004760  064066                  DH1
2420  004762  066606                  DT1
2421  004764  067240                  DF1
2422                          ;ERROR 127
2423  004766  061264                  EM39            ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2424  004770  065043                  DH21            ;AFTER SCLR
2425  004772  066606                  DT1
2426  004774  067334                  DF10
2427                          ;ERROR 130
2428  004776  061321                  EM40            ;CYL ADDR IN RKMR3 NOT CLEARED
2429  005000  065043                  DH21
2430  005002  066606                  DT1
2431  005004  067334                  DF10
2432                          ;ERROR 131
2433  005006  060044                  EM12            ;NO RDY
2434  005010  065140                  DH25            ;AFTER SEEK CMD
2435  005012  066606                  DT1
2436  005014  067334                  DF10
2437                          ;ERROR 132
2438  005016  060075                  EM13            ;NO ATTN
2439  005020  065140                  DH25
2440  005022  066606                  DT1
2441  005024  067334                  DF10
2442                          ;ERROR 133
2443  005026  060223                  EM17            ;MSG A0 ERROR
2444  005030  065140                  DH25
2445  005032  067054                  DT13
2446  005034  067454                  DF20
2447                          ;ERROR 134
2448  005036  060236                  EM18            ;MSG B0 ERROR
2449  005040  065140                  DH25
2450  005042  067054                  DT13
2451  005044  067454                  DF20
2452                          ;ERROR 135
2453  005046  060251                  EM19            ;MSG A1 ERROR
2454  005050  065140                  DH25
2455  005052  067054                  DT13
2456  005054  067454                  DF20
2457                          ;ERROR 136
2458  005056  060264                  EM20            ;MSG B1 ERROR.
2459  005060  065140                  DH25
2460  005062  067054                  DT13
2461  005064  067454                  DF20
2462                          ;ERROR 137
2463  005066  061264                  EM39            ;CYL DIFF/OFFSET IN A2 NOT CLEARED
2464  005070  065140                  DH25
2465  005072  067114                  DT14
2466  005074  067520                  DF22
2467                          ;ERROR 140
2468  005076  061321                  EM40            ;CYL ADDR IN B2 NOT CLEARED
```

```
2469  005100  065140                    DH25
2470  005102  067114                    DT14
2471  005104  067520                    DF22
2472                            ;ERROR 141
2473  005106  062114                    EM52          ;20 SECTOR FORMAT NOT SET IN RKMR2
2474  005110  064066                    DH1
2475  005112  066606                    DT1
2476  005114  067240                    DF1
2477                            ;ERROR 142
2478  005116  062153                    EM53          ;SECTOR 0 NOT FOUND WITHIN 50 MS
2479  005120  064066                    DH1
2480  005122  066606                    DT1
2481  005124  067240                    DF1
2482                            ;ERROR 143
2483  005126  062204                    EM54          ;DIFF SECTOR NOT FOUND WITHIN 3MS
2484  005130  064[illegible]             DH1
2485  005132  066[illegible]             DT1
2486  005134  067[illegible]             DF1
2487                            ;ERROR 144
2488  005136  06164[illegible]           EM48          ;MSG B3 ERROR
2489  005140  065502                    DH34          ;SECTOR REG UNSTABLE
2490  005142  066606                    DT1
2491  005144  067334                    DF10
```

ERROR POINTER TABLE

```
2492
2493  005146  061642          ;ERROR 145
2494  005150  065526                EM48
2495  005152  066714                DH35          ;BETWEEN SECTOR COUNTS
2496  005154  067350                DT6
2497                           DF12
2498  005156  060251          ;ERROR 146
2499  005160  064642                EM19          ;MSG A1 ERROR
2500  005162  067114                DH14          ;AFTER SEEK WITH BAD PARITY
2501  005164  067520                DT14
2502                           DF22
2503  005166  060264          ;ERROR 147
2504  005170  064642                EM20          ;MSG B1 ERROR
2505  005172  067114                DH14
2506  005174  067520                DT14
2507                           DF22
2508  005176  060251          ;ERROR 150
2509  005200  065636                EM19          ;MSG A1 ERROR
2510  005202  066606                DH37
2511  005204  067334                DT1
2512                           DF10
2513  005206  060044          ;ERROR 151
2514  005210  065071                EM12          ;NO RDY
2515  005212  066606                DH22          ;AFTER CLEAR CMD
2516  005214  067334                DT1
2517                           DF10
2518  005216  000000          ;ERROR 152
2519  005220  000000                0
2520  005222  000000                0
2521  005224  000000                0
2522                           0
2523  005226  000000          ;ERROR 153
2524  005230  000000                0
2525  005232  000000                0
2526  005234  000000                0
2527                           0
2528  005236  062237          ;ERROR 154
2529  005240  065071                EM55          ;ATTN NOT CLEARED
2530  005242  066606                DH22
2531  005244  067334                DT1
2532                           DF10
2533  005246  063527          ;ERROR 155
2534  005250  064734                EM85          ;IDAE NOT CLEARED
2535  005252  066606                DH17          ;AFTER RECAL CMD
2536  005254  067334                DT1
2537                           DF10
2538  005256  060044          ;ERROR 156
2539  005260  066436                EM12          ;CONT NOT READY
2540  005262  066606                DH51          ;AFTER SEEK TO SELF
2541  005264  067334                DT1
2542                           DF10
2543  005266  060075          ;ERROR 157
2544  005270  066436                EM13          ;NO ATTN
2545  005272  066606                DH51
2546  005274  067334                DT1
2547                           DF10
                               ;ERROR 160
```

```
2548  005276  062417                EM59            ;LIMIT DETECT NOT FOUND
2549  005300  064066                DH1
2550  005302  066606                DT1
2551  005304  067240                DF1
2552                          ;ERROR 161
2553  005306  060223                EM17            ;MSG A0 ERROR
2554  005310  065702                DH38            ;AFTER LIMIT DETECT
2555  005312  067054                DT13
2556  005314  067454                DF20
2557                          ;ERROR 162
2558  005316  060236                EM18            ;MSG B0 ERROR
2559  005320  065702                DH38
2560  005322  067054                DT13
2561  005324  067454                DF20
2562                          ;ERROR 163
2563  005326  060251                EM19            ;MSG A1 ERROR
2564  005330  065702                DH38
2565  005332  067054                DT13
2566  005334  067454                DF20
2567                          ;ERROR 164
2568  005336  060264                EM20            ;MSG B1 ERROR
2569  005340  065702                DH38
2570  005342  067054                DT13
2571  005344  067454                DF20
2572                          ;ERROR 165
2573  005346  060075                EM13            ;NO ATTN
2574  005350  065702                DH38
2575  005352  066606                DT1
2576  005354  067334                DF10
2577                          ;ERROR 166
2578  005356  062450                EM60            ;HEADS HOME NOT FOUND
2579  005360  065702                DH38
2580  005362  066606                DT1
2581  005364  067334                DF10
2582                          ;ERROR 167
2583  005366  062504                EM61            ;LOAD HEADS NOT FOUND
2584  005370  065702                DH38
2585  005372  066606                DT1
2586  005374  067334                DF10
2587                          ;ERROR 170
2588  005376  063036                EM72            ;FORMAT TEST BYPASSED
2589  005400  066155                DH45            ;LIMIT DETECT ERROR ON PREVIOUS TEST
2590  005402  066606                DT1
2591  005404  067424                DF16
2592                          ;ERROR 171
2593  005406  060044                EM12            ;NO RDY
2594  005410  065345                DH30            ;AFTER READ HEADER CMD
2595  005412  066606                DT1
2596  005414  067334                DF10
2597                          ;ERROR 172
2598  005416  061264                EM39            ;CYL DIFF/OFFSET NOT CLEARED
2599  005420  065345                DH30            ;AFTER READ HEADER CMD
2600  005422  067114                DT14
2601  005424  067520                DF22
2602                          ;ERROR 173
2603  005426  062540                EM63            ;DLT SET
```

```
 2604  005430  065345                    DH30
 2605  005432  066672                    DT5
 2606  005434  067410                    DF15
 2607                          ;ERROR 174
 2608  005436  060277                    EM21                    ;CERR SET
 2609  005440  065345                    DH30
 2610  005442  066672                    DT5
 2611  005444  067410                    DF15
 2612                          ;ERROR 175
 2613  005446  061264                    EM39                    ;CYL DIFF NOT CLEARED
 2614  005450  064444                    DH10                    ;AT END OF HEAD LOADING
```

```
005452  067114                  DT14
005454  067520                  DF22
                ;ERROR 176
005456  061321                  EM40            ;CYL ADDR NOT CLEARED.
005460  064444                  DH10
005462  067114                  DT14
005464  067520                  DF22
                ;ERROR 177
005466  063036                  EM72            ;FORMAT TEST BYPASSED
005470  066221                  DH46            ;COULD NOT READ BSE INFO
005472  066606                  DT1
005474  067424                  DF16
                ;ERROR 200
005476  060044                  EM12            ;NO RDY
005500  065720                  DH39            ;AFTER WRITE HEADER CMD
005502  066672                  DT5
005504  067364                  DF13
                ;ERROR 201
005506  060277                  EM21            ;CERR SET
005510  065720                  DH39
005512  066672                  DT5
005514  067364                  DF13
                ;ERROR 202
005516  062610                  EM65            ;READ HEADER ERROR
005520  064066                  DH1
005522  066734                  DT7
005524  067400                  DF14
                ;ERROR 203
005526  060223                  EM17            ;MSG A0 ERROR
005530  065462                  DH33            ;DURING SEEK CMD
005532  067054                  DT13
005534  067454                  DF20
                ;ERROR 204
005536  060236                  EM18            ;MSG B0 ERROR
005540  065462                  DH33
005542  067054                  DT13
005544  067454                  DF20
                ;ERROR 205
005546  060251                  EM19            ;MSG A1 ERROR
005550  065462                  DH33
005552  067054                  DT13
005554  067454                  DF20
                ;ERROR 206
005556  060264                  EM20            ;MSG B1 ERROR
005560  065462                  DH33
005562  067054                  DT13
005564  067454                  DF20
                ;ERROR 207
005566  061113                  EM36            ;CYL ADDR IN RKMR3 INCORRECT
005570  065140                  DH25            ;AFTER SEEK CMD
005572  066652                  DT4
005574  067320                  DF6
                ;ERROR 210
005576  060277                  EM21            ;CERR SET
005600  065140                  DH25
005602  066606                  DT1
```

```
2671  005604  067334                    DF10
2672
2673  005606  062664                ;ERROR 211
2674  005610  065140                    EM67                ;READ CYL 0 HEADERS ON CYL 1
2675  005612  066606                    DH25
2676  005614  067334                    DT1
2677                                    DF10
2678  005616  061213                ;ERROR 212
2679  005620  065462                    EM38                ;CYL DIFF IN RKMR2 NOT = CALDIF
2680  005622  066652                    DH33                ;DURING SEEK CMD
2681  005624  067320                    DT4
2682                                    DF6
2683  005626  060223                ;ERROR 213
2684  005630  066024                    EM17                ;MSG AD ERROR
2685  005632  067054                    DH41                ;DURING RECAL CMD
2686  005634  067454                    DT13
2687                                    DF20
2688  005636  060236                ;ERROR 214
2689  005640  066024                    EM18                ;MSG BD ERROR
2690  005642  067054                    DH41
2691  005644  067454                    DT13
2692                                    DF20
2693  005646  060251                ;ERROR 215
2694  005650  066024                    EM19                ;MSG A1 ERROR
2695  005652  067054                    DH41
2696  005654  067454                    DT13
2697                                    DF20
2698  005656  060264                ;ERROR 216
2699  005660  066024                    EM20                ;MSG B1 ERROR
2700  005662  067054                    DH41
2701  005664  067454                    DT13
2702                                    DF20
2703  005666  061213                ;ERROR 217
2704  005670  066024                    EM38                ;CYL DIFF IN RKMR2 NOT=CALDIF
2705  005672  066652                    DH41
2706  005674  067320                    DT4
2707                                    DF6
2708  005676  060277                ;ERROR 220
2709  005700  064734                    EM21                ;CERR SET
2710  005702  066606                    DH17                ;AFTER RECAL CMD
2711  005704  067334                    DT1
2712                                    DF10
2713  005706  060223                ;ERROR 221
2714  005710  064734                    EM17                ;MSG AD ERROR
2715  005712  067054                    DH17
2716  005714  067454                    DT13
2717                                    DF20
2718  005716  060251                ;ERROR 222
2719  005720  064734                    EM19                ;MSG A1 ERROR
2720  005722  067054                    DH17
2721  005724  067454                    DT13
2722                                    DF20
2723  005726  061264                ;ERROR 223
2724  005730  064734                    EM39                ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2725  005732  066606                    DH17
2726  005734  067334                    DT1
                                        DF10
```

```
2727
2728  005736  062630          ;ERROR 224
2729  005740  064734                EM66                    ;CYL ADDR IN RKMR3 INCORRECT
2730  005742  066606                DH17
2731  005744  067334                DT1
2732                                 DF10
2733  005746  062723          ;ERROR 225
2734  005750  064734                EM68                    ;READING CYL 1 HEADERS ON CYL 0
2735  005752  066606                DH17
2736  005754  067334                DT1
2737                                 DF10
2738  005756  060044          ;ERROR 226
2739  005760  065157                EM12                    ;NO RDY
2740  005762  066606                DH26                    ;AFTER READ DATA CMD
2741  005764  067334                DT1
2742                                 DF10
2743  005766  060277          ;ERROR 227
2744  005770  065157                EM21                    ;CERR SET
2745  005772  066672                DH26
2746  005774  067410                DT5
2747                                 DF15
2748  005776  063645          ;ERROR 230
2749  006000  066536                EM87                    ;CANT READ BSE INFO
2750  006002  066606                DH53                    ;ON SECT 10,12,14,16,18,20
2751  006004  067434                DT1
2752                                 DF17
2753  006006  000000          ;ERROR 231
2754  006010  000000                0
2755  006012  000000                0
2756  006014  000000                0
2757                                 0
2758  006016  061113          ;ERROR 232
2759  006020  065140                EM36                    ;CYL ADDR IN RKMR3 NOT=RKDC
2760  006022  066630                DH25                    ;AFTER SEEK CMD
2761  006024  067304                DT3
2762                                 DF5
2763  006026  063645          ;ERROR 233
2764  006030  066045                EM87                    ;CANNOT READ BSE INFO
2765  006032  066606                DH42                    ;ON SECT 0,2,4,6,8
2766  006034  067434                DT1
2767                                 DF17
2768  006036  000000          ;ERROR 234
2769  006040  000000                0
2770  006042  000000                0
2771  006044  000000                0
2772                                 0
2773  006046  062762          ;ERROR 235
2774  006050  066105                EM69                    ;ALIGN CARTRIDGE USED
2775  006052  066606                DH44                    ;WILL BYPASS FORMAT & ALL R/W TESTS
2776  006054  067334                DT1
2777                                 DF10
2778  006056  062272          ;ERROR 236
2779  006060  065117                EM56                    ;UNEXP MEM PARITY TRAP
2780  006062  067030                DH23                    ;TEST #, TRAP PC
2781  006064  067270                DT11
2782                                 DF3
                               ;ERROR 237
```

ERROR POINTER TABLE

```
2783  006066  063015                EM71                    ;DSC SET
2784  006070  065071                DH22                    ;AFTER DRIVE CLEAR CMD
2785  006072  066606                DT1
2786  006074  067334                DF10
2787                          ;ERROR 240
2788  006076  062723                EM68                    ;READ CYL 1 HEADERS ON CYL 0
2789  006100  064734                DH17                    ;AFTER RECAL CMD
2790  006102  066606                DT1
2791  006104  067334                DF10
2792                          ;ERROR 241
2793  006106  061113                EM36                    ;RKMR3 NOT = RKDC
2794  006110  064642                DH14                    ;AFTER SEEK WITH BAD PARITY
2795  006112  066754                DT8
2796  006114  067320                DF6
2797                          ;ERROR 242
2798  006116  061213                EM38                    ;CYL DIFF IN RKMR2 INCORRECT
2799  006120  064642                DH14
2800  006122  066754                DT8
2801  006124  067320                DF6
2802
2803                          ;ERROR 243
2804  006126  061113                EM36                    ;CYL ADDR IN RKMR3 INCORRECT
2805  006130  065140                DH25                    ;AFTER SEEK CMD
2806  006132  066754                DT8
2807  006134  067320                DF6
2808                          ;ERROR 244
2809  006136  063104                EM74                    ;RTZ NOT SET
2810  006140  066024                DH41                    ;DURING RECAL CMD
2811  006142  066606                DT1
2812  006144  067334                DF10
2813                          ;ERROR 245
2814  006146  060075                EM13                    ;NO ATTN
2815  006150  066330                DH48                    ;AFTER SEEK TO INVALID CYL
2816  006152  066606                DT1
2817  006154  067334                DF10
2818                          ;ERROR 246
2819  006156  063131                EM75                    ;IDAE NOT SET
2820  006160  066330                DH48
2821  006162  066652                DT4
2822  006164  067320                DF6
2823                          ;ERROR 247
2824  006166  060725                EM32                    ;FAULT NOT SET
2825  006170  066330                DH48
2826  006172  066652                DT4
2827  006174  067320                DF6
2828                          ;ERROR 250
2829  006176  063157                EM76                    ;PIP SET
2830  006200  066330                DH48
2831  006202  066652                DT4
2832  006204  067320                DF6
2833                          ;ERROR 251
2834  006206  060176                EM16                    ;DSC NOT SET
2835  006210  066330                DH48
2836  006212  066652                DT4
2837  006214  067320                DF6
2838                          ;ERROR 252
```

```
2839  006216  060223                EM17                    ;MSG A0 ERROR
2840  006220  066330                DH48
2841  006222  067054                DT13
2842  006224  067454                DF20
2843                          ;ERROR 253
2844  006226  060236                EM18                    ;MSG B0 ERROR
2845  006230  066330                DH48
2846  006232  067054                DT13
2847  006234  067454                DF20
2848                          ;ERROR 254
2849  006236  060251                EM19                    ;MSG A1 ERROR
2850  006240  066330                DH48
2851  006242  067054                DT13
2852  006244  067454                DF20
2853                          ;ERROR 255
2854  006246  060264                EM20                    ;MSG B1 ERROR
2855  006250  066330                DH48
2856  006252  067054                DT13
2857  006254  067454                DF20
2858                          ;ERROR 256
2859  006256  061213                EM38                    ;CYL DIFF IN RKMR2 NOT='CYL DIF'
2860  006260  066330                DH48
2861  006262  066652                DT4
2862  006264  067320                DF6
2863                          ;ERROR 257
2864  006266  061113                EM36                    ;CYL ADDR IN RKMR3 NOT=RKDC
2865  006270  066330                DH48
2866  006272  066652                DT4
2867  006274  067320                DF6
2868                          ;ERROR 260
2869  006276  000000                0
2870  006300  000000                0
2871  006302  000000                0
2872  006304  000000                0
2873                          ;ERROR 261
2874  006306  000000                0
2875  006310  000000                0
2876  006312  000000                0
2877  006314  000000                0
2878                          ;ERROR 262
2879  006316  063200                EM77                    ;FAULT NOT CLEARED
2880  006320  065071                DH22                    ;AFTER DRIVE CLEAR CMD
2881  006322  066606                DT1
2882  006324  067334                DF10
2883                          ;ERROR 263
2884  006326  063226                EM78                    ;CYL DIFF IN RKMR2 NOT=1 IN SEEK TO SELF
2885  006330  064642                DH14                    ;AFTER SEEK WITH BAD PARITY
2886  006332  066754                DT8
2887  006334  067320                DF6
2888                          ;ERROR 264
2889  006336  061347                EM41                    ;CYL ADDR NOT CLEARED
2890  006340  065345                DH30                    ;AFTER READ HEADER CMD
2891  006342  067114                DT14
2892  006344  067520                DF22
2893                          ;ERROR 265
2894  006346  060236                EM18                    ;MSG B0 ERROR
```

```
 2895  006350  065071                    DH22                    ;AFTER DRIVE CLEAR CMD
 2896  006352  067054                    DT13
 2897  006354  067454                    DF20
 2898                            ;ERROR 266
 2899  006356  060264                    EM20                    ;MSG B1 ERROR
 2900  006360  065071                    DH22
 2901  006362  067054                    DT13
 2902  006364  067454                    DF20
 2903                            ;ERROR 267
 2904  006366  060236                    EM18
 2905  006370  065720                    DH39                    ;AFTER WRITE HEADER CMD
 2906  006372  067054                    DT13
 2907  006374  067454                    DF20
 2908                            ;ERROR 270
 2909  006376  060264                    EM20
 2910  006400  065720                    DH39
 2911  006402  067054                    DT13
 2912  006404  067454                    DF20
 2913                            ;ERROR 271
 2914  006406  060236                    EM18
 2915  006410  065345                    DH30                    ;AFTER READ HEADER CMD
 2916  006412  067054                    DT13
 2917  006414  067454                    DF20
 2918                            ;ERROR 272
 2919  006416  060264                    EM20
 2920  006420  065345                    DH30
 2921  006422  067054                    DT13
 2922  006424  067454                    DF20
 2923                            ;ERROR 273
 2924  006426  060223                    EM17                    ;MSG A0 ERROR
 2925  006430  065071                    DH22                    ;AFTER DRV CLR CMD
 2926  006432  067054                    DT13
 2927  006434  067454                    DF20
 2928                            ;ERROR 274
 2929  006436  060251                    EM19                    ;MSG A1 ERROR
 2930  006440  065071                    DH22
 2931  006442  067054                    DT13
 2932  006444  067454                    DF20
 2933                            ;ERROR 275
 2934  006446  060236                    EM18                    ;MSG B0 ERROR
 2935  006450  064734                    DH17                    ;AFTER RECAL CMD
 2936  006452  067054                    DT13
 2937  006454  067454                    DF20
 2938                            ;ERROR 276
 2939  006456  060264                    EM20                    ;MSG B1 ERROR
 2940  006460  064734                    DH17
 2941  006462  067054                    DT13
 2942  006464  067454                    DF20
 2943                            ;ERROR 277
 2944  006466  060223                    EM17                    ;MSG A0 ERROR
 2945  006470  065720                    DH39                    ;AFTER WRITE HEADER CMD
 2946  006472  067054                    DT13
 2947  006474  067454                    DF20
 2948                            ;ERROR 300
 2949  006476  060251                    EM19                    ;MSG A1 ERROR
 2950  006500  065720                    DH39
```

```
  2951  006502  067054                    DT13
  2952  006504  067454                    DF20
  2953                            ;ERROR 301
  2954  006506  060223                    EM17
  2955  006510  065345                    DH30            ;AFTER READ HEADER CMD
  2956  006512  067054                    DT13
  2957  006514  067454                    DF20
  2958                            ;ERROR 302
  2959  006516  060251                    EM19
  2960  006520  065345                    DH30
  2961  006522  067054                    DT13
  2962  006524  067454                    DF20
  2963                            ;ERROR 303
  2964  006526  061264                    EM39            ;CYL DIFF/OFFSET NOT CLEARED
  2965  006530  065720                    DH39            ;AFTER WRITE HEADER CMD
  2966  006532  067114                    DT14
  2967  006534  067520                    DF22
  2968                            ;ERROR 304
  2969  006536  061347                    EM41            ;CYL ADDR NOT CLEARED
  2970  006540  065720                    DH39
  2971  006542  067114                    DT14
  2972  006544  067520                    DF22
  2973
  2974                            ;ERROR 305
  2975  006546  063334                    EM80            ;UNLD NOT SET
  2976  006550  065702                    DH38            ;AFTER LIMIT DETECT
  2977  006552  066606                    DT1
  2978  006554  067334                    DF10
  2979                            ;ERROR 306
  2980  006556  063364                    EM81            ;SPIN NOT SET
  2981  006560  064400                    DH9             ;AFTER START SPIN CMD.
  2982  006562  066606                    DT1
  2983  006564  067334                    DF10
  2984                            ;ERROR 307
  2985  006566  063412                    EM82            ;RTZ NOT SET
  2986  006570  066024                    DH41            ;DURING RECAL CMD
  2987  006572  066606                    DT1
  2988  006574  067334                    DF10
  2989                            ;ERROR 310
  2990  006576  063437                    EM83            ;READ HEADER ERROR
  2991  006600  064066                    DH1
  2992  006602  066774                    DT9
  2993  006604  067510                    DF21
  2994                            ;ERROR 311
  2995  006606  063437                    EM83
  2996  006610  064066                    DH1
  2997  006612  067012                    DT10
  2998  006614  067510                    DF21
  2999                            ;ERROR 312
  3000  006616  063477                    EM84            ;FORMAT NOT SET
  3001  006620  065720                    DH39            ;AFTER WRITE HEADER CMD
  3002  006622  066606                    DT1
  3003  006624  067334                    DF10
  3004                            ;ERROR 313
  3005  006626  063477                    EM84
  3006  006630  065345                    DH30            ;AFTER READ HEADER CMD
```

```
  3007  006632  066606                    DT1
  3008  006634  067334                    DF10
  3009                            ;ERROR 314
  3010  006636  062315                    EM57          ;WCE AT CYL 411,TRK 2,SEC 21
  3011  006640  064066                    DH1
  3012  006642  067034                    DT12
  3013  006644  067274                    DF4
  3014                            ;ERROR 315
  3015  006646  062351                    EM58          ;SPOK NOT CLEARED
  3016  006650  064754                    DH18          ;AFTER UNLD CMD
  3017  006652  066606                    DT1
  3018  006654  067334                    DF10
  3019                            ;ERROR 316
  3020  006656  063002                    EM70          ;UNEXP ATTN
  3021  006660  064400                    DH9           ;AFTER START SPIN CMD
  3022  006662  066606                    DT1
  3023  006664  067334                    DF10
  3024                            ;ERROR 317
  3025  006666  063002                    EM70
  3026  006670  064473                    DH11          ;AFT SPIN CMD &  FWD SET
  3027  006672  066606                    DT1
  3028  006674  067334                    DF10
  3029                            ;ERROR 320
  3030  006676  063002                    EM70
  3031  006700  064532                    DH12          ;AT INNER LIMIT FROM ST SPIN CMD
  3032  006702  066606                    DT1
  3033  006704  067334                    DF10
  3034                            ;ERROR 321
  3035  006706  063002                    EM70
  3036  006710  064573                    DH13          ;FROM OUTER LIM TO CYL 0 DURING LOADING
  3037  006712  066606                    DT1
  3038  006714  067334                    DF10
  3039                            ;ERROR 322
  3040  006716  060251                    EM19          ;MSG A1 ERROR
  3041  006720  064672                    DH16          ;AFT LD HEAD REG & SEEK
  3042  006722  067054                    DT13
  3043  006724  067454                    DF20
  3044                            ;ERROR 323
  3045  006726  060264                    EM20          ;MSG B1 ERROR.
  3046  006730  064672                    DH16
  3047  006732  067054                    DT13
  3048  006734  067454                    DF20
  3049                            ;ERROR 324
  3050  006736  061614                    EM46          ;MSG A2 ERROR
  3051  006740  064672                    DH16
  3052  006742  067114                    DT14
  3053  006744  067520                    DF22
  3054                            ;ERROR 325
  3055  006746  061627                    EM47          ;MSG B2 ERROR
  3056  006750  064672                    DH16
  3057  006752  067114                    DT14
  3058  006754  067520                    DF22
  3059                            ;ERROR 326
  3060  006756  063063                    EM73          ;CTO SET
  3061  006760  063560                    EM86          ;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
  3062  006762  066606                    DT1
```

```
  3063  006764  067250                        DF2
  3064                              ;ERROR 327
  3065  006766  063313                        EM79                    ;NED SET
  3066  006770  063560                        EM86
  3067  006772  066606                        DT1
  3068  006774  067250                        DF2
  3069
  3070                              ;ERROR 330
  3071  006776  057464                        EM5                     ;MDS SET
  3072  007000  063560                        EM86
  3073  007002  066606                        DT1
  3074  007004  067250                        DF2
```

```
3075
3076                                    .SBTTL  PROGRAM SETUP
3077
3078  007006  012737  000001  001336  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
3079  007014  005037  001340                  CLR     BYPT16
3080  007020  005037  001342                  CLR     MODTST
3081  007024  000450                          BR      PRGSRT          ;START PROGRAM
3082
3083  007026  005037  001336          BYT16:  CLR     PARAM
3084  007032  012737  000001  001340          MOV     #1,BYPT16       ;SET FLAG TO BYPASS TEST 16
3085  007040  005037  001342                  CLR     MODTST
3086  007044  000440                          BR      PRGSRT
3087
3088  007046  012737  000001  001336  BYT16A: MOV     #1,PARAM
3089  007054  012737  000001  001340          MOV     #1,BYPT16
3090  007062  005037  001342                  CLR     MODTST
3091  007066  000427                          BR      PRGSRT
3092
3093  007070  005037  001336          MDTST:  CLR     PARAM
3094  007074  005037  001340                  CLR     BYPT16
3095  007100  012737  000001  001342          MOV     #1,MODTST
3096  007106  000417                          BR      PRGSRT
3097
3098  007110  005037  001336          MDTSTA: CLR     PARAM
3099  007114  012737  000001  001340          MOV     #1,BYPT16
3100  007122  012737  000001  001342          MOV     #1,MODTST
3101  007130  000406                          BR      PRGSRT
3102  007132  105037  001336          START:  CLRB    PARAM           ;CLEAR FOR 200 START
3103  007136  005037  001340                  CLR     BYPT16
3104  007142  005037  001342                  CLR     MODTST
3105  007146  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
3106  007150  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3107  007154  012746  000000                  MOV     #PRO,-(SP)      ;PSW LOADED TO BE
3108  007160  012746  007166                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
```

```
                                        RTI                     ;ENABLE ALL INTERRORUPTS

007166  004737  052556          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
                                                                ;& TURN ON KB INTERRORUPT



                                        ;*** CPU PRIORITY LEVEL NOW AT 0 ***
                                        ;*** ANY DEVICE WHICH SETS ITS   ***
                                        ;*** INTERRORUPT ENABLE BIT WILL   ***
                                        ;*** SERVICED.                    ***

                                        ;CLOCK INTERRORUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
                                        ;RK06 CONTROLLER INTERRORUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
                                        ;KEYBOARD INTERRORUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)

                                        ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)

                                                                ;SYSMAC 'SETUP'
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
007172  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
007176  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
007200  022706  001140                  CMP     #SWR,R6 ;;DONE?
007204  001374                          BNE     .-6             ;;LOOP BACK IF NO
007206  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
007212  012737  050664  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
007220  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
007226  012737  051144  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
007234  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
007242  012737  054662  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
007250  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
007256  012737  050576  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
007264  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
007272  013737  043552  043544          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
007300  005037  001174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
007304  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
007310  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
007316  012737  007316  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
007324  012737  007324  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
007332  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
007336  012737  007372  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
007344  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
007352  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
007360  022777  177777  171552          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
007366  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
007370  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
007372  012716  007400          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
007376  000002                          RTI
007400  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
007406  012737  000174  001142          MOV     #DISPREG,DISPLAY
007414  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
```

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

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

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

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


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


                                ;
                                ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
                                ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
                                ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
                                ;       EX:     DRIVES TO BE TESTED:  1,2,4<CR>
                                ;


                                ST2:    TST     PARAM
                                        BNE     1$                      ;BR IF 220 START
                                        JMP     ST4                     ;200 START, DEFAULT & SIZE THE BUSS
                                1$:     TYPE    ,MSG3                   ;DRIVES TO BE TESTED
                                        JSR     PC,GDRVS                ;GET DR NOS.
                                        TYPE    ,MSG4                   ;BUSS ADDR
                                        JSR     PC,GBA                  ;GET BA
                                        TYPE    ,MSG5                   ;CONT INT VECTOR
```

```
3221  007654  004737  044126            JSR     PC,GINT         ;GET INT VECTOR
3222  007660  000427                    BR      ST5
3223
3224                                    ;
3225                                    ;AUTO MODE
3226                                    ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
3227                                    ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
3228                                    ;ON THE BUSS
3229                                    ;
3230
3231  007662  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
3232  007670  001007                    BNE     1$
3233  007672  005237  003476            INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
3234  007676  004737  043660            JSR     PC,TITLE
3235  007702  104401  056201            TYPE    ,MSG7           ;DRO NOT TSTD
3236  007706  000402                    BR      ST4
3237  007710  005237  003450        1$: INC     ACT11           ;SET ACT AUTO FLAG.
3238
3239  007714  012737  177440  001264  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
3240  007722  012737  000210  001314          MOV     #210,RKVEC      ;DEFAULT VALUE
3241  007730  004737  044160            JSR     PC,SETINT
3242  007734  005237  003504            INC     SIZFLG          ;DO "SIZE THE BUSS" TEST
3243
```

```
3244  007740  005037  003316          ST5:    CLR     UNLD            ;INITIALIZE FLAGS
3245  007744  005037  003320                  CLR     BADHDR                  ;USED IN 'STOP ROUTINE
3246  007750  005037  003322                  CLR     HPEND                   ;FOR VALID PROGRAM HALTS
3247  007754  012737  003456  001346          MOV     #DRIVQ,DRVPTR   ;SETUP
3248  007762  005037  001220                  CLR     $DEVCT          ;NO. OF DRVS DONE
3249  007766  005037  001222                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
3250  007772  012737  010040  000004          MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
3251  010000  005777  171322                  TST     @LKS            ;SEE IF L-CLOCK THERE
3252  010004  005237  003476                  INC     LCLKF           ;PRESENT, SET FLAG.
3253  010010  013700  001330                  MOV     LCVEC,R0        ;VECTOR ADDR
3254  010014  012737  010102  000004          MOV     #2$,ERRVEC
3255  010022  005777  171272                  TST     @PKS            ;SEE IF P-CLOCK THERE
3256  010026  005237  003500                  INC     PCLKF           ;PRESENT, SET FLAG
3257  010032  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3258  010036  000412                          BR      3$
3259
3260  010040  022626                  1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
3261  010042  012737  010106  000004          MOV     #4$,ERRVEC
3262  010050  005777  171244                  TST     @PKS            ;SEE IF P-CLOCK THERE
3263  010054  005237  003500                  INC     PCLKF           ;PRESENT, SET FLAG
3264  010060  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3265  010064  005237  003502          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
3266  010070  012720  047654                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
3267  010074  012710  000300                  MOV     #PR6,(R0)
3268  010100  000407                          BR      TST1            ;;GO TO NEXT TEST
3269
3270  010102  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
3271  010104  000767                          BR      3$
3272
3273  010106  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
3274  010110  005037  003502                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
3275  010114  104401  056440                  TYPE    ,MSG13          ;ALL TIMING TESTS BYPASSED
3276
3277
```

```
                                .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'
                                ;*
                                ;;*****************************************************************
010120  000004                TST1:   SCOPE
010122  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
010130  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
                                
010134  012746  000000                MOV     #PRO,-(SP)      ;RESET PSW TO PRIORITY 0
010140  012746  010146                MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
010144  000002                        RTI
010146                          5$:
                                
010146  012737  010264  000004        MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
010154  013705  001264                MOV     $BASE,R5        ;SETUP INDEX REG.
010160  005765  000000                TST     RKCS1(R5)       ;REFERENCE ALL THE
010164  005765  000010                TST     RKCS2(R5)       ;CONTROLLER REGISTERS
010170  005765  000002                TST     RKWC(R5)
010174  005765  000004                TST     RKBA(R5)
010200  005765  000006                TST     RKDA(R5)
010204  005765  000012                TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
010210  005765  000014                TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
010214  005765  000016                TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
```

```
3309  010220  005765  000020                TST     RKDC(R5)          ;TESTING SHOULD NOT PROCEED
3310  010224  005765  000024                TST     RKDB(R5)          ;UNTIL THIS IS REMEDIED.
3311  010230  005765  000026                TST     RKMR1(R5)
3312  010234  005765  000034                TST     RKMR2(R5)
3313  010240  005765  000036                TST     RKMR3(R5)
3314  010244  005765  000030                TST     RKECPS(R5)
3315  010250  005765  000032                TST     RKECPT(R5)
3316
3317  010254  012737  050426  000004        MOV     #BADTMO,ERRVEC    ;SETUP TIMEOUT HANDLER
3318  010262  000404                        BR      TST2              ;;GO TO NEXT TEST
3319
3320  010264  022626                1$:     CMP     (SP)+,(SP)+       ;RESTORE STACK POINTER
3321  010266  104007                        ERROR   7                 ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
3322  010270  000137  043516                JMP     $EOP1
3323
```

```
                                ;;*****************************************************************
                                ;*TEST 2          SIZE THE BUSS
                                ;*
                                ;*      THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                                ;*      EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                                ;*      MANUAL MODE.
                                ;*      EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
                                ;*      CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
                                ;*      DRIVE WILL BE TESTED.  IF SET, THE PROGRAM WILL BYPASS
                                ;*      TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                ;*      MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                ;*      DICATING THE OTHER PORT IS ACCESSED.
                                ;*
                                ;;*****************************************************************
010274  000004                  TST2:   SCOPE
010276  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
010304  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
010310  005237  001462                  INC     BYPCERR         ;DO NOT TEST CERR IN 'FRDY'

010314  132737  000200  001231          BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
010322  001002                          BNE     14$             ;BR IF YES
010324  000137  010440                  JMP     12$             ;ELSE DO NORM SIZING OR VERIFY

010330  104401  056322          14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
010334  005037  003454                  CLR     DRIVS           ;# OF DRIVES PRESENT
010340  005000                          CLR     R0              ;DRV ADDR
010342  012701  003456                  MOV     #DRIV0,R1       ;DRV FLAG
010346  013702  001266                  MOV     $DEVM,R2        ;APT DEVICE MAP

010352  032702  000001          15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
010356  001410                          BEQ     16$             ;BR IF NO
010360  005237  003454                  INC     DRIVS           ;ELSE INCR DRIVE COUNT
010364  005211                          INC     (R1)            ;& SET DRIVE PRESENT FLAG
010366  104401  001205                  TYPE    ,$CRLF
010372  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DRIVE #
010374  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
010376     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
010377     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

010400  005721          16$:    TST     (R1)+           ;ADV POINTER TO NEXT FLAG
010402  005200                  INC     R0              ;INC DRIVE #
010404  022700  000010          CMP     #8.,R0          ;ALL 8 TESTED?
010410  001402                  BEQ     17$             ;BR IF YES

010412  006002                  ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
010414  000756                  BR      15$             ;& TRY AGAIN

010416  005737  003454  17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
010422  001402                  BEQ     18$             ;BR IF NO
010424  000137  011356          JMP     NUDRV           ;ELSE EXIT TEST

010430  104126          18$:    ERROR   126             ;NO DRIVES FOUND IN $DEVM
010432  000000                  HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
010434  000137  007740          JMP     ST5             ;TO TRY AGAIN
```

```
3380
3381  010440  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3382  010446  013737  001414  003372          MOV     T10,TEMP1
3383  010454  004737  044176                  JSR     PC,FRDY         ;FIND RDY
3384  010460  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
3385  010462  005737  003504                  TST     SIZFLG
3386  010466  001562                          BEQ     TST3            ;;DO NOT SIZE, GOTO NEXT TEST
3387  010470  104401  056322                  TYPE    ,MSG10          ;WILL TEST DRIVES
3388  010474  005037  003454                  CLR     DRIVS           ;# OF DRIVES PRESENT
3389  010500  005000                          CLR     R0              ;DRV ADDR
3390  010502  012701  003456                  MOV     #DRIV0,R1       ;DRV FLAG
3391  010506                          1$:
3392  010506  104415                          SCOP1
3393  010510  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3394
3395  010514  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
3396  010522  013737  001414  003372          MOV     T10,TEMP1
3397  010530  004737  044176                  JSR     PC,FRDY         ;FIND RDY
3398  010534  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
3399  010536  010065  000010                  MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
3400  010542  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
3401  010550  013737  001414  003372          MOV     T10,TEMP1
3402  010556  004737  044176                  JSR     PC,FRDY         ;FIND RDY
3403  010562  104117                          ERROR   117             ;NO RDY AFTER SELECT DR. CMD
3404  010564  032737  100000  003334          BIT     #CERR,HCS1
3405  010572  001046                          BNE     2$
3406  010574  013737  003362  003372          MOV     HMR2,TEMP1
3407  010602  042737  177770  003372          BIC     #^C<DRVMSK>,TEMP1
3408  010610  020037  003372                  CMP     R0,TEMP1        ;S/B SAME
3409  010614  001016                          BNE     3$
3410  010616  005700                          TST     R0
3411  010620  001003                          BNE     4$
3412  010622  005737  003446                  TST     DDPCH           ;SEE IF XXDP CHAIN MODE
3413  010626  001014                          BNE     5$
3414  010630  005237  003454          4$:     INC     DRIVS           ;INC DRIVE COUNT.
3415  010634  005211                          INC     (R1)            ;SET DRIVE PRESENT FLAG
3416  010636  104401  001205                  TYPE    ,$CRLF
3417  010642  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3418                                                                  ;;TYPE DR #
3419  010644  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3420  010646     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3421  010647     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3422  010650  000403                          BR      5$
3423
3424  010652  004737  044704          3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
3425  010656  104001                          ERROR   1               ;SELECTED DR # DOES NOT MATCH RKMR2 DR #
3426
3427  010660  005721                  5$:     TST     (R1)+           ;ADVANCE PTR TO NEXT DR. FLAG
3428  010662  005200                          INC     R0              ;INC DR #
3429  010664  022700  000010                  CMP     #8.,R0
3430  010670  001306                          BNE     1$              ;MORE LEFT.
3431  010672  005737  003454                  TST     DRIVS
3432  010676  001054                          BNE     10$
3433  010700  104123                          ERROR   123             ;NO DRIVES FOUND
3434  010702  000000                          HALT                    ;SETUP CORRECTLY
3435  010704  000137  007740                  JMP     ST5             ;& PRESS 'CONT'
```

```
3436
3437  010710  032737  001000  003336  2$:     BIT     #MDS,HCS2
3438  010716  001015                          BNE     6$
3439  010720  032737  000400  003336          BIT     #UFE,HCS2
3440  010726  001015                          BNE     7$
3441  010730  032737  000001  003346          BIT     #DRA,HDS
3442  010736  001015                          BNE     8$
3443  010740  032737  010000  003336          BIT     #NED,HCS2
3444  010746  001424                          BEQ     9$
3445  010750  000743                          BR      5$
3446
3447  010752  004737  044704          6$:     JSR     PC,BYP          ;TYPE BYP DR #
3448  010756  104002                          ERROR   2               ;MDS DETECTED
3449  010760  000737                          BR      5$
3450
3451  010762  004737  044704          7$:     JSR     PC,BYP
3452  010766  104003                          ERROR   3               ;UFE DETECTED
3453  010770  000733                          BR      5$
3454
3455  010772  032737  010000  003336  8$:     BIT     #NED,HCS2
3456  011000  001713                          BEQ     4$
3457  011002  104401  056541                  TYPE    ,MSG15          ;DRV#
3458  011006  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3459                                                                  ;;TYPE DR#
3460  011010  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3461  011012     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3462  011013     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3463  011014  104010                          ERROR   10              ;DRA & NED BOTH SET
3464  011016  000720                          BR      5$
3465
3466  011020  004737  044704          9$:     JSR     PC,BYP
3467  011024  104004                          ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
3468  011026  000714                          BR      5$
3469  011030  000137  011356          10$:    JMP     NUDRV
3470
3471                                  ;;*****************************************************************
3472                                  ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
3473                                  ;*
3474                                  ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
3475                                  ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
3476                                  ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
3477                                  ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
3478                                  ;*      TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
3479                                  ;*      BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
3480                                  ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
3481                                  ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
3482                                  ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
3483                                  ;*      VERIFY IT WAS NOT SPECIFIED.
3484                                  ;*
3485                                  ;;*****************************************************************
3486  011034  000004                  TST3:   SCOPE
3487  011036  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3488  011044  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3489  011050  005000                          CLR     R0              ;DRIVE ADDR
3490  011052  012701  003456                  MOV     #DRIV0,R1       ;DRIVE FLAG
3491  011056                          1$:
```

```
                        SCOP1
011056  104415          
011060  012706  001100        MOV     #STACK,SP       ;RESTORE STK PTR
                        
011064  012765  000040  000010  MOV     #SCLR,RKCS2(R5)
011072  013737  001414  003372  MOV     T10,TEMP1
011100  004737  044176          JSR     PC,FRDY         ;FIND RDY
011104  104120                  ERROR   120             ;NO RDY AFTER SCLR
011106  010065  000010          MOV     R0,RKCS2(R5)    ;DRV ADDR
011112  012765  000001  000000  MOV     #SELDRV,RKCS1(R5)
011120  013737  001414  003372  MOV     T10,TEMP1
011126  004737  044176          JSR     PC,FRDY         ;FIND RDY
011132  104117                  ERROR   117             ;NO RDY AFTER SELDRV CMD
011134  032737  100000  003334  BIT     #CERR,HCS1
011142  001036                  BNE     2$
011144  013737  003362  003372  MOV     HMR2,TEMP1
011152  042737  177770  003372  BIC     #^C<DRVMSK>,TEMP1
011160  020037  003372          CMP     R0,TEMP1        ;S/B SAME
011164  001010                  BNE     3$
011166  005711          11$:    TST     (R1)
011170  001417                  BEQ     5$
011172  005721          4$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR FLAG
011174  005200                  INC     R0              ;INC DR#
011176  022700  000010          CMP     #8.,R0
011202  001325                  BNE     1$              ;MORE LEFT
011204  000466                  BR      TST4            ;;GO TO NEXT TEST
                        
011206  004737  044704  3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
011212  104001                  ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
011214  005711                  TST     (R1)
011216  001765                  BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
011220  005337  003454  12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
011224  005011                  CLR     (R1)            ;CLEAR DRIVE FLAG
011226  000761                  BR      4$
                        
011230  004737  044704  5$:     JSR     PC,BYP
011234  104005                  ERROR   5               ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
011236  000755                  BR      4$
                        
011240  032737  001000  003336  2$:     BIT     #MDS,HCS2
011246  001027                  BNE     6$
011250  032737  000400  003336          BIT     #UFE,HCS2
011256  001027                  BNE     7$
011260  032737  000001  003346          BIT     #DRA,HDS
011266  001005                  BNE     8$
011270  032737  010000  003336          BIT     #NED,HCS2
011276  001423                  BEQ     9$
011300  000404                  BR      10$
011302  032737  010000  003336  8$:     BIT     #NED,HCS2
011310  001726                  BEQ     11$
011312  005711          10$:    TST     (R1)
011314  001726                  BEQ     4$
                        
011316  004737  044704          JSR     PC,BYP          ;TYPE BYPASS DRIVE#
011322  104006                  ERROR   6
011324  000735                  BR      12$
```

```
3548  011326  004737  044704          6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
3549  011332  104002                          ERROR   2               ;MDS DETECTED
3550  011334  000762                          BR      8$
3551
3552  011336  004737  044704          7$:     JSR     PC,BYP
3553  011342  104003                          ERROR   3               ;UFE DETECTED
3554  011344  000756                          BR      8$
3555
3556  011346  004737  044704          9$:     JSR     PC,BYP
3557  011352  104004                          ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
3558  011354  000752                          BR      8$
3559
3560
3561                                          ;
3562                                          ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
3563                                          ;DRIVE PRESENT
3564                                          ;
3565                                          ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
3566                                          ;UNDER TEST
```

```
                                NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
                                                                ;& TEST CERR IN 'FRDY'

                                ;;*****************************************************************
                                ;*TEST 4         FIND NEXT DRIVE TO BE TESTED
                                ;*
                                ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                ;*      ADDRESS IN '$UNIT'.
                                ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                                ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
                                ;*
                                ;;*****************************************************************
011362  000004                TST4:   SCOPE
011364  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
011372  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
011376  012737  000004  001214          MOV     #$TN-1,$TESTN
011404  012737  000004  001102          MOV     #$TN-1,$TSTNM

011412  005737  003454                  TST     DRIVS           ;ANY DRIVES PRESENT?
011416  001004                          BNE     4$              ;YES BRANCH
011420  104401  056642                  TYPE    ,MSG19          ;ALL DRIVES TESTED
011424  000137  043516                  JMP     $EOP1           ;NO, GO TO END

011430  013701  001346        4$:       MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
011434  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
011440  001402                          BEQ     2$              ;YES, BRANCH
011442  005237  001222        1$:       INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
011446  005721                2$:       TST     (R1)+           ;IS DRIVE PRESENT?
011450  001774                          BEQ     1$              ;NO,FIND NEXT DRIVE PRESENT
011452  005737  003446                  TST     DDPCH           ;DDP CHAIN MODE?
011456  001403                          BEQ     3$              ;NO, BRANCH.
011460  005737  001222                  TST     $UNIT           ;YES, IS IT DRIVE 0?
011464  001766                          BEQ     1$              ;IF YES, DON'T TEST DR 0
011466  010137  001346        3$:       MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT DR. FLAG
011472  104401  056541                  TYPE    ,MSG15          ;"DRIVE"
011476  013700  001222                  MOV     $UNIT,R0
011502  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;DRIVE #
011504  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
011506     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
011507     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

011510  104401  001205                  TYPE    ,$CRLF

                                ;;*****************************************************************
                                ;*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
                                ;*
                                ;;*****************************************************************
011514  000004                TST5:   SCOPE
011516  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
3623  011524  012706  001100              MOV     #STACK,SP
3624
3625  011530  005237  003316              INC     UNLD            ;USED TO CHECK VALID HALT
3626
3627  011534  004737  046106              JSR     PC,SUBCLR
3628  011540  104024                      ERROR   24              ;CERR AFTER SCLR
3629
3630  011542  012765  000007  000000      MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
3631  011550  013737  001414  003372      MOV     T10,TEMP1
3632  011556  004737  044176              JSR     PC,FRDY         ;FIND RDY
3633  011562  104011                      ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
3634  011564  004737  044460              JSR     PC,TSTATN
3635  011570  104012                      ERROR   12              ;NO ATTN AFTER UNLOAD CMD
3636
3637  011572  004737  046106              JSR     PC,SUBCLR
3638  011576  104024                      ERROR   24              ;CERR AFTER SCLR
3639
3640  011600  012737  177777  001160      MOV     #-1,$TMPO       ;SETUP TIMEOUT, ALL 1'S
3641  011606  012765  100000  000000  64$: MOV     #CCLR,RKCS1(R5)
3642  011614  012765  000001  000026      MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3643  011622  004737  045534              JSR     PC,GSTAT
3644  011626  032737  001000  003362      BIT     #D.SPOK,HMR2    ;SEE IF SPEED OK GONE
3645  011634  001404                      BEQ     65$             ;BR IF YES
3646  011636  005337  001160              DEC     $TMPO           ;ELSE TRY AGAIN IF TIME NOT UP
3647  011642  001361                      BNE     64$
3648  011644  104315                      ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
3649  011646                          65$:
3650
3651
3652
3653
3654
3655  011646                          PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
3656                                  .SBTTL STATIC & CYCLE UP TESTS
3657
3658                                  ;;*****************************************************************
3659                                  ;*TEST 6        REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3
3660                                  ;*
3661                                  ;*       CHECKS THE ABILITY TO REFERENCE ALL
3662                                  ;*      DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.
3663                                  ;*
3664                                  ;;*****************************************************************
3665  011646  000004                  TST6:   SCOPE
3666  011650  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3667  011656  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3668
3669  011662  004737  046106                  JSR     PC,SUBCLR
3670  011666  104024                          ERROR   24              ;CERR AFTER SCLR
3671
3672  011670  004737  045534                  JSR     PC,GSTAT
3673  011674  032737  000100  003362          BIT     #D.VV,HMR2
3674  011702  001004                          BNE     4$              ;BR IF VV SET
3675  011704  012737  000040  003424          MOV     #D.DRA,E.AO     ;LOAD EXPECTED VALUE FOR AO
3676  011712  000403                          BR      5$
3677
3678  011714  012737  000140  003424  4$:     MOV     #<D.DRA!D.VV>,E.AO
```

```
                                5$:     CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.HDHM!D.BRHM!D.DOOR!D.CART>,E.A1      ;EXPECTED MSG A1
                                        MOV     #1,E.B1         ;EXPECTED MSG B1
                                        CLR     E.A2            ;EXPECTED MSG A2
                                        MOV     #2,E.B2         ;EXPECTED MSG B2
                                        MOV     #3,E.B3         ;EXPECTED MSG B3

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                                        .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
                                        ERROR   16              ;MSG A0 ERROR FOR DRIVE UNLOADED
                                        ERROR   17              ;MSG B0 ERROR
                                        ERROR   20              ;MSG A1 ERROR
                                        ERROR   21              ;MSG B1 ERROR

                                        TST     CYLDIF          ;SEE IF MSG A2=0
                                        BEQ     64$             ;BR IF YES
                                        ERROR   22              ;MSG A2 NOT CLEARED FOR DRIVE UNLOADED
                                64$:    TST     CYLADD          ;SEE IF MSG B2=0
                                        BEQ     65$             ;BR IF YES
                                        ERROR   23              ;MSG B2 NOT CLEARED FOR DRIVE UNLOADED
                                65$:
                                        CMP     HEADA,#1        ;FOR HEAD 0, B3=1
                                        BEQ     TST7            ;;GO TO NXT TST IF YES
                                        ERROR   56              ;HEAD REG IN B3 NOT 0 IN UNLOAD


                                ;;*****************************************************************
                                ;*TEST 7        PRINT DRIVE SERIAL NUMBER
                                ;*
                                ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
                                ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
                                ;*
                                ;*****************************************************************
                                TST7:   SCOPE
                                        MOV     #1,$TIMES       ;;DO 1 ITERATION
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                        TST     $PASS
                                        BNE     TST10           ;;GO TO NEXT IF NOT FIRST PASS
                                        JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
                                        ERROR   24              ;CERR AFTER SCLR

                                        TYPE    ,MSG16          ;DRIVE SERIAL NO.
                                        MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
                                        JSR     PC,GSTAT        ;GET STATUS
                                        MOV     HMR2,R1         ;GET SERIAL #
                                        MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
                                        MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
                                        MOV     #3,R3           ;SETUP CHAR COOUNT
                                        ROL     R1              ;INITIALIZE BIT POSITIONS
                                        ROL     R1
                                1$:     ROL     R1              ;GET NEXT 4 BITS
                                        ROL     R1
                                        ROL     R1
                                        ROL     R1
```

```
3735  012132  010100                          MOV     R1,R0           ;GET WORKING COPY
3736  012134  042700  177760                  BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
3737  012140  052700  000060                  BIS     #60,R0          ;CONVERT TO ASCII DIGIT
3738  012144  110024                          MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
3739  012146  005303                          DEC     R3
3740  012150  001364                          BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
3741  012152  105014                          CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR
3742  012154  004737  054526                  JSR     PC,$SUPRS       ;TYPE
3743  012160  104401  001205                  TYPE    ,$CRLF
3744  012164  104401  001205                  TYPE    ,$CRLF
3745
3746                                  ;;*****************************************************************
3747                                  ;*TEST 10       SET VV WITH PACK CMD
3748                                  ;*
3749                                  ;*      IF VV IS RESET, THE PACK CMD IS USED TO SET IT.
3750                                  ;*
3751                                  ;;*****************************************************************
3752  012170  000004                  TST10:  SCOPE
3753  012172  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3754  012200  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3755
3756  012204  005065  000026                  CLR     RKMR1(R5)       ;SELECT BYTE 0
3757  012210  004737  045534                  JSR     PC,GSTAT        ;GET STATUS
3758  012214  032737  000100  003362          BIT     #D.VV,HMR2
3759  012222  001024                          BNE     TST11           ;;GO TO NEXT TEST IF VV SET
3760
3761  012224  104415                          SCOP1
3762  012226  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3763
3764  012232  004737  046106                  JSR     PC,SUBCLR
3765  012236  104024                          ERROR   24              ;CERR AFTER SCLR
3766
3767  012240  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;CMD TO SET VV
3768  012246  013737  001414  003372          MOV     T10,TEMP1
3769  012254  004737  044176                  JSR     PC,FRDY         ;FIND RDY
3770  012260  104116                          ERROR   116             ;RDY NOT SET AFTER PACK CMD
3771
3772  012262  032737  000100  003362          BIT     #D.VV,HMR2
3773  012270  001001                          BNE     TST11           ;;GO TO NEXT TEST IF VV NOW SET
3774  012272  104027                          ERROR   27              ;PACK DID NOT SET V.V.
3775
3776                                  ;;*****************************************************************
3777                                  ;*TEST 11       RELEASE DRIVE
3778                                  ;*
3779                                  ;*      TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK
3780                                  ;*
3781                                  ;;*****************************************************************
3782  012274  000004                  TST11:  SCOPE
3783  012276  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3784  012304  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3785
3786  012310  004737  046106                  JSR     PC,SUBCLR       ;DO SUBSYS CLEAR & GET STATUS
3787  012314  104024                          ERROR   24              ;CONTR ERROR SET AFTER SCLR
3788  012316  032737  000400  003336          BIT     #UFE,HCS2
3789  012324  001401                          BEQ     1$
3790  012326  104003                          ERROR   3               ;UFE SET AFTER SCLR
```

```
3791
3792  012330                                  1$:
3793  012330  104415                                  SCOP1
3794  012332  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR
3795
3796  012336  004737  046106                          JSR     PC,SUBCLR
3797  012342  104024                                  ERROR   24              ;CERR AFTER SCLR
3798
3799  012344  062765  000010  000010                  ADD     #RLS,RKCS2(R5)  ;ADD RELEASE BIT TO $UNIT
3800  012352  004737  045534                          JSR     PC,GSTAT        ;GET STATUS
3801
3802  012356  032737  100000  003334                  BIT     #CERR,HCS1      ;CHECK FOR CONTR ERROR
3803  012364  001401                                  BEQ     2$
3804  012366  104025                                  ERROR   25              ;RLS SET CERR
3805  012370  032737  000400  003336  2$:             BIT     #UFE,HCS2
3806  012376  001401                                  BEQ     TST12           ;;GO TO NEXT TEST IF SET
3807  012400  104026                                  ERROR   26              ;SACK SET AFTER RLS SENT
3808
3809                                          ;;**************************************************************
3810                                          ;*TEST 12        DRIVE TYPE TEST
3811                                          ;*
3812                                          ;*      THIS TEST COMPARES DRIVE TYPE IN MSG A AGAINST 'DDT' IN RKDS.
3813                                          ;*      WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.
3814                                          ;*
3815                                          ;;**************************************************************
3816  012402  000004                          TST12:  SCOPE
3817  012404  012737  000001  001174                  MOV     #1,$TIMES       ;;DO 1 ITERATION
3818  012412  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR
3819
3820  012416  004737  046106                          JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
3821  012422  104024                                  ERROR   24              ;CONT ERROR SET AFT SUBSYS CLEAR
3822  012424  032737  000400  003362                  BIT     #D.DDT,HMR2
3823  012432  001401                                  BEQ     2$
3824  012434  104030                                  ERROR   30              ;DR TYPE SET IN MR2
3825  012436  032737  000400  003346  2$:             BIT     #DDT,HDS
3826  012444  001401                                  BEQ     3$
3827  012446  104031                                  ERROR   31              ;DDT SET IN RKDS
3828  012450  032737  000040  003350  3$:             BIT     #DTYE,HER
3829  012456  001401                                  BEQ     4$
3830  012460  104032                                  ERROR   32              ;DTYE SET IN RKER
3831
3832  012462                                  4$:
3833  012462  104415                                  SCOP1
3834  012464  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR
3835
3836  012470  004737  046106                          JSR     PC,SUBCLR
3837  012474  104024                                  ERROR   24              ;CERR AFTER SCLR
3838
3839  012476  012765  002001  000000                  MOV     #<CDT!SELDRV>,RKCS1(R5)  ;GET STATUS WITH CDT SET
3840  012504  013737  001414  003372                  MOV     T10,TEMP1
3841  012512  004737  044176                          JSR     PC,FRDY         ;FIND RDY
3842  012516  104117                                  ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
3843  012520  032737  000400  003362                  BIT     #D.DDT,HMR2
3844  012526  001401                                  BEQ     5$
3845  012530  104030                                  ERROR   30              ;DR TYPE SET IN MR2
3846  012532  032737  000400  003346  5$:             BIT     #DDT,HDS
```

```
                                        BEQ     6$
                                        ERROR   31              ;DDT SET IN RKDS
012544  032737  000040  003350  6$:   BIT     #DTYE,HER       ;DTYE=DDT(NOT)*CDT
012552  001001                        BNE     7$
012554  104033                        ERROR   33              ;DTYE NOT SET AFT WRITING CDT=1
012556  032737  100000  003334  7$:   BIT     #CERR,HCS1
012564  001001                        BNE     TST13           ;;GO TO NEXT TEST
012566  104034                        ERROR   34              ;CERR NOT SET AFT WRITING CDT=1

                                ;;*****************************************************************
                                ;*TEST 13       C-D PARITY ERROR DETECTION
                                ;*
                                ;*      TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
                                ;*      THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
                                ;*      THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
                                ;*      THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
                                ;*      CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
                                ;*      AN ERROR AND SET DCPAR.
                                ;*      THE ERROR CONDITION IS RESET WITH THE CLEAR CMD
                                ;*
                                ;;*****************************************************************
012570  000004                TST13:  SCOPE
012572  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
012600  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

012604  004737  046106                JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
012610  104024                        ERROR   24              ;CONT ERROR AFTER SUBSYS CLR
012612  032737  001000  003364        BIT     #D.PAR,HMR3
012620  001401                        BEQ     2$
012622  104035                        ERROR   35              ;C-D PARITY ERROR SET IN MR3
012624  032737  020000  003334  2$:   BIT     #DCPAR,HCS1
012632  001401                        BEQ     3$
012634  104036                        ERROR   36              ;DCPAR SET IN CS1

012636                          3$:
012636  104415                SCOP1
012640  012706  001100        MOV     #STACK,SP       ;RESTORE STK PTR

012644  004737  046106                JSR     PC,SUBCLR
012650  104024                        ERROR   24              ;CERR AFTER SCLR

012652  012765  000020  000026        MOV     #PAT,RKMR1(R5)  ;SELECT BYTE 0 & EVEN PARITY
012660  004737  045534                JSR     PC,GSTAT        ;GET STATUS
012664  032737  000200  003364        BIT     #D.FLT,HMR3
012672  001001                        BNE     4$
012674  104037                        ERROR   37              ;FAULT NOT SET IN MR3
012676  032737  001000  003364  4$:   BIT     #D.PAR,HMR3
012704  001001                        BNE     5$
012706  104040                        ERROR   40              ;C-D PARITY ERROR NOT SET IN MR3
012710  032737  020000  003334  5$:   BIT     #DCPAR,HCS1
012716  001001                        BNE     6$
012720  104041                        ERROR   41              ;DCPAR NOT SET AFT WRITING PAT IN MR1
012722  032737  100000  003334  6$:   BIT     #CERR,HCS1
012730  001001                        BNE     TST14           ;;GO TO NEXT TEST
012732  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.
                                ;*
                                ;;*****************************************************************
012734  000004                  TST14:  SCOPE
012736  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
012744  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

012750  004737  046106                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
012754  104024                          ERROR   24              ;CERR AFTER SCLR

012756  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5) ;START SPINDLE CMD
012764  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
012772  004737  044176                  JSR     PC,FRDY
012776  104121                          ERROR   121             ;RDY NOT SET AFTER START SPIN CMD

013000  004737  045534                  JSR     PC,GSTAT        ;WORD 0
013004  032737  010000  003362          BIT     #D.SPIN,HMR2
013012  001001                          BNE     13$
013014  104306                          ERROR   306             ;SPIN NOT SET AFTER START SPIN CMD

013016  012737  014360  001176  13$:    MOV     #25$,$ESCAPE
013024  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
013030  000401                          BR      15$
013032  104316                          ERROR   316
013034  012737  010140  003424  15$:    MOV     #<D.SPIN!D.VV!D.DRA>,E.A0       ;LOAD IN EXPECTED VALUES
013042  005037  003426                  CLR     E.B0
013046  012737  000740  003430          MOV     #<D.CART!D.DOOR!D.HDHM!D.BRHM>,E.A1
013054  012737  000001  003432          MOV     #1,E.B1

013062  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
013066  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
013070  104057                          ERROR   57              ;MSG A0 ERROR AFTER START SPIN CMD REC'D BY DRIVE
013072  104060                          ERROR   60              ;MSG B0 ERROR
013074  104061                          ERROR   61              ;MSG A1 ERROR
013076  104062                          ERROR   62              ;MSG B1 ERROR
013100  005737  003502                  TST     DOTIM
013104  001135                          BNE     3$              ;BRANCH IF P OR L CLOCK PRESENT
013106  012737  014404  001176  1$:     MOV     #30$,$ESCAPE
013114  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
013122  013737  001416  003374          MOV     T100,TEMP2      ;SETUP TIMEOUT
013130  004737  044512                  JSR     PC,FATT1        ;FIND ATTN
013134  104067                          ERROR   67              ;NO ATTN AFTER HEAD LOADING
013136                          2$:

013136  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
013144  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
013150  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
013156  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
013164  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
```

```
3959  013170  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
3960  013176  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
3961
3962  013204  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
3963  013210  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
3964  013212  104063                          ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOAD
3965  013214  104064                          ERROR   64              ;MSG B0 ERROR
3966  013216  104065                          ERROR   65              ;MSG A1 ERROR
3967  013220  104066                          ERROR   66              ;MSG B1 ERROR
3968
3969  013222  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
3970  013226  001401                          BEQ     64$             ;BR IF YES
3971  013230  104175                          ERROR   175             ;MSG A2 NOT CLEARED AT END OF HEAD LOAD
3972  013232  005737  001364          64$:    TST     CYLADD          ;SEE IF MSG B2=0
3973  013236  001401                          BEQ     65$             ;BR IF YES
3974  013240  104176                          ERROR   176             ;MSG B2 NOT CLEARED AT END OF HEAD LOAD
3975  013242                          65$:
3976
3977  013242  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3978  013250  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
3979  013256  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3980  013264  013737  001414  003372          MOV     T10,TEMP1
3981  013272  004737  044176                  JSR     PC,FRDY         ;FIND RDY
3982  013276  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3983  013300  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
3984  013304  000401                          BR      66$
3985  013306  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3986  013310                          66$:
3987
3988  013310  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
3989  013316  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
3990  013322  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
3991  013330  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
3992  013336  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
3993  013342  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
3994  013350  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
3995
3996  013356  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
3997  013362  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
3998  013364  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
3999  013366  104265                          ERROR   265             ;MSG B0 ERROR
4000  013370  104274                          ERROR   274             ;MSG A1 ERROR
4001  013372  104266                          ERROR   266             ;MSG B1 ERROR
4002
4003  013374  000137  014140                  JMP     12$
4004
4005  013400  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
4006  013406  013737  001370  001372          MOV     HZ,COUNT
4007  013414  012737  000074  001374          MOV     #60.,SEC
4008  013422  004737  047614                  JSR     PC,CLKON        ;TURN CLK INTR ON FOR 60 SEC MAX
4009  013426  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4010  013434  004737  045534          4$:     JSR     PC,GSTAT
4011  013440  032737  002000  003362          BIT     #D.FWD,HMR2
4012  013446  001004                          BNE     5$
4013  013450  005737  001376                  TST     TIMUP           ;IS 60 SEC DELAY UP?
4014  013454  001767                          BEQ     4$              ;BRANCH IF NO & REPEAT
```

```
4015  013456  104070                          ERROR   70              ;FWD NOT SET WITHIN 60 SEC FROM
4016                                                                  ;START SPINDLE CMD.
4017  013460  004737  047710          5$:     JSR     PC,CLKOF        ;TURN OFF CLOCK INTERRORUPT
4018  013464  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4019  013472  012737  011610  003372          MOV     #5000.,TEMP1
4020  013500  004737  044666                  JSR     PC,DLY          ;WAIT APPROX 100MS FOR PIP
4021  013504  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
4022  013510  000401                          BR      16$
4023  013512  104317                          ERROR   317             ;UNEXP ATTN AFTER ST SPIN & FWD DETECTED
4024  013514  012737  030140  003424  16$:    MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
4025  013522  005037  003426                  CLR     E.B0
4026  013526  012737  013700  003430          MOV     #<D.LOAD!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM>,E.A1
4027  013534  012737  000001  003432          MOV     #1,E.B1
4028
4029  013542  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4030  013546  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4031  013550  104071                          ERROR   71              ;MSG A0 ERROR AFTER START SPIN & FWD DETECTED
4032  013552  104072                          ERROR   72              ;MSG B0 ERROR
4033  013554  104073                          ERROR   73              ;MSG A1 ERROR
4034  013556  104074                          ERROR   74              ;MSG B1 ERROR
4035
4036  013560  013737  001370  001372          MOV     HZ,COUNT
4037  013566  012737  000005  001374          MOV     #5,SEC
4038  013574  004737  047614                  JSR     PC,CLKON        ;TURN CLK INTR ON FOR 5 SEC MAX
4039  013600  012765  000001  000026  6$:     MOV     #1,RKMR1(R5)    ;WORD 1
4040  013606  004737  045534                  JSR     PC,GSTAT
4041  013612  032737  002000  003362          BIT     #D.FWD,HMR2
4042  013620  001404                          BEQ     7$
4043  013622  005737  001376                  TST     TIMUP
4044  013626  001764                          BEQ     6$
4045  013630  104075                          ERROR   75              ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
4046                                                                  ;FROM START SPINDLE CMD.
4047  013632  004737  047710          7$:     JSR     PC,CLKOF        ;TURN OFF CLK INTERRORUPT
4048  013636  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
4049  013642  000401                          BR      17$
4050  013644  104320                          ERROR   320             ;UNEXP ATTN AFTER INNER LIM DETECT
4051  013646  012737  030140  003424  17$:    MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
4052  013654  005037  003426                  CLR     E.B0
4053  013660  012737  025720  003430          MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4054  013666  012737  000001  003432          MOV     #1,E.B1
4055
4056  013674  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4057  013700  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4058  013702  104076                          ERROR   76              ;MSG A0 ERROR AT INNER LIMIT DETECT
4059  013704  104077                          ERROR   77              ;MSG B0 ERROR
4060  013706  104100                          ERROR   100             ;MSG A1 ERROR
4061  013710  104101                          ERROR   101             ;MSG B1 ERROR
4062
4063  013712  013737  001370  001372          MOV     HZ,COUNT
4064  013720  012737  000004  001374          MOV     #4,SEC
4065  013726  004737  047614                  JSR     PC,CLKON        ;TURN CLK INTR ON FOR 4 SEC MAX
4066  013732  012765  000001  000026  8$:     MOV     #1,RKMR1(R5)    ;WORD 1
4067  013740  004737  045534                  JSR     PC,GSTAT
4068  013744  032737  002000  003362          BIT     #D.FWD,HMR2
4069  013752  001004                          BNE     9$
4070  013754  005737  001376                  TST     TIMUP
```

```
4071  013760  001764                        BEQ     8$
4072  013762  104102                        ERROR   102             ;FWD NOT DETECTED WITHIN 4 SEC IN RTZ PORTION OF
4073                                                                  ;START SPINDLE CMD.
4074  013764  004737  047710        9$:     JSR     PC,CLKOF        ;TURN CLOCK INTR OFF.
4075  013770  004737  044460                JSR     PC,TSTATN       ;TEST FOR ATTN
4076  013774  000401                        BR      18$
4077  013776  104321                        ERROR   321             ;UNEXP ATTN AFTER OUTER LIM TO CYL 0
4078  014000  012737  030140  003424  18$:  MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
4079  014006  005037  003426                CLR     E.B0
4080  014012  012737  023720  003430        MOV     #<D.RTZ!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4081  014020  012737  000001  003432        MOV     #1,E.B1
4082
4083  014026  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4084  014032  000000                        .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4085  014034  104103                        ERROR   103             ;MSG A0 ERROR FROM OUT LIM TO CYL 0 DURING LOAD
4086  014036  104104                        ERROR   104             ;MSG B0 ERROR
4087  014040  104105                        ERROR   105             ;MSG A1 ERROR
4088  014042  104106                        ERROR   106             ;MSG B1 ERROR
4089  014044  013737  001370  001372        MOV     HZ,COUNT
4090  014052  012737  000001  001374        MOV     #1,SEC
4091  014060  004737  047614                JSR     PC,CLKON        ;TURN CLK INTR ON FOR 1 SEC MAX
4092  014064  005065  000026        10$:    CLR     RKMR1(R5)       ;WORD 0
4093  014070  004737  045534                JSR     PC,GSTAT
4094  014074  032737  000200  003362        BIT     #D.DRDY,HMR2    ;SEE IF DRIVE READY
4095  014102  001004                        BNE     11$
4096  014104  005737  001376                TST     TIMUP
4097  014110  001765                        BEQ     10$
4098  014112  104107                        ERROR   107             ;DRIVE READY NOT SET WITHIN 1 SEC FROM
4099                                                                  ;FWD IN RTZ PORTION OF START SPIN CMD
4100  014114  004737  047710        11$:    JSR     PC,CLKOF        ;TURN CLOCKS OFF
4101  014120  004737  044460                JSR     PC,TSTATN       ;TEST ATTN
4102  014124  104067                        ERROR   67              ;NO ATTN AFTER START SPIN CMD
4103  014126  012737  014404  001176        MOV     #30$,$ESCAPE
4104  014134  000137  013136                JMP     2$              ;CHECK RKMR 2 & 3 WORDS 0 & 1
4105
4106  014140  005037  001176        12$:    CLR     $ESCAPE
4107  014144  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
4108  014152  005065  000026                CLR     RKMR1(R5)
4109  014156  004737  045534                JSR     PC,GSTAT
4110  014162  032737  010000  003362        BIT     #D.SPIN,HMR2
4111  014170  001003                        BNE     14$
4112  014172  104306                        ERROR   306             ;SPIN NOT SET AFTER ST. SPIN CMD
4113  014174  000137  043470                JMP     $EOP            ;ABORT DRIVE
4114  014200  005037  003316        14$:    CLR     UNLD            ;USED FOR VALID HALT
4115  014204  004737  050036                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
4116  014210  000507                        BR      TST15           ;;GO TO NEXT TEST
4117                                                                  ;RETURN HERE IF SW 14 IS SET OR
4118                                                                  ;SW 8 WITH SWR <7:0> APPLY
4119  014212  005237  003316        20$:    INC     UNLD
4120
4121  014216  004737  046106                JSR     PC,SUBCLR
4122  014222  104024                        ERROR   24              ;CERR AFTER SCLR
4123
4124  014224  012765  000007  000000        MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
4125  014232  013737  001414  003372        MOV     T10,TEMP1
4126  014240  004737  044176                JSR     PC,FRDY         ;FIND RDY
```

```
4127  014244  104011                        ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
4128  014246  004737  044460                JSR     PC,TSTATN
4129  014252  104012                        ERROR   12              ;NO ATTN AFTER UNLOAD CMD
4130
4131  014254  004737  046106                JSR     PC,SUBCLR
4132  014260  104024                        ERROR   24              ;CERR AFTER SCLR
4133
4134  014262  012737  177777  001160        MOV     #-1,$TMPO       ;SETUP TIMEOUT, ALL 1'S
4135  014270  012765  100000  000000  67$:   MOV     #CCLR,RKCS1(R5)
4136  014276  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4137  014304  004737  045534                JSR     PC,GSTAT
4138  014310  032737  001000  003362        BIT     #D.SPOK,HMR2    ;SEE IF SPEED OK GONE
4139  014316  001404                        BEQ     68$             ;BR IF YES
4140  014320  005337  001160                DEC     $TMPO           ;ELSE TRY AGAIN IF TIME NOT UP
4141  014324  001361                        BNE     67$
4142  014326  104315                        ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
4143  014330                        68$:
4144
4145  014330  004737  046106                JSR     PC,SUBCLR
4146  014334  104024                        ERROR   24              ;CERR AFTER SCLR
4147
4148  014336  005037  001176                CLR     $ESCAPE
4149  014342  005737  001410                TST     LPFLG
4150  014346  001402                        BEQ     69$
4151  014350  000177  164534                JMP     @$LPERR         ;SW 9 WAS SET.
4152  014354  000177  164526        69$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
4153
4154
4155
4156  014360  004737  047710        25$:    JSR     PC,CLKOF
4157  014364  005237  001410                INC     LPFLG
4158  014370  032777  001000  164542        BIT     #SW9,@SWR       ;LOOP ON ERROR?
4159  014376  001305                        BNE     20$             ;YES, RECONDITION DRIVE
4160  014400  000137  013106                JMP     1$              ;RETURN TO MAINLINE
4161  014404  004737  047710        30$:    JSR     PC,CLKOF
4162  014410  005237  001410                INC     LPFLG
4163  014414  032777  001000  164516        BIT     #SW9,@SWR       ;LOOP ON ERROR?
4164  014422  001273                        BNE     20$             ;YES, RECONDITION DRIVE
4165  014424  000137  014140                JMP     12$             ;RETURN TO MAINLINE
4166                                .SBTTL SEEK/READ HEADER/WRITE HEADER TESTS
4167
4168                                ;;*****************************************************************
4169                                ;*TEST 15       STATIC CYL DIFF AND CYL ADDR REG TEST; PART 1
4170                                ;*
4171                                ;*      THIS TEST CHECKS EACH BIT OF THE CYL DIFFERENCE
4172                                ;*      AND CYL ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
4173                                ;*      MAJOR CYLS (0,1,2,4,8,16,32,64,128,256) WITH EVEN PARITY SET.
4174                                ;*      THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKING.
4175                                ;*      THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
4176                                ;*      NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.
4177                                ;*
4178                                ;;*****************************************************************
4179  014430  000004                TST15:  SCOPE
4180  014432  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
4181  014440  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
4182
```

```
 4183  014444  005000                          CLR     R0                      ;CYL # REGISTER
 4184  014446  012737  100000  003402          MOV     #BIT15,TEMP5
 4185
 4186  014454                          1$:
 4187  014454  104415                          SCOP1
 4188  014456  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR
 4189
 4190  014462  004737  046106                  JSR     PC,SUBCLR
 4191  014466  104024                          ERROR   24                      ;CERR AFTER SCLR
 4192
 4193
 4194  014470  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4195  014476  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 4196  014504  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
 4197                                                                          ;RESET CYL DIFF/OFFSET & CYL ADDR REG
 4198                                                                          ;IN RKMR2 & RKMR3 RESP.
 4199  014512  013737  001414  003372          MOV     T10,TEMP1
 4200  014520  004737  044176                  JSR     PC,FRDY                 ;FIND RDY
 4201  014524  104124                          ERROR   124                     ;RDY NOT SET AFTER RECAL CMD
 4202
 4203  014526  012765  000001  000026          MOV     #1,RKMR1(R5)            ;SELECT WORD 1
 4204  014534  004737  045534                  JSR     PC,GSTAT
 4205  014540  032737  020000  003362          BIT     #D.RTZ,HMR2
 4206  014546  001001                          BNE     64$
 4207  014550  104244                          ERROR   244                     ;RTZ NOT SET DURING RECAL CMD
 4208  014552  013737  001412  003374  64$:    MOV     T1,TEMP2                ;SETUP TIMEOUT
 4209  014560  004737  044512                  JSR     PC,FATT1                ;FIND ATTN
 4210  014564  104055                          ERROR   55                      ;NO ATTN AFTER RECAL CMD
 4211
 4212  014566  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4213  014574  005037  003426                  CLR     E.B0                    ;EXPECTED MSG B0
 4214  014600  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4215  014606  012737  000001  003432          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
 4216  014614  005037  003434                  CLR     E.A2                    ;EXPECTED MSG A2
 4217  014620  012737  000002  003436          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
 4218  014626  012737  000003  003442          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
 4219
 4220  014634  004737  044720                  JSR     PC,CHKMSG               ;CHECK MSGS A0,B0,A1,B1
 4221  014640  000007                          .WORD   T.A2!T.B2!T.B3                  ;& MSGS SPECIFIED HERE
 4222  014642  104221                          ERROR   221                     ;MSG A0 ERROR AFTER RECAL CMD
 4223  014644  104275                          ERROR   275                     ;MSG B0 ERROR
 4224  014646  104222                          ERROR   222                     ;MSG A1 ERROR
 4225  014650  104276                          ERROR   276                     ;MSG B1 ERROR
 4226
 4227  014652  005737  001362                  TST     CYLDIF                  ;SEE IF MSG A2=0
 4228  014656  001401                          BEQ     65$                     ;BR IF YES
 4229  014660  104047                          ERROR   47                      ;MSG A2 NOT CLEARED AFTER RECAL CMD
 4230  014662  005737  001364          65$:    TST     CYLADD                  ;SEE IF MSG B2=0
 4231  014666  001401                          BEQ     66$                     ;BR IF YES
 4232  014670  104050                          ERROR   50                      ;MSG B2 NOT CLEARED AFTER RECAL CMD
 4233  014672                          66$:
 4234
 4235  014672  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4236  014700  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4237  014706  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 4238  014714  013737  001414  003372          MOV     T10,TEMP1
```

```
4239  014722  004737  044176          JSR     PC,FRDY         ;FIND RDY
4240  014726  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4241  014730  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
4242  014734  000401                  BR      67$
4243  014736  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4244  014740                  67$:
4245
4246  014740  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
4247  014746  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
4248  014752  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
4249  014760  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4250  014766  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
4251  014772  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4252  015000  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4253
4254  015006  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4255  015012  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4256  015014  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4257  015016  104265                  ERROR   265             ;MSG B0 ERROR
4258  015020  104274                  ERROR   274             ;MSG A1 ERROR
4259  015022  104266                  ERROR   266             ;MSG B1 ERROR
4260
4261
4262  015024  104415                  SCOP1
4263  015026  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
4264
4265  015032  004737  046106          JSR     PC,SUBCLR
4266  015036  104024                  ERROR   24              ;CERR AFTER SCLR
4267
4268  015040  005237  001462          INC     BYPCERR         ;DO NOT TEST CERR IN GSTAT1
4269  015044  012765  000020  000026  MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
4270  015052  010065  000020          MOV     R0,RKDC(R5)     ;CYL ADDR
4271  015056  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD.
4272  015064  013737  001414  003372  MOV     T10,TEMP1
4273  015072  004737  044176          JSR     PC,FRDY         ;FIND RDY
4274  015076  104122                  ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
4275  015100  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
4276  015104  104125                  ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
4277  015106  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
4278  015114  012737  001200  003426  MOV     #<D.FLT!D.PAR>,E.B0
4279  015122  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4280  015130  012737  000001  003432  MOV     #1,E.B1
4281  015136  010037  003434          MOV     R0,E.A2
4282  015142  010037  003436          MOV     R0,E.B2
4283  015146  052737  000002  003436  BIS     #2,E.B2         ;ADD MSG ID
4284
4285  015154  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4286  015160  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4287  015162  104110                  ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
4288  015164  104111                  ERROR   111             ;MSG B0 ERROR
4289  015166  104146                  ERROR   146             ;MSG A1 ERROR
4290  015170  104147                  ERROR   147             ;MSG B1 ERROR
4291
4292  015172  020037  001364          CMP     R0,CYLADD
4293  015176  001401                  BEQ     2$
4294  015200  104043                  ERROR   43              ;CYL ADDR IN B2 NOT=RKDC
```

```
4295
4296  015202  020037  001362          2$:     CMP     R0,CYLDIF
4297  015206  001401                          BEQ     3$
4298  015210  104044                          ERROR   44              ;CYL DIFF IN A2 NOT=RKDC
4299
4300  015212  005037  001462          3$:     CLR     BYPCERR         ;ALLOW CHECKING FOR ANY CERR IN GSTAT1
4301  015216  006137  003402                  ROL     TEMP5           ;SET CARRY ONLY ONCE
4302  015222  006100                          ROL     R0              ;SELECT NEXT MAJOR CYL
4303  015224  020027  001000                  CMP     R0,#1000        ;ALL MAJOR CYL DONE?
4304  015230  001001                          BNE     4$              ;BRANCH IF NO
4305  015232  000402                          BR      TST16           ;;GO TO  NEXT TST
4306  015234  000137  014454          4$:     JMP     1$
4307
4308                                  ;;**************************************************************
4309                                  ;*TEST 16       STATIC CYL DIFF & CYL ADDR REG TEST-PART 2
4310                                  ;*
4311                                  ;*      THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
4312                                  ;*      DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
4313                                  ;*      ALL CYLS FROM EVERY OTHER CYL. (N SQUARE SEEKS).
4314                                  ;*      IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.
4315                                  ;*
4316                                  ;;**************************************************************
4317  015240  000004                  TST16:  SCOPE
4318  015242  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4319  015250  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4320
4321  015254  005737  001340                  TST     BYPT16
4322  015260  001404                          BEQ     13$
4323  015262  104401  056274                  TYPE    ,MSG9           ;BYPASSING TEST 16
4324  015266  000137  016530                  JMP     12$
4325
4326  015272  104401  056236          13$:    TYPE    ,MSG8           ;PLEASE WAIT, LONG TEST
4327
4328  015276  005037  001350                  CLR     FRCYL           ;FROM CYL
4329  015302  005037  001352                  CLR     TOCYL           ;TO CYL
4330  015306  005037  001354                  CLR     CCYL            ;CURRENT CYL
4331  015312  005037  001356                  CLR     PCYL            ;PREV CYL
4332
4333  015316  104415                          SCOP1
4334  015320  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4335
4336  015324  004737  046106                  JSR     PC,SUBCLR
4337  015330  104024                          ERROR   24              ;CERR AFTER SCLR
4338
4339
4340  015332  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4341  015340  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
4342  015346  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
4343                                                                  ;RESET CYL DIFF/OFFSET & CYL ADDR REG
4344                                                                  ;IN RKMR2 & RKMR3 RESP.
4345  015354  013737  001414  003372          MOV     T10,TEMP1
4346  015362  004737  044176                  JSR     PC,FRDY         ;FIND RDY
4347  015366  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
4348
4349  015370  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4350  015376  004737  045534                  JSR     PC,GSTAT
```

```
 4351  015402  032737  020000  003362          BIT     #D.RTZ,HMR2
 4352  015410  001001                          BNE     64$
 4353  015412  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
 4354  015414  013737  001412  003374  64$:    MOV     T1,TEMP2        ;SETUP TIMEOUT
 4355  015422  004737  044512                  JSR     PC,FATT1        ;FIND ATTN
 4356  015426  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
 4357
 4358  015430  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4359  015436  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
 4360  015442  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4361  015450  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4362  015456  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
 4363  015462  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4364  015470  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4365
 4366  015476  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4367  015502  000007                          .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
 4368  015504  104221                          ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
 4369  015506  104275                          ERROR   275             ;MSG B0 ERROR
 4370  015510  104222                          ERROR   222             ;MSG A1 ERROR
 4371  015512  104276                          ERROR   276             ;MSG B1 ERROR
 4372
 4373  015514  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
 4374  015520  001401                          BEQ     65$             ;BR IF YES
 4375  015522  104047                          ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
 4376  015524  005737  001364          65$:    TST     CYLADD          ;SEE IF MSG B2=0
 4377  015530  001401                          BEQ     66$             ;BR IF YES
 4378  015532  104050                          ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
 4379  015534                          66$:
 4380
 4381  015534  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4382  015542  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4383  015550  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 4384  015556  013737  001414  003372          MOV     T10,TEMP1
 4385  015564  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 4386  015570  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4387  015572  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4388  015576  000401                          BR      67$
 4389  015600  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4390  015602                          67$:
 4391
 4392  015602  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
 4393  015610  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
 4394  015614  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4395  015622  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4396  015630  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
 4397  015634  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4398  015642  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4399
 4400  015650  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4401  015654  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4402  015656  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 4403  015660  104265                          ERROR   265             ;MSG B0 ERROR
 4404  015662  104274                          ERROR   274             ;MSG A1 ERROR
 4405  015664  104266                          ERROR   266             ;MSG B1 ERROR
 4406
```

```
4407
4408  015666  104415                          SCOP1
4409  015670  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4410
4411  015674  004737  046106                  JSR     PC,SUBCLR
4412  015700  104024                          ERROR   24              ;CERR AFTER SCLR
4413
4414
4415  015702  012765  000020  000026  1$:     MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
4416  015710  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;SET TO CYL ADDR
4417  015716  013737  001352  001354          MOV     TOCYL,CCYL      ;CURRENT CYL
4418  015724  013737  001354  003376          MOV     CCYL,TEMP3
4419  015732  013737  001356  003400          MOV     PCYL,TEMP4      ;PREV CYL
4420  015740  163737  003376  003400          SUB     TEMP3,TEMP4
4421  015746  100002                          BPL     2$              ;BR IF TEMP4 IS POS
4422  015750  005437  003400                  NEG     TEMP4
4423  015754  013737  003400  001360  2$:     MOV     TEMP4,CALDIF
4424  015762  013737  001354  001356          MOV     CCYL,PCYL
4425  015770  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD.
4426  015776  013737  001414  003372          MOV     T10,TEMP1
4427  016004  004737  044176                  JSR     PC,FRDY         ;FIND RDY
4428  016010  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
4429  016012  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
4430  016016  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
4431  016020  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
4432  016026  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
4433  016034  004737  046476                  JSR     PC,RDCYLA       ;READ CYL ADDR
4434  016040  023737  001352  001364          CMP     TOCYL,CYLADD    ;SEE IF TO CYL ECHOED OK
4435  016046  001401                          BEQ     3$
4436  016050  104045                          ERROR   45              ;MR3 NOT=RKDC
4437
4438  016052  004737  046412          3$:     JSR     PC,RDCYLD       ;READ CYL DIFF
4439  016056  023737  001360  001362          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF CORRECT
4440  016064  001401                          BEQ     4$
4441  016066  104046                          ERROR   46              ;CYL DIFF IN RKMR2 INCORRECT
4442
4443  016070                          4$:
4444
4445  016070  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4446  016076  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4447  016104  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4448  016112  013737  001414  003372          MOV     T10,TEMP1
4449  016120  004737  044176                  JSR     PC,FRDY         ;FIND RDY
4450  016124  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4451  016126  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
4452  016132  000401                          BR      68$
4453  016134  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4454  016136                          68$:
4455
4456
4457  016136  104415                          SCOP1
4458  016140  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4459
4460  016144  004737  046106                  JSR     PC,SUBCLR
4461  016150  104024                          ERROR   24              ;CERR AFTER SCLR
4462
```

```
 4463  016152  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
 4464  016160  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;SET RETURN CYL ADDR
 4465  016166  013737  001350  001354          MOV     FRCYL,CCYL
 4466  016174  013737  001354  003376          MOV     CCYL,TEMP3
 4467  016202  013737  001356  003400          MOV     PCYL,TEMP4
 4468  016210  163737  003376  003400          SUB     TEMP3,TEMP4
 4469  016216  100002                          BPL     5$              ;BR IF TEMP4 IS POS
 4470  016220  005437  003400                  NEG     TEMP4
 4471  016224  013737  003400  001360  5$:     MOV     TEMP4,CALDIF
 4472  016232  013737  001354  001356          MOV     CCYL,PCYL
 4473  016240  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
 4474  016246  013737  001414  003372          MOV     T10,TEMP1
 4475  016254  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 4476  016260  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
 4477  016262  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4478  016266  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
 4479  016270  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
 4480  016276  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
 4481  016304  004737  046476                  JSR     PC,RDCYLA       ;READ CYL ADDR
 4482  016310  023737  001350  001364          CMP     FRCYL,CYLADD    ;SEE IF RETURN CYL ECHOED OK
 4483  016316  001401                          BEQ     6$
 4484  016320  104241                          ERROR   241             ;MR3 NOT=RKDC
 4485
 4486  016322  023737  001352  001350  6$:     CMP     TOCYL,FRCYL     ;SEE IF TO=FROM
 4487  016330  001022                          BNE     10$             ;DO NORMAL TEST IF NO
 4488  016332  005737  001352                  TST     TOCYL           ;SEE IF=0
 4489  016336  001007                          BNE     9$
 4490  016340  004737  046412                  JSR     PC,RDCYLD       ;CYL DIFF S/B 0 ON CYL 0
 4491  016344  005737  001362                  TST     CYLDIF
 4492  016350  001421                          BEQ     7$
 4493  016352  104242                          ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
 4494  016354  000417                          BR      7$
 4495
 4496  016356  004737  046412          9$:     JSR     PC,RDCYLD       ;CYL DIFF/OFFSET SHOULD NOT
 4497  016362  023727  001362  000001          CMP     CYLDIF,#1       ;CHANGE IN SEEK TO SELF
 4498  016370  001411                          BEQ     7$              ;SHOULD = 1 IN THIS TEST
 4499  016372  104263                          ERROR   263             ;CYL DIFF IN RKMR2 DID NOT REMAIN = 0
 4500  016374  000407                          BR      7$
 4501  016376  004737  046412          10$:    JSR     PC,RDCYLD       ;READ CYL DIFF
 4502  016402  023737  001360  001362          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF OK
 4503  016410  001401                          BEQ     7$
 4504  016412  104242                          ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
 4505
 4506  016414                          7$:
 4507
 4508  016414  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4509  016422  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4510  016430  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 4511  016436  013737  001414  003372          MOV     T10,TEMP1
 4512  016444  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 4513  016450  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4514  016452  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4515  016456  000401                          BR      69$
 4516  016460  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4517  016462                          69$:
 4518
```

```
4519
4520  016462  005237  001352              INC     TOCYL
4521  016466  023727  001352  000633      CMP     TOCYL,#411.     ;SEE IF SCANNED ALL CYLS
4522  016474  001402                      BEQ     8$              ;BR IF YES
4523  016476  000137  015702              JMP     1$              ;ELSE REPEAT
4524
4525  016502  005237  001350      8$:     INC     FRCYL
4526  016506  023727  001350  000633      CMP     FRCYL,#411.     ;SEE IF ALL DONE
4527  016514  001405                      BEQ     TST17           ;;GO TO NEXT TST
4528  016516  013737  001350  001352      MOV     FRCYL,TOCYL     ;FRCYL ALWAYS = OR > TOCYL
4529  016524  000137  015702              JMP     1$              ;ELSE REPEAT
4530  016530                      12$:
4531
4532
4533                          ;;*****************************************************************
4534                          ;*TEST 17       HEAD REGISTER TEST
4535                          ;*
4536                          ;*      THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
4537                          ;*      VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
4538                          ;*      HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.
4539                          ;*
4540                          ;*      SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK CMDS,
4541                          ;*      SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
4542                          ;*      ILLEGAL ADDRESS. IF NOT,THIS MEANS THAT CHANGING HEAD ADDRESSES
4543                          ;*      ARE NOT TIED TO SEEK CMDS
4544                          ;*
4545                          ;;*****************************************************************
4546  016530  000004      TST17:  SCOPE
4547  016532  012737  000001  001174      MOV     #1,$TIMES       ;;DO 1 ITERATION
4548  016540  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
4549
4550  016544  005000                      CLR     R0              ;HEAD #
4551  016546                      1$:
4552  016546  104415                      SCOP1
4553  016550  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
4554
4555  016554  004737  046106              JSR     PC,SUBCLR
4556  016560  104024                      ERROR   24              ;CERR AFTER SCLR
4557
4558
4559
4560  016562  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
4561  016570  013765  001222  000010      MOV     $UNIT,RKCS2(R5)
4562  016576  012765  000013  000000      MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
4563                                                                      ;RESET CYL DIFF/OFFSET & CYL ADDR REG
4564                                                                      ;IN RKMR2 & RKMR3 RESP.
4565  016604  013737  001414  003372      MOV     T10,TEMP1
4566  016612  004737  044176              JSR     PC,FRDY         ;FIND RDY
4567  016616  104124                      ERROR   124             ;RDY NOT SET AFTER RECAL CMD
4568
4569  016620  012765  000001  000026      MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4570  016626  004737  045534              JSR     PC,GSTAT
4571  016632  032737  020000  003362      BIT     #D.RTZ,HMR2
4572  016640  001001                      BNE     64$
4573  016642  104244                      ERROR   244             ;RTZ NOT SET DURING RECAL CMD
4574  016644  013737  001412  003374  64$:MOV     T1,TEMP2        ;SETUP TIMEOUT
```

```
                                        JSR     PC,FATT1        ;FIND ATTN
                                        ERROR   55              ;NO ATTN AFTER RECAL CMD

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

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                                        .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
                                        ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
                                        ERROR   275             ;MSG B0 ERROR
                                        ERROR   222             ;MSG A1 ERROR
                                        ERROR   276             ;MSG B1 ERROR

                                        TST     CYLDIF          ;SEE IF MSG A2=0
                                        BEQ     65$             ;BR IF YES
                                        ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
                                65$:    TST     CYLADD          ;SEE IF MSG B2=0
                                        BEQ     66$             ;BR IF YES
                                        ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
                                66$:

                                        MOV     #CCLR,RKCS1(R5)
                                        MOV     $UNIT,RKCS2(R5) ;DRIVE#
                                        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
                                        MOV     T10,TEMP1
                                        JSR     PC,FRDY         ;FIND RDY
                                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                                        JSR     PC,TSTATN       ;TEST FOR ATTN
                                        BR      67$
                                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                                67$:

                                        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
                                        CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
                                        CLR     E.A2            ;EXPECTED MSG A2
                                        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
                                        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
                                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
                                        ERROR   265             ;MSG B0 ERROR
                                        ERROR   274             ;MSG A1 ERROR
                                        ERROR   266             ;MSG B1 ERROR


                                        CMP     HEADA,#1        ;FOR HEAD 0, B3=1
                                        BEQ     3$
                                        ERROR   53              ;RECAL DID NOT RESET HEAD REG IN B3.
```

```
                                3$:
017130                                  SCOP1
017130  104415                          MOV     #STACK,SP       ;RESTORE STK PTR
017132  012706  001100
                                        JSR     PC,SUBCLR
017136  004737  046106                  ERROR   24              ;CERR AFTER SCLR
017142  104024
                                        SWAB    R0
017144  000300                          MOV     R0,RKDA(R5)     ;HEAD #
017146  010065  000006                  SWAB    R0
017152  000300
                                        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
017154  012765  000017  000000          MOV     T10,TEMP1
017162  013737  001414  003372          JSR     PC,FRDY         ;FIND RDY
017170  004737  044176                  ERROR   156             ;NO RDY AFTER SEEK TO SELF
017174  104156                          JSR     PC,TSTATN
017176  004737  044460                  ERROR   157             ;NO ATTN AFTER SEEK TO SELF
017202  104157                          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
017204  012737  050340  003424          CMP     R0,#3
017212  020027  000003                  BEQ     4$              ;BR FOR HEAD 3
017216  001403                          CLR     E.B0            ;FOR HEADS 0,1,2
017220  005037  003426                  BR      5$
017224  000403
017226  012737  002240  003426  4$:     MOV     #<D.SKI!D.FLT!D.IDAE>,E.B0      ;FOR HEAD 3
017234  012737  001720  003430  5$:     MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
017242  012737  000001  003432          MOV     #1,E.B1
017250  005037  003434                  CLR     E.A2
017254  012737  000002  003436          MOV     #2,E.B2
017262  005700                          TST     R0              ;SEE IF HEAD 0
017264  001004                          BNE     6$              ;BR IF NO
017266  012737  001003  003442          MOV     #<BIT9!3>,E.B3  ;LOAD EXPECTED B3 FOR HEAD 0 & MSG ID
017274  000412                          BR      8$
017276  020027  000001          6$:     CMP     R0,#1           ;SEE IF HEAD 1
017302  001004                          BNE     7$              ;BR IF NO
017304  012737  002003  003442          MOV     #<BIT10!3>,E.B3 ;B3 FOR HEAD 1
017312  000403                          BR      8$
017314  012737  004003  003442  7$:     MOV     #<BIT11!3>,E.B3 ;B3 FOR HEAD 2
017322                          8$:

017322  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
017326  000007                          .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
017330  104114                          ERROR   114             ;MSG A0 ERROR AFTER LOAD HEAD REG & SEEK CMD
017332  104115                          ERROR   115             ;MSG B0 ERROR
017334  104322                          ERROR   322             ;MSG A1 ERROR
017336  104323                          ERROR   323             ;MSG B1 ERROR

017340  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
017344  001401                          BEQ     68$             ;BR IF YES
017346  104324                          ERROR   324             ;MSG A2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
017350  005737  001364          68$:    TST     CYLADD          ;SEE IF MSG B2=0
017354  001401                          BEQ     69$             ;BR IF YES
017356  104325                          ERROR   325             ;MSG B2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
017360                          69$:

017360  020027  000003                  CMP     R0,#3
017364  001412                          BEQ     9$              ;BR IF HEAD 3
```

```
                                CLR     TEMP1
017366  005037  003372
017372  116037  003324  003372 MOVB    ATTN(R0),TEMP1
017400  023737  003372  001432 CMP     TEMP1,HEADA     ;FOR RKDA=HEAD 0, HEAD=1 IN B3
                                                       ;FOR RKDA=HEAD 1, HEAD=2 IN B3
                                                       ;FOR RKDA=HEAD 2, HEAD=4 IN B3
017406  001401                BEQ     9$
017410  104054                ERROR   54              ;HEAD DECODE IN B3 INCORRECT

017412  005200          9$:   INC     R0
017414  020027  000004        CMP     R0,#4           ;0 THRU 3 DONE?
017420  001402                BEQ     10$             ;BR IF YES
017422  000137  016546        JMP     1$              ;ELSE REPEAT

017426                  10$:

017426  012765  100000  000000 MOV     #CCLR,RKCS1(R5)
017434  013765  001222  000010 MOV     $UNIT,RKCS2(R5)
017442  012765  000013  000000 MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
017450  013737  001414  003372 MOV     T10,TEMP1
017456  004737  044176        JSR     PC,FRDY         ;FIND RDY
017462  104124                ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD

017464  012765  100000  000000 MOV     #CCLR,RKCS1(R5)
017472  013765  001222  000010 MOV     $UNIT,RKCS2(R5) ;DRIVE#
017500  012765  000005  000000 MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
017506  013737  001414  003372 MOV     T10,TEMP1
017514  004737  044176        JSR     PC,FRDY         ;FIND RDY
017520  104151                ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
017522  004737  044460        JSR     PC,TSTATN       ;TEST FOR ATTN
017526  000401                BR      71$
017530  104154                ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
017532                  71$:

017532  004737  045534        JSR     PC,GSTAT
017536  032737  000040  003364 BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
017544  001401                BEQ     70$             ;BR IF YES
017546  104155                ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD

017550  012765  100000  000000 70$:   MOV     #CCLR,RKCS1(R5)
017556  013737  001412  003374 MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
017564  004737  044512        JSR     PC,FATT1
017570  104055                ERROR   55              ;NO ATTN AFTER RECAL CMD


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

```
 4743  017572  000004
 4744  017574  012737  000001  001174  TST20:  SCOPE
 4745  017602  012706  001100                  MOV     #1,$TIMES       ;;DO 1 ITERATION
 4746                                          MOV     #STACK,SP       ;RESTORE STK PTR
 4747  017606  004737  046106
 4748  017612  104024                          JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
 4749  017614  004737  046476                  ERROR   24              ;CERR AFTER SCLR
 4750  017620  005737  001364                  JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
 4751  017624  001401                          TST     CYLADD
 4752  017626  104130                          BEQ     1$
 4753  017630                                  ERROR   130             ;CYL ADDR NOT CLEARED AFTER SCLR
 4754  017630  104415                  1$:     SCOP1
 4755  017632  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4756
 4757  017636  004737  046106                  JSR     PC,SUBCLR
 4758  017642  104024                          ERROR   24              ;CERR AFTER SCLR
 4759
 4760  017644  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD: SEEK TO SELF
 4761  017652  012737  000005  003372          MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
 4762  017660  004737  044176                  JSR     PC,FRDY         ;FIND RDY & GET STATUS
 4763  017664  104131                          ERROR   131             ;RDY NOT SET AFTER SEEK CMD
 4764  017666  012737  000005  003372          MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
 4765
 4766  017674  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
 4767  017700  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 4768  017702  032737  100000  003334          BIT     #CERR,HCS1
 4769  017710  001401                          BEQ     64$
 4770  017712  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 4771  017714                          64$:
 4772
 4773  017714  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4774  017722  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
 4775  017726  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 4776  017734  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4777  017742  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
 4778  017746  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4779  017754  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4780
 4781  017762  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 4782  017766  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4783  017770  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 4784  017772  104134                          ERROR   134             ;MSG B0 ERROR
 4785  017774  104135                          ERROR   135             ;MSG A1 ERROR
 4786  017776  104136                          ERROR   136             ;MSG B1 ERROR
 4787  020000  005737  001362                  TST     CYLDIF
 4788  020004  001401                          BEQ     65$
 4789  020006  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 4790
 4791  020010                          65$:
 4792
 4793  020010  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4794  020016  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4795  020024  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 4796  020032  013737  001414  003372          MOV     T10,TEMP1
 4797  020040  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 4798  020044  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
```

```
4799  020046  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
4800  020052  000401                  BR      66$
4801  020054  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4802  020056                  66$:
4803
4804  020056  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
4805  020064  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
4806  020070  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4807  020076  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4808  020104  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
4809  020110  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4810  020116  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4811
4812  020124  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4813  020130  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4814  020132  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4815  020134  104265                  ERROR   265             ;MSG B0 ERROR
4816  020136  104274                  ERROR   274             ;MSG A1 ERROR
4817  020140  104266                  ERROR   266             ;MSG B1 ERROR
4818
4819  020142  005737  001364          TST     CYLADD
4820  020146  001401                  BEQ     TST21           ;;GO TO NEXT TEST
4821  020150  104140                  ERROR   140             ;CYL ADDR IN B2 NOT CLEARED AFT SEEK CMD.
4822
4823                          ;;*****************************************************************
4824                          ;*TEST 21       TEST SECTOR COUNT REG. IN MSG B3
4825                          ;;*****************************************************************
4826  020152  000004          TST21:  SCOPE
4827  020154  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
4828  020162  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
4829
4830  020166  004737  046106          JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
4831  020172  104024                  ERROR   24              ;CERR AFTER SCLR
4832  020174  012737  020324  001176  MOV     #2$,$ESCAPE     ;GO TO NEXT TEST IF ANY ERROR DETECTED
4833
4834  020202  012737  000025  001400  MOV     #21.,SECNT      ;22 SECTOR FORMAT TEST
4835
4836  020210  004737  046236          JSR     PC,FSO22                ;FIND SECTOR 0
4837  020214  104142                  ERROR   142             ;SECTOR 0 NOT FOUND BY TIMEOUT
4838
4839  020216  005037  001402          CLR PSEC                ;PREVIOUS SECTOR
4840  020222  004737  046322  64$:    JSR     PC,FNS22                ;FIND NEXT SECTOR
4841  020226  104143                  ERROR   143             ;DIFFERENT SECTOR  NOT FOUND BY TIMEOUT
4842  020230  013737  001402  001404  MOV     PSEC,ESEC
4843  020236  062737  000001  001404  ADD     #1,ESEC         ;SETUP EXPECTED SECTOR
4844  020244  013737  001406  001402  MOV     SECTOR,PSEC     ;UPDATE PREV SECTOR
4845  020252  004737  046166          JSR     PC,RDSEC                ;READ SECTOR
4846  020256  023737  001406  001402  CMP     SECTOR,PSEC
4847  020264  001407                  BEQ     65$             ;BR IF READ SAME TWICE
4848  020266  004737  046166          JSR     PC,RDSEC
4849  020272  023737  001406  001402  CMP     SECTOR,PSEC
4850  020300  001401                  BEQ     65$             ;TRY 1 MORE TIME
4851  020302  104144                  ERROR   144             ;MSG B3 ERROR, SECTOR REG UNSTABLE
4852                                                          ;MAY BE DURING SECTOR PULSE TIME
4853  020304  023737  001406  001404  65$:    CMP     SECTOR,ESEC
4854  020312  001401                  BEQ     66$
```

```
                        020314  104145                  ERROR   145             ;MSG B3 ERROR BETWEEN SECTOR COUNTS
                        020316  005337  001400  66$:    DEC     SECNT
                        020322  001337                  BNE     64$             ;BR IF SECTOR COUNT NOT DONE
                        020324  005037  001176  2$:     CLR     $ESCAPE
                                                ;;**********************************************************************
                                                ;*TEST 22       DETECT OUTER LIMIT
                                                ;*
                                                ;*      THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYL 0
                                                ;*      BY DETECTING OUTER LIMIT AS THE ADJACENT CYL.
                                                ;*      AN ERROR IN THIS TEST INDICATES:
                                                ;*
                                                ;*              A.  HEADS WERE NOT ON CYL 0
                                                ;*      AND/OR  B.  COULD NOT SEEK IN REVERSE DIRECTION.
                                                ;*
                                                ;;**********************************************************************
                        020330  000004  TST22:  SCOPE
                        020332  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
                        020340  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
                        020344  004737  046106          JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
                        020350  104024                  ERROR   24              ;CERR AFTER SCLR
                        020352  005037  001410          CLR     LPFLG
                        020356  005237  001462          INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
                        020362  005237  003316          INC     UNLD            ;USED FOR VALID HALT
                        020366  012765  000020  000026  MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
                        020374  012765  000001  000020  MOV     #1,RKDC(R5)     ;CYL 1
                        020402  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD
                        020410  013737  001414  003372  MOV     T10,TEMP1
                        020416  004737  044176          JSR     PC,FRDY         ;FIND RDY
                        020422  104122                  ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
                        020424  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
                        020430  104125                  ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
                        020432  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
                        020440  012737  001200  003426  MOV     #<D.FLT!D.PAR>,E.B0
                        020446  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
                        020454  012737  000001  003432  MOV     #1,E.B1
                        020462  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                        020466  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
                        020470  104110                  ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
                        020472  104111                  ERROR   111             ;MSG B0 ERROR
                        020474  104146                  ERROR   146             ;MSG A1 ERROR
                        020476  104147                  ERROR   147             ;MSG B1 ERROR
                        020500  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
                        020506  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
                        020514  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
                        020522  013737  001414  003372  MOV     T10,TEMP1
                        020530  004737  044176          JSR     PC,FRDY         ;FIND RDY
                        020534  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                        020536  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
```

```
4911  020542  000401                        BR      64$
4912  020544  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4913  020546                          64$:
4914
4915  020546  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
4916  020554  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
4917  020560  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4918  020566  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4919  020574  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
4920  020600  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4921  020606  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4922
4923  020614  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4924  020620  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4925  020622  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4926  020624  104265                        ERROR   265             ;MSG B0 ERROR
4927  020626  104274                        ERROR   274             ;MSG A1 ERROR
4928  020630  104266                        ERROR   266             ;MSG B1 ERROR
4929
4930
4931
4932  020632  012765  000000  000020        MOV     #0,RKDC(R5)     ;CYL 0
4933  020640  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL 0
4934  020646  013737  001414  003372        MOV     T10,TEMP1
4935  020654  004737  044176                JSR     PC,FRDY         ;FIND RDY
4936  020660  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
4937  020662  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
4938  020670  004737  045534                JSR     PC,GSTAT
4939  020674  004737  046742                JSR     PC,FLIM         ;FIND LIMIT DETECT
4940  020700  104160                        ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
4941
4942
4943  020702  032737  040000  003362        BIT     #D.UNLD,HMR2
4944  020710  001003                        BNE     1$
4945  020712  104305                        ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
4946  020714  000137  021422                JMP     30$             ;BYPASS REST OF TEST
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964  020720  012737  021336  001176  1$:   MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
4965  020726  012737  070140  003424        MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
4966  020734  012737  002200  003426        MOV     #<D.SKI!D.FLT>,E.B0
```

```
4967  020742  012737  045720  003430          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4968  020750  012737  030001  003432          MOV     #<D.LIMD!D.NMOV!1>,E.B1
4969
4970  020756  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
4971  020762  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4972  020764  104161                          ERROR   161             ;MSG A0 ERROR AFTER OUTER LIMIT DETECT
4973  020766  104162                          ERROR   162             ;MSG B0 ERROR
4974  020770  104163                          ERROR   163             ;MSG A1 ERROR
4975  020772  104164                          ERROR   164             ;MSG B1 ERROR
4976
4977  020774  004737  044460                  JSR     PC,TSTATN
4978  021000  104165                          ERROR   165             ;NO ATTN AFTER OUTER LIMIT DETECT
4979  021002  005037  001462                  CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1
4980
4981  021006  004737  046106                  JSR     PC,SUBCLR       ;SUBSYS CLR
4982  021012  104024                          ERROR   24              ;CERR AFTER SCLR
4983  021014  013737  001414  003374          MOV     T10,TEMP2       ;SET UP TIMEOUT
4984  021022  004737  047020                  JSR     PC,FHDHM        ;FIND HEAD HOME
4985  021026  104166                          ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
4986  021030  004737  047074                  JSR     PC,FLOAD        ;FIND LOAD HEADS
4987  021034  104167                          ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
4988  021036  013737  001414  003374          MOV     T10,TEMP2       ;SETUP TIMEOUT
4989  021044  004737  044512                  JSR     PC,FATT1        ;FIND ATTN
4990  021050  104067                          ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
4991  021052  005037  001176          2$:     CLR     $ESCAPE
4992
4993  021056  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4994  021064  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
4995  021070  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
4996  021076  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4997  021104  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
4998  021110  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4999  021116  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5000
5001  021124  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5002  021130  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5003  021132  104063                          ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOADING
5004  021134  104064                          ERROR   64              ;MSG B0 ERROR
5005  021136  104065                          ERROR   65              ;MSG A1 ERROR
5006  021140  104066                          ERROR   66              ;MSG B1 ERROR
5007
5008  021142  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5009  021146  001401                          BEQ     65$             ;BR IF YES
5010  021150  104175                          ERROR   175             ;MSG A2 NOT CLEARED AT END OF HEAD LOADING
5011  021152  005737  001364          65$:    TST     CYLADD          ;SEE IF MSG B2=0
5012  021156  001401                          BEQ     66$             ;BR IF YES
5013  021160  104176                          ERROR   176             ;MSG B2 NOT CLEARED AT END OF HEAD LOADING
5014  021162                          66$:
5015
5016  021162  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5017  021170  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5018  021176  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5019  021204  013737  001414  003372          MOV     T10,TEMP1
5020  021212  004737  044176                  JSR     PC,FRDY         ;FIND RDY
5021  021216  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5022  021220  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
```

```
5023  021224  000401                        BR      67$
5024  021226  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5025  021230                          67$:
5026
5027  021230  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5028  021236  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
5029  021242  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5030  021250  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5031  021256  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
5032  021262  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5033  021270  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5034
5035  021276  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5036  021302  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5037  021304  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5038  021306  104265                        ERROR   265             ;MSG B0 ERROR
5039  021310  104274                        ERROR   274             ;MSG A1 ERROR
5040  021312  104266                        ERROR   266             ;MSG B1 ERROR
5041
5042  021314  005037  003316                CLR     UNLD            ;USED FOR VALID HALT
5043  021320  004737  050036                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5044  021324  000436                        BR      TST23           ;;GO TO NEXT TEST
5045                                                                ;RETURN HERE IF SW 14 IS SET OR
5046                                                                ;SW 8 WITH SWR <7:0> APPLY
5047
5048
5049  021326  005037  001176          10$:  CLR     $ESCAPE
5050  021332  000177  157550                JMP     @$LPADR
5051  021336                          20$:
5052
5053  021336  004737  046106                JSR     PC,SUBCLR
5054  021342  104024                        ERROR   24              ;CERR AFTER SCLR
5055
5056  021344  012765  000011  000000        MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
5057  021352  013737  001414  003372        MOV     T10,TEMP1       ;SET TIMEOUT
5058  021360  004737  044176                JSR     PC,FRDY         ;FIND RDY
5059  021364  104121                        ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
5060
5061  021366  013737  001416  003374        MOV     T100,TEMP2      ;SETUP  TIMEOUT
5062  021374  004737  044512                JSR     PC,FATT1        ;FIND ATTN
5063  021400  104067                        ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
5064  021402  005237  001410                INC     LPFLG
5065  021406  032777  001000  157524        BIT     #SW9,@SWR       ;LOOP ON ERROR?
5066  021414  001344                        BNE     10$             ;YES, RECONDITION DRIVE
5067  021416  000137  021052                JMP     2$              ;RETURN TO MAINLINE
5068  021422                          30$:
5069
5070
5071
5072                                ;;*****************************************************************
5073                                ;*TEST 23       READ HEADERS, ALL TRACKS, CYL 0
5074                                ;*
5075                                ;*      THIS TEST DOES NOT CHECK THE HEADERS, BUT ONLY THE FACT
5076                                ;*      THAT HEADERS CAN BE READ CORRECTLY BY CHECKING CERR & DTE.
5077                                ;*
5078                                ;*      THIS TEST IS VALID ONLY IF THE PACK HAS NOT BEEN ZERO'D OUT.
```

```
                                        ;*
                                        ;;*****************************************************************
021422  000004                  TST23:  SCOPE
021424  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
021432  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

021436  005037  001430                  CLR     HEAD            ;HEAD COUNTER

021442                          1$:
021442  104415                          SCOP1
021444  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

021450  004737  046106                  JSR     PC,SUBCLR
021454  104024                          ERROR   24              ;CERR AFTER SCLR

021456  000337  001430                  SWAB    HEAD
021462  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;LOAD HEAD REG
021470  000337  001430                  SWAB    HEAD

021474  012700  001674                  MOV     #RHTAB,R0
021500  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
021506  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
021514  004737  044176                  JSR     PC,FRDY         ;FIND RDY
021520  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
021522  032737  100000  003334          BIT     #CERR,HCS1
021530  001405                          BEQ     64$
021532  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
021534  104401  056606                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
021540  000137  043470                  JMP     $EOP            ;ABORT DRIVE

021544  016520  000024          64$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
021550  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
021554  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD

021560  032765  100000  000010          BIT     #DLT,RKCS2(R5)
021566  001407                          BEQ     65$
021570  004737  045534                  JSR     PC,GSTAT
021574  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
021576  104401  056606                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
021602  000137  043470                  JMP     $EOP            ;ABORT DRIVE
021606                          65$:

021606  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
021614  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
021620  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
021626  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
021634  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
021640  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
021646  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

021654  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
021660  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
021662  104301                          ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
021664  104271                          ERROR   271             ;MSG B0 ERROR
021666  104302                          ERROR   302             ;MSG A1 ERROR
```

```
5135  021670  104272                          ERROR   272             ;MSG B1 ERROR
5136
5137
5138  021672  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5139  021676  001401                          BEQ     66$             ;BR IF YES
5140  021700  104172                          ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5141  021702  005737  001364          66$:    TST     CYLADD          ;SEE IF MSG B2=0
5142  021706  001401                          BEQ     67$             ;BR IF YES
5143  021710  104264                          ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5144  021712                          67$:
5145  021712  005237  001430                  INC     HEAD
5146  021716  023727  001430  000003          CMP     HEAD,#3         ;ALL HEADS DONE?
5147  021724  001246                          BNE     1$              ;REPEAT IF NO
5148
5149
5150                                  ;;*****************************************************************
5151                                  ;*TEST 24       BASIC WRITE/READ HEADER & HEAD SWITCHING TEST
5152                                  ;*
5153                                  ;*      THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
5154                                  ;*      ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
5155                                  ;*      UNIQUE. 22 SECTOR FORMAT IS USED
5156                                  ;*
5157                                  ;*      I.E.    TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
5158                                  ;*              TRACK 1: 0101 FOR ALL SECTOR HEADERS
5159                                  ;*              TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS
5160                                  ;*
5161                                  ;;*****************************************************************
5162  021726  000004                  TST24:  SCOPE
5163  021730  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5164  021736  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5165
5166  021742  005237  003320                  INC     BADHDR          ;USED FOR VALID HALT
5167  021746  005037  001430                  CLR     HEAD            ;HEAD CTR
5168
5169  021752  104415                          SCOP1
5170  021754  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5171
5172  021760  004737  046106                  JSR     PC,SUBCLR
5173  021764  104024                          ERROR   24              ;CERR AFTER SCLR
5174
5175  021766  052765  000020  000010  1$:     BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
5176  021774  012765  001470  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
5177  022002  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT.
5178  022010  000337  001430                  SWAB    HEAD
5179  022014  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
5180  022022  000337  001430                  SWAB    HEAD
5181
5182  022026  013700  001430                  MOV     HEAD,R0
5183  022032  006300                          ASL     R0              ;DOUBLE R0
5184  022034  016037  001444  001470          MOV     DATA0(R0),HDTAB ;SETUP HEADER WORD FOR RKBA
5185
5186  022042  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
5187  022050  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5188  022056  004737  044176                  JSR     PC,FRDY         ;FIND RDY
5189  022062  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5190  022064  004737  045534                  JSR     PC,GSTAT        ;GET FRESH STATUS
```

```
5191  022070  032737  100000  003334          BIT     #CERR,HCS1
5192  022076  001405                          BEQ     64$
5193  022100  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
5194  022102  104401  056606                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5195  022106  000137  043470                  JMP     $EOP            ;ABORT DRIVE
5196  022112                          64$:
5197
5198  022112  012737  010340  003424          MOV     #<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5199  022120  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
5200  022124  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
5201  022132  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5202  022140  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
5203  022144  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5204  022152  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5205
5206  022160  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5207  022164  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5208  022166  104277                          ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
5209  022170  104267                          ERROR   267             ;MSG B0 ERROR
5210  022172  104300                          ERROR   300             ;MSG A1 ERROR
5211  022174  104270                          ERROR   270             ;MSG B1 ERROR
5212
5213
5214  022176  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5215  022202  001401                          BEQ     65$             ;BR IF YES
5216  022204  104303                          ERROR   303             ;MSG A2 NOT CLEARED AFTER WRITE HEADER CMD
5217  022206  005737  001364          65$:    TST     CYLADD          ;SEE IF MSG B2=0
5218  022212  001401                          BEQ     66$             ;BR IF YES
5219  022214  104304                          ERROR   304             ;MSG B2 NOT CLEARED AFTER WRITE HEADER CMD
5220  022216                          66$:
5221
5222  022216  005237  001430                  INC     HEAD
5223  022222  023727  001430  000003          CMP     HEAD,#3
5224  022230  001256                          BNE     1$
5225
5226  022232  005037  001430                  CLR     HEAD            ;HEAD CTR
5227  022236  104415                          SCOP1
5228  022240  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5229
5230  022244  004737  046106                  JSR     PC,SUBCLR
5231  022250  104024                          ERROR   24              ;CERR AFTER SCLR
5232
5233
5234
5235  022252  000337  001430          2$:     SWAB    HEAD
5236  022256  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
5237  022264  000337  001430                  SWAB    HEAD
5238
5239  022270  012700  001674                  MOV     #RHTAB,R0
5240  022274  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
5241  022302  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5242  022310  004737  044176                  JSR     PC,FRDY         ;FIND RDY
5243  022314  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
5244  022316  032737  100000  003334          BIT     #CERR,HCS1
5245  022324  001405                          BEQ     67$
5246  022326  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
```

```
5247  022330  104401  056606                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5248  022334  000137  043470                JMP     $EOP            ;ABORT DRIVE
5249
5250  022340  016520  000024        67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5251  022344  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
5252  022350  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
5253
5254
5255  022354  032765  100000  000010        BIT     #DLT,RKCS2(R5)
5256  022362  001407                        BEQ     68$
5257  022364  004737  045534                JSR     PC,GSTAT
5258  022370  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
5259  022372  104401  056606                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5260  022376  000137  043470                JMP     $EOP            ;ABORT DRIVE
5261  022402                        68$:
5262
5263  022402  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5264  022410  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
5265  022414  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5266  022422  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5267  022430  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
5268  022434  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5269  022442  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5270
5271  022450  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5272  022454  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5273  022456  104301                        ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
5274  022460  104271                        ERROR   271             ;MSG B0 ERROR
5275  022462  104302                        ERROR   302             ;MSG A1 ERROR
5276  022464  104272                        ERROR   272             ;MSG B1 ERROR
5277
5278
5279  022466  005737  001362                TST     CYLDIF          ;SEE IF MSG A2=0
5280  022472  001401                        BEQ     69$             ;BR IF YES
5281  022474  104172                        ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5282  022476  005737  001364        69$:    TST     CYLADD          ;SEE IF MSG B2=0
5283  022502  001401                        BEQ     70$             ;BR IF YES
5284  022504  104264                        ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5285  022506                        70$:
5286  022506  000337  001430                SWAB    HEAD
5287  022512  013765  001430  000006        MOV     HEAD,RKDA(R5)   ;RESTORE RKDA
5288  022520  000337  001430                SWAB    HEAD
5289
5290  022524  012701  001674                MOV     #RHTAB,R1
5291
5292  022530  005037  001442                CLR     WDCNT           ;HEADER WORD COUNT
5293  022534  013700  001430                MOV     HEAD,R0
5294  022540  006300                        ASL     R0              ;DOUBLE R0
5295  022542  016037  001444  003372        MOV     DATA0(R0),TEMP1 ;GET THE 'SHOULD BE' DATA
5296  022550  012137  001454        3$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
5297  022554  023737  001454  003372        CMP     HDWD,TEMP1
5298  022562  001401                        BEQ     4$
5299  022564  104202                        ERROR   202             ;READ HEADER MISMATCH
5300  022566  005237  001442        4$:     INC     WDCNT
5301  022572  023727  001442  000003        CMP     WDCNT,#3        ;DO ONLY 1 SECTOR
5302  022600  001363                        BNE     3$
```

```
                                                INC     HEAD
022602  005237  001430                         CMP     HEAD,#3         ;ALL 3 HEADS DONE?
022606  023727  001430  000003                 BEQ     TST25           ;;GO TO NXT TST IF YES
022614  001402                                 JMP     2$              ;ELSE REPEAT
022616  000137  022252
                                        ;;*************************************************************
                                        ;*TEST 25       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.
                                        ;*
                                        ;;*************************************************************
022622  000004                         TST25:  SCOPE
022624  012737  000001  001174                 MOV     #1,$TIMES       ;;DO 1 ITERATION
022632  012706  001100                         MOV     #STACK,SP       ;RESTORE STK PTR

022636  004737  046106                         JSR     PC,SUBCLR
022642  104024                                 ERROR   24              ;CERR AFTER SCLR
022644  052765  000020  000010                 BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
022652  012765  001450  000004                 MOV     #DATA1,RKBA(R5) ;XFER 1'S ONLY
022660  012765  177704  000002                 MOV     #-60.,RKWC(R5)  ;WORD COUNT

022666  012765  010027  000000                 MOV     #<CFMT!WRHEAD>,RKCS1(R5)        ;WRITE HEADER CMD
022674  013737  001426  003372                 MOV     T50000,TEMP1    ;SETUP TIMEOUT
022702  004737  044176                         JSR     PC,FRDY         ;FIND RDY
022706  104200                                 ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
022710  004737  045534                         JSR     PC,GSTAT        ;GET FRESH STATUS
022714  032737  100000  003334                 BIT     #CERR,HCS1
022722  001405                                 BEQ     64$
022724  104201                                 ERROR   201             ;CERR AFTER WRITE HEADER CMD
022726  104401  056606                         TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
022732  000137  043470                         JMP     $EOP            ;ABORT DRIVE
022736                                 64$:

022736  012765  010001  000000                 MOV     #<CFMT!SELDRV>,RKCS1(R5)        ;GET 20 SECTOR STATUS
022744  013737  001414  003372                 MOV     T10,TEMP1
022752  004737  044176                         JSR     PC,FRDY         ;FIND RDY
022756  104117                                 ERROR   117             ;NO RDY AFTER SELDRV CMD
022760  032737  001000  003362                 BIT     #D.FORM,HMR2
022766  001001                                 BNE     1$
022770  104312                                 ERROR   312             ;FORMAT NOT SET AFTER WRITE HDR CMD

022772                                 1$:

022772  012737  010340  003424                 MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
023000  005037  003426                         CLR     E.B0            ;EXPECTED MSG B0
023004  012737  001720  003430                 MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
023012  012737  000001  003432                 MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
023020  005037  003434                         CLR     E.A2            ;EXPECTED MSG A2
023024  012737  000002  003436                 MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
023032  012737  000003  003442                 MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
```

```
5359  023040  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5360  023044  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5361  023046  104277                  ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
5362  023050  104267                  ERROR   267             ;MSG B0 ERROR
5363  023052  104300                  ERROR   300             ;MSG A1 ERROR
5364  023054  104270                  ERROR   270             ;MSG B1 ERROR
5365
5366  023056  005037  001400          CLR     SECNT           ;SECTOR COUNT
5367  023062                  2$:
5368  023062  104415                  SCOP1
5369  023064  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
5370
5371  023070  004737  046106          JSR     PC,SUBCLR
5372  023074  104024                  ERROR   24              ;CERR AFTER SCLR
5373
5374
5375  023076  012700  001674          MOV     #RHTAB,R0
5376  023102  012765  010025  000000  MOV     #<CFMT!RDHEAD>,RKCS1(R5)        ;READ HEADER CMD
5377  023110  013737  001426  003372  MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5378  023116  004737  044176          JSR     PC,FRDY         ;FIND RDY
5379  023122  104171                  ERROR   171             ;NO RDY AFTER READ HEADER CMD
5380  023124  032737  100000  003334  BIT     #CERR,HCS1
5381  023132  001405                  BEQ     65$
5382  023134  104174                  ERROR   174             ;CERR AFTER READ HEADER CMD
5383  023136  104401  056606          TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5384  023142  000137  043470          JMP     $EOP            ;ABORT DRIVE
5385
5386  023146  016520  000024  65$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5387  023152  016520  000024          MOV     RKDB(R5),(R0)+  ;2'ND WORD
5388  023156  016520  000024          MOV     RKDB(R5),(R0)+  ;3'RD WORD
5389
5390
5391  023162  032765  100000  000010  BIT     #DLT,RKCS2(R5)
5392  023170  001407                  BEQ     66$
5393  023172  004737  045534          JSR     PC,GSTAT
5394  023176  104173                  ERROR   173             ;DLT AFTER READ HEADER CMD
5395  023200  104401  056606          TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5396  023204  000137  043470          JMP     $EOP            ;ABORT DRIVE
5397  023210                  66$:
5398
5399  023210  012765  010001  000000  MOV     #<CFMT!SELDRV>,RKCS1(R5)
5400  023216  013737  001414  003372  MOV     T10,TEMP1
5401  023224  004737  044176          JSR     PC,FRDY         ;FIND RDY
5402  023230  104117                  ERROR   117             ;NO RDY AFTER SELDRV CMD
5403  023232  032737  001000  003362  BIT     #D.FORM,HMR2
5404  023240  001001                  BNE     6$
5405  023242  104313                  ERROR   313             ;FORMAT NOT SET AFTER READ HDR CMD
5406
5407  023244                  6$:
5408
5409  023244  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5410  023252  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5411  023256  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
5412  023264  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5413  023272  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5414  023276  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
```

```
5415  023304  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5416
5417  023312  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5418  023316  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5419  023320  104301                          ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
5420  023322  104271                          ERROR   271             ;MSG B0 ERROR
5421  023324  104302                          ERROR   302             ;MSG A1 ERROR
5422  023326  104272                          ERROR   272             ;MSG B1 ERROR
5423
5424  023330  005737  001362                  TST     CYLDIF          ;SEE IF MSG A2=0
5425  023334  001401                          BEQ     67$             ;BR IF YES
5426  023336  104172                          ERROR   172             ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5427  023340  005737  001364          67$:    TST     CYLADD          ;SEE IF MSG B2=0
5428  023344  001401                          BEQ     68$             ;BR IF YES
5429  023346  104264                          ERROR   264             ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5430  023350                          68$:
5431  023350  012701  001674                  MOV     #RHTAB,R1
5432
5433  023354  005037  001442          3$:     CLR     WDCNT           ;HEADER WORD COUNT
5434  023360  013737  001450  003372          MOV     DATA1,TEMP1     ;GET 'SHOULD BE' DATA
5435  023366  012137  001454          4$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
5436  023372  023737  001454  003372          CMP     HDWD,TEMP1      ;MATCH OK?
5437  023400  001401                          BEQ     5$              ;BR IF YES
5438  023402  104202                          ERROR   202             ;READ HEADER MISMATCH
5439  023404  005237  001442          5$:     INC     WDCNT
5440  023410  023727  001442  000003          CMP     WDCNT,#3        ;JUST 1 SECTOR AND 1 HEAD
5441  023416  001363                          BNE     4$
5442
5443
5444
5445
5446                                  ;;*****************************************************************
5447                                  ;*TEST 26       WRITE & READ HEADERS CYL 0, HEAD 0
5448                                  ;;*****************************************************************
5449  023420  000004                  TST26:  SCOPE
5450  023422  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5451  023430  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5452
5453  023434  004737  046106                  JSR     PC,SUBCLR
5454  023440  104024                          ERROR   24              ;CERR AFTER SCLR
5455
5456  023442  005237  001464                  INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
5457
5458  023446  012765  001470  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
5459  023454  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT.
5460  023462  012737  000000  001352          MOV     #0,TOCYL
5461
5462  023470  013737  001352  001366          MOV     TOCYL,CALADD    ;SETUP
5463  023476  012737  000000  001430          MOV     #0,HEAD         ;TO FILL
5464  023504  012737  000000  001436          MOV     #0,FORMAT       ;HEADER
5465  023512  004737  047142                  JSR     PC,FHDTAB       ;TABLE
5466
5467  023516  012765  000000  000020          MOV     #0,RKDC(R5)     ;CYL#
5468
5469  023524  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
5470  023532  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
```

```
5471  023540  004737  044176          JSR     PC,FRDY         ;FIND RDY
5472  023544  104200                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5473  023546  004737  045534          JSR     PC,GSTAT        ;GET FRESH STATUS
5474  023552  032737  100000  003334  BIT     #CERR,HCS1
5475  023560  001405                  BEQ     64$
5476  023562  104201                  ERROR   201             ;CERR AFTER WRITE HEADER CMD
5477  023564  104401  056606          TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5478  023570  000137  043470          JMP     $EOP            ;ABORT DRIVE
5479  023574                  64$:
5480
5481  023574  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5482  023602  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5483  023606  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
5484  023614  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5485  023622  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5486  023626  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5487  023634  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5488
5489  023642  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5490  023646  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5491  023650  104277                  ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
5492  023652  104267                  ERROR   267             ;MSG B0 ERROR
5493  023654  104300                  ERROR   300             ;MSG A1 ERROR
5494  023656  104270                  ERROR   270             ;MSG B1 ERROR
5495
5496  023660  005037  001400          CLR     SECNT           ;SECTOR COUNT
5497  023664  104415                  SCOP1
5498  023666  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
5499
5500  023672  004737  046106          JSR     PC,SUBCLR
5501  023676  104024                  ERROR   24              ;CERR AFTER SCLR
5502
5503  023700  012765  000000  000020  MOV     #0,RKDC(R5)     ;CYL #
5504
5505  023706  012700  001674          MOV     #RHTAB,R0
5506
5507  023712  012765  000025  000000  65$:    MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
5508  023720  013737  001420  003372  MOV     T500,TEMP1      ;SETUP  TIMEOUT
5509  023726  004737  044176          JSR     PC,FRDY         ;FIND RDY
5510  023732  104171                  ERROR   171             ;NO RDY AFTER READ HEADER CMD
5511  023734  032737  100000  003334  BIT     #CERR,HCS1
5512  023742  001405                  BEQ     66$
5513  023744  104174                  ERROR   174             ;CERR AFTER READ HEADER  CMD
5514  023746  104401  056606          TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5515  023752  000137  043470          JMP     $EOP            ;ABORT DRIVE
5516
5517  023756  016520  000024  66$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5518  023762  016520  000024          MOV     RKDB(R5),(R0)+  ;2'ND WORD
5519  023766  016520  000024          MOV     RKDB(R5),(R0)+  ;3'RD WORD
5520
5521  023772  032765  100000  000010  BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
5522  024000  001407                  BEQ     67$
5523  024002  004737  045534          JSR     PC,GSTAT
5524  024006  104173                  ERROR   173             ;DATA LATE ON READ HEADER
5525  024010  104401  056606          TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5526  024014  000137  043470          JMP     $EOP            ;ABORT DRIVE
```

```
5527
5528  024020  020027  002100          67$:    CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
5529  024024  001332                          BNE     65$             ;BR IF NO
5530
5531  024026  004737  047464                  JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
5532                                                                  ;BEGINS WITH SECTOR 0
5533  024032  005037  001442                  CLR     WDCNT           ;WORD COUNT
5534  024036  012700  002100                  MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
5535  024042  012701  001470                  MOV     #HDTAB,R1       ;CALC HEADER TABLE
5536
5537  024046  012037  001454          68$:    MOV     (R0)+,HDWD
5538  024052  012137  003372                  MOV     (R1)+,TEMP1
5539  024056  023737  001454  003372          CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
5540  024064  001401                          BEQ     69$             ;BR IF COMPARE
5541  024066  104202                          ERROR   202             ;READ HEADER MISMATCH
5542
5543  024070  005237  001442          69$:    INC     WDCNT
5544  024074  023727  001442  000102          CMP     WDCNT,#66.      ;ALL WORDS DONE?
5545  024102  001361                          BNE     68$             ;BR IF NO
5546
5547
5548  024104  005037  001464                  CLR BYPFMT              ;ALLOW CORRECT FORMATTING
5549
5550
5551
```

```
                                ;;*****************************************************************
                                ;*TEST 27       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.
                                ;*
                                ;;*****************************************************************
024110  000004                  TST27:  SCOPE
024112  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
024120  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
024124  004737  046106                  JSR     PC,SUBCLR
024130  104024                          ERROR   24              ;CERR AFTER SCLR
024132  005037  001350                  CLR     FRCYL
024136  012737  000001  001352          MOV     #1,TOCYL
024144  012737  000001  001360          MOV     #1,CALDIF
024152  012765  000001  000020          MOV     #1,RKDC(R5)     ;SET FOR CYL 1
024160  012737  025160  001176          MOV     #10$,$ESCAPE
024166  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
024174  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
024202  004737  044176                  JSR     PC,FRDY         ;FIND RDY
024206  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
024210  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
024216  005037  003426                  CLR     E.B0
024222  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
024230  012737  000001  003432          MOV     #1,E.B1
024236  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
024242  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
024244  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
024246  104204                          ERROR   204             ;MSG B0 ERROR
024250  104205                          ERROR   205             ;MSG A1 ERROR
024252  104206                          ERROR   206             ;MSG B1 ERROR
024254  023727  001362  000001          CMP     CYLDIF,#1
024262  001401                          BEQ     1$
024264  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK CMD.
024266  012737  025200  001176  1$:     MOV     #12$,$ESCAPE
024274  013737  001422  003372          MOV     T2500,TEMP1     ;SETUP TIMEOUT
```

```
5608
5609  024302  004737  044606          JSR     PC,FATT2        ;FIND ATTN
5610  024306  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
5611  024310  032737  100000  003334  BIT     #CERR,HCS1
5612  024316  001401                  BEQ     64$
5613  024320  104210                  ERROR   210             ;CERR AFTER SEEK CMD
5614  024322                  64$:
5615
5616  024322  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5617  024330  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5618  024334  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5619  024342  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5620  024350  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5621  024354  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5622  024362  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5623
5624  024370  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5625  024374  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5626  024376  104133                  ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
5627  024400  104134                  ERROR   134             ;MSG B0 ERROR
5628  024402  104135                  ERROR   135             ;MSG A1 ERROR
5629  024404  104136                  ERROR   136             ;MSG B1 ERROR
5630  024406  005737  001362          TST     CYLDIF
5631  024412  001401                  BEQ     65$
5632  024414  104137                  ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
5633
5634  024416                  65$:
5635
5636  024416  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5637  024424  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5638  024432  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5639  024440  013737  001414  003372  MOV     T10,TEMP1
5640  024446  004737  044176          JSR     PC,FRDY         ;FIND RDY
5641  024452  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5642  024454  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
5643  024460  000401                  BR      66$
5644  024462  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5645  024464                  66$:
5646
5647  024464  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5648  024472  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5649  024476  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5650  024504  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5651  024512  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5652  024516  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5653  024524  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5654
5655  024532  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5656  024536  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5657  024540  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5658  024542  104265                  ERROR   265             ;MSG B0 ERROR
5659  024544  104274                  ERROR   274             ;MSG A1 ERROR
5660  024546  104266                  ERROR   266             ;MSG B1 ERROR
5661
5662  024550  005737  001364          TST     CYLADD
5663  024554  023727  001364  000001  CMP     CYLADD,#1
```

```
5664  024562  001401                        BEQ     2$
5665  024564  104207                        ERROR   207             ;CYL ADDR INCORRECT AFTER SEEK CMD
5666
5667
5668  024566                          2$:
5669  024566  104415                        SCOP1
5670  024570  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5671
5672  024574  004737  046106                JSR     PC,SUBCLR
5673  024600  104024                        ERROR   24              ;CERR AFTER SCLR
5674
5675  024602  005037  001176                CLR     $ESCAPE
5676  024606  012765  000001  000020        MOV     #1,RKDC(R5)     ;CYL #
5677
5678  024614  012700  001674                MOV     #RHTAB,R0
5679  024620  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
5680  024626  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5681  024634  004737  044176                JSR     PC,FRDY         ;FIND RDY
5682  024640  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
5683  024642  032737  100000  003334        BIT     #CERR,HCS1
5684  024650  001405                        BEQ     67$
5685  024652  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
5686  024654  104401  056606                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5687  024660  000137  043470                JMP     $EOP            ;ABORT DRIVE
5688
5689  024664  016520  000024        67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5690  024670  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
5691  024674  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
5692
5693
5694  024700  032765  100000  000010        BIT     #DLT,RKCS2(R5)
5695  024706  001407                        BEQ     68$
5696  024710  004737  045534                JSR     PC,GSTAT
5697  024714  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
5698  024716  104401  056606                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5699  024722  000137  043470                JMP     $EOP            ;ABORT DRIVE
5700  024726                          68$:
5701
5702
5703  024726  005737  001674                TST     RHTAB           ;CHECK 1'ST WORD ONLY:CYL#
5704  024732  001001                        BNE     3$
5705  024734  104211                        ERROR   211             ;CYL 0 HEADER ON CYL 1
5706
5707  024736  023727  001674  000001  3$:   CMP     RHTAB,#1
5708  024744  001401                        BEQ     4$
5709  024746  104202                        ERROR   202             ;READ CYL WORD HEADER ERROR
5710  024750                          4$:
5711  024750  004737  050036                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5712  024754  000521                        BR      TST30           ;;GO TO NEXT TEST
5713                                                                ;RETURN HERE IF SW 14 IS SET OR
5714                                                                ;SW 8 WITH SWR <7:0> APPLY
5715  024756  004737  046106          6$:   JSR     PC,SUBCLR
5716  024762  104024                        ERROR   24              ;CERR AFTER SCLR
5717
5718  024764  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
5719  024772  013737  001414  003372        MOV     T10,TEMP1       ;SETUP TIMEOUT
```

```
5720  025000  004737  044176          JSR     PC,FRDY         ;FIND RDY
5721  025004  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD.
5722
5723  025006  013737  001424  003372  MOV     T5000,TEMP1
5724  025014  004737  044606          JSR     PC,FATT2        ;FIND ATTN
5725  025020  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
5726  025022  032737  100000  003334  BIT     #CERR,HCS1
5727  025030  001401                  BEQ     69$
5728  025032  104210                  ERROR   210             ;CERR AFTER SEEK CMD.
5729
5730  025034  004737  046106  69$:    JSR     PC,SUBCLR
5731  025040  104024                  ERROR   24              ;CERR AFTER SCLR
5732
5733
5734  025042  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5735  025050  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5736  025054  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5737  025062  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5738  025070  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5739  025074  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5740  025102  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5741
5742  025110  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5743  025114  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5744  025116  104133                  ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
5745  025120  104134                  ERROR   134             ;MSG B0 ERROR
5746  025122  104135                  ERROR   135             ;MSG A1 ERROR
5747  025124  104136                  ERROR   136             ;MSG B1 ERROR
5748  025126  005737  001364          TST     CYLADD
5749  025132  001401                  BEQ     7$
5750  025134  104043                  ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
5751
5752  025136                  7$:
5753  025136  005037  001176          CLR     $ESCAPE
5754  025142  005737  001410          TST     LPFLG
5755  025146  001402                  BEQ     70$
5756  025150  000177  153734          JMP     @$LPERR         ;SW 9 WAS SET.
5757  025154  000177  153726  70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
5758
5759  025160                  10$:
5760  025160  005237  001410          INC     LPFLG
5761  025164  032777  001000  153746  BIT     #SW9,@SWR       ;LOOP ON ERROR?
5762  025172  001271                  BNE     6$              ;YES, RECONDITION DRIVE
5763  025174  000137  024266          JMP     1$              ;RETURN TO MAINLINE
5764  025200                  12$:
5765  025200  005237  001410          INC     LPFLG
5766  025204  032777  001000  153726  BIT     #SW9,@SWR       ;LOOP ON ERROR?
5767  025212  001261                  BNE     6$              ;YES, RECONDITION DRIVE
5768  025214  000137  024566          JMP     2$              ;RETURN TO MAINLINE
5769
5770
5771
5772
5773                    ;;*************************************************************
5774                    ;*TEST 30       WRITE & READ HEADERS CYL 1, HEAD 0
5775                    ;;*************************************************************
```

```
5776  025220  000004                    TST30:  SCOPE
5777  025222  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
5778  025230  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
5779
5780  025234  004737  046106                    JSR     PC,SUBCLR
5781  025240  104024                            ERROR   24              ;CERR AFTER SCLR
5782
5783  025242  005237  001464                    INC     BYPFMT          ;SET BIT 14 & 15  IN HEADER
5784
5785  025246  012765  001470  000004            MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
5786  025254  012765  177676  000002            MOV     #-66.,RKWC(R5)  ;WORD COUNT.
5787  025262  012737  000001  001352            MOV     #1,TOCYL
5788
5789  025270  013737  001352  001366            MOV     TOCYL,CALADD    ;SETUP
5790  025276  012737  000000  001430            MOV     #0,HEAD         ;TO FILL
5791  025304  012737  000000  001436            MOV     #0,FORMAT       ;HEADER
5792  025312  004737  047142                    JSR     PC,FHDTAB       ;TABLE
5793
5794  025316  012765  000001  000020            MOV     #1,RKDC(R5)     ;CYL#
5795
5796  025324  012765  000027  000000            MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
5797  025332  013737  001426  003372            MOV     T50000,TEMP1    ;SETUP TIMEOUT
5798  025340  004737  044176                    JSR     PC,FRDY         ;FIND RDY
5799  025344  104200                            ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5800  025346  004737  045534                    JSR     PC,GSTAT        ;GET FRESH STATUS
5801  025352  032737  100000  003334            BIT     #CERR,HCS1
5802  025360  001405                            BEQ     64$
5803  025362  104201                            ERROR   201             ;CERR AFTER WRITE HEADER CMD
5804  025364  104401  056606                    TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5805  025370  000137  043470                    JMP     $EOP            ;ABORT DRIVE
5806  025374                            64$:
5807
5808  025374  012737  010340  003424            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5809  025402  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
5810  025406  012737  001720  003430            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5811  025414  012737  000001  003432            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5812  025422  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
5813  025426  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5814  025434  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5815
5816  025442  004737  044720                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5817  025446  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5818  025450  104277                            ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
5819  025452  104267                            ERROR   267             ;MSG B0 ERROR
5820  025454  104300                            ERROR   300             ;MSG A1 ERROR
5821  025456  104270                            ERROR   270             ;MSG B1 ERROR
5822
5823  025460  005037  001400                    CLR     SECNT           ;SECTOR COUNT
5824  025464  104415                            SCOP1
5825  025466  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
5826
5827  025472  004737  046106                    JSR     PC,SUBCLR
5828  025476  104024                            ERROR   24              ;CERR AFTER SCLR
5829
5830  025500  012765  000001  000020            MOV     #1,RKDC(R5)     ;CYL #
5831
```

```
 5832  025506  012700  001674                    MOV     #RHTAB,R0
 5833
 5834  025512  012765  000025  000000  65$:      MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
 5835  025520  013737  001420  003372            MOV     T500,TEMP1      ;SETUP  TIMEOUT
 5836  025526  004737  044176                    JSR     PC,FRDY         ;FIND RDY
 5837  025532  104171                            ERROR   171             ;NO RDY AFTER READ HEADER CMD
 5838  025534  032737  100000  003334            BIT     #CERR,HCS1
 5839  025542  001405                            BEQ     66$
 5840  025544  104174                            ERROR   174             ;CERR AFTER READ HEADER  CMD
 5841  025546  104401  056606                    TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 5842  025552  000137  043470                    JMP     $EOP            ;ABORT DRIVE
 5843
 5844  025556  016520  000024          66$:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 5845  025562  016520  000024                    MOV     RKDB(R5),(R0)+  ;2'ND WORD
 5846  025566  016520  000024                    MOV     RKDB(R5),(R0)+  ;3'RD WORD
 5847
 5848  025572  032765  100000  000010            BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
 5849  025600  001407                            BEQ     67$
 5850  025602  004737  045534                    JSR     PC,GSTAT
 5851  025606  104173                            ERROR   173             ;DATA LATE ON READ HEADER
 5852  025610  104401  056606                    TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 5853  025614  000137  043470                    JMP     $EOP            ;ABORT DRIVE
 5854
 5855  025620  020027  002100          67$:      CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
 5856  025624  001332                            BNE     65$             ;BR IF NO
 5857
 5858  025626  004737  047464                    JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
 5859                                                                    ;BEGINS WITH SECTOR 0
 5860  025632  005037  001442                    CLR     WDCNT           ;WORD COUNT
 5861  025636  012700  002100                    MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
 5862  025642  012701  001470                    MOV     #HDTAB,R1       ;CALC HEADER TABLE
 5863
 5864  025646  012037  001454          68$:      MOV     (R0)+,HDWD
 5865  025652  012137  003372                    MOV     (R1)+,TEMP1
 5866  025656  023737  001454  003372            CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
 5867  025664  001401                            BEQ     69$             ;BR IF COMPARE
 5868  025666  104202                            ERROR   202             ;READ HEADER MISMATCH
 5869
 5870  025670  005237  001442          69$:      INC     WDCNT
 5871  025674  023727  001442  000102            CMP     WDCNT,#66.      ;ALL WORDS DONE?
 5872  025702  001361                            BNE     68$             ;BR IF NO
 5873
 5874
 5875  025704  005037  001464                    CLR BYPFMT              ;ALLOW CORRECT FORMATTING
 5876
 5877
 5878
 5879                          ;;*****************************************************************
 5880                          ;*TEST 31       TEST RECALIBRATE CMD & READ HEADERS
 5881                          ;*
 5882                          ;*      THIS TEST DOES A RECALIBRATE & READS HEADERS.
 5883                          ;*      IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS
 5884                          ;*      TEST DID NOT OVERWRITE CYL 0 HEADERS.
 5885                          ;*      AN ERROR IN THIS TEST INDICATES THAT HEADS:
 5886                          ;*
 5887                          ;*
```

```
                                        ;*              A. MOVED TO A CYL OTHER THAN 1
                                        ;*      OR      B. DID NOT GET BACK TO CYL 0
                                        ;*
                                        ;;*****************************************************************
025710  000004                  TST31:  SCOPE
025712  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
025720  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

025724  004737  046106                  JSR     PC,SUBCLR
025730  104024                          ERROR   24              ;CERR AFTER SCLR
025732  012737  000001  001350          MOV     #1,FRCYL        ;PARAMETERS
025740  005037  001352                  CLR     TOCYL           ;FOR
025744  012737  000001  001360          MOV     #1,CALDIF       ;ERROR TYPEOUTS
025752  012737  026776  001176          MOV     #10$,$ESCAPE
025760  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
025766  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
025774  004737  044176                  JSR     PC,FRDY         ;FIND RDY
026000  104124                          ERROR   124             ;NO RDY AFTER RECAL CMD
026002  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
026010  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
026016  004737  045534                  JSR     PC,GSTAT
026022  032737  020000  003362          BIT     #D.RTZ,HMR2
026030  001001                          BNE     1$
026032  104307                          ERROR   307             ;RTZ NOT SET DURING RECAL CMD

026034  012737  030140  003424  1$:     MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
026042  005037  003426                  CLR     E.B0
026046  012737  025720  003430          MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
026054  012737  000001  003432          MOV     #1,E.B1

026062  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
026066  000001                          .WORD   T.A2!0!0                ;& MSGS SPECIFIED HERE
026070  104213                          ERROR   213             ;MSG A0 ERROR DURING RECAL CMD
026072  104214                          ERROR   214             ;MSG B0 ERROR
026074  104215                          ERROR   215             ;MSG A1 ERROR
026076  104216                          ERROR   216             ;MSG B1 ERROR
026100  005737  001362                  TST     CYLDIF
026104  001401                          BEQ     2$
026106  104217                          ERROR   217             ;CYL DIFF INCORRECT DURING RECAL CMD.

026110  012737  027016  001176  2$:     MOV     #12$,$ESCAPE
026116  012737  177777  003372          MOV     #-1,TEMP1       ;SETUP TIMEOUT
026124  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
026130  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
026132  032737  100000  003334          BIT     #CERR,HCS1
026140  001401                          BEQ     3$
026142  104220                          ERROR   220             ;CERR AFTER RECAL CMD
026144                          3$:

026144  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
026152  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
026156  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
026164  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
026172  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
026176  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
026204  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
```

```
5944
5945  026212  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5946  026216  000003                  .WORD   T.A2!T.B2!0             ;& MSG$ SPECIFIED HERE
5947  026220  104221                  ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
5948  026222  104275                  ERROR   275             ;MSG B0 ERROR
5949  026224  104222                  ERROR   222             ;MSG A1 ERROR
5950  026226  104276                  ERROR   276             ;MSG B1 ERROR
5951
5952  026230  005737  001362          TST     CYLDIF          ;SEE IF MSG A2=0
5953  026234  001401                  BEQ     64$             ;BR IF YES
5954  026236  104047                  ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
5955  026240  005737  001364  64$:    TST     CYLADD          ;SEE IF MSG B2=0
5956  026244  001401                  BEQ     65$             ;BR IF YES
5957  026246  104050                  ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
5958  026250                  65$:
5959
5960  026250  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5961  026256  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5962  026264  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5963  026272  013737  001414  003372  MOV     T10,TEMP1
5964  026300  004737  044176          JSR     PC,FRDY         ;FIND RDY
5965  026304  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5966  026306  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
5967  026312  000401                  BR      66$
5968  026314  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5969  026316                  66$:
5970
5971  026316  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5972  026324  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
5973  026330  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
5974  026336  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5975  026344  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
5976  026350  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5977  026356  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5978
5979  026364  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
5980  026370  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5981  026372  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5982  026374  104265                  ERROR   265             ;MSG B0 ERROR
5983  026376  104274                  ERROR   274             ;MSG A1 ERROR
5984  026400  104266                  ERROR   266             ;MSG B1 ERROR
5985
5986  026402  004737  046106    4$:   JSR     PC,SUBCLR
5987  026406  104024                  ERROR   24              ;CERR AFTER SCLR
5988
5989  026410  005037  001176          CLR     $ESCAPE
5990
5991  026414  012700  001674          MOV     #RHTAB,R0
5992  026420  012765  000025  000000  MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
5993  026426  013737  001426  003372  MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5994  026434  004737  044176          JSR     PC,FRDY         ;FIND RDY
5995  026440  104171                  ERROR   171             ;NO RDY AFTER READ HEADER CMD
5996  026442  032737  100000  003334  BIT     #CERR,HCS1
5997  026450  001405                  BEQ     67$
5998  026452  104174                  ERROR   174             ;CERR AFTER READ HEADER CMD
5999  026454  104401  056606          TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
```

```
6000  026460  000137  043470                JMP     $EOP                ;ABORT DRIVE
6001
6002  026464  016520  000024        67$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
6003  026470  016520  000024                MOV     RKDB(R5),(R0)+      ;2'ND WORD
6004  026474  016520  000024                MOV     RKDB(R5),(R0)+      ;3'RD WORD
6005
6006
6007  026500  032765  100000  000010        BIT     #DLT,RKCS2(R5)
6008  026506  001407                        BEQ     68$
6009  026510  004737  045534                JSR     PC,GSTAT
6010  026514  104173                        ERROR   173                 ;DLT AFTER READ HEADER CMD
6011  026516  104401  056606                TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
6012  026522  000137  043470                JMP     $EOP                ;ABORT DRIVE
6013  026526                        68$:
6014
6015  026526  023727  001674  000001        CMP     RHTAB,#1            ;CHECK WORD 0, CYL # ONLY
6016  026534  001001                        BNE     5$
6017  026536  104240                        ERROR   240                 ;CYL 1 HEADERS ON CYL 0
6018
6019  026540  005737  001674          5$:   TST     RHTAB
6020  026544  001401                        BEQ     6$
6021  026546  104202                        ERROR   202                 ;READ CYL WORD HEADER ERROR
6022  026550                          6$:
6023  026550  004737  050036                JSR     PC,SWTST            ;SEE IF SW 14 OR 8 IS SET
6024  026554  000530                        BR      TST32               ;;GO TO NEXT TEST
6025                                                                    ;RETURN HERE IF SW 14 IS SET OR
6026                                                                    ;SW 8 WITH SWR <7:0> APPLY
6027  026556  004737  046106          8$:   JSR     PC,SUBCLR
6028  026562  104024                        ERROR   24                  ;CERR AFTER SCLR
6029  026564  012765  000001  000020        MOV     #1,RKDC(R5)         ;RECONDITION BACK TO CYL 1
6030
6031  026572  012765  000017  000000        MOV     #SEEK,RKCS1(R5)     ;SEEK CMD TO RECONDITION DRIVE.
6032  026600  013737  001414  003372        MOV     T10,TEMP1           ;SETUP TIMEOUT
6033  026606  004737  044176                JSR     PC,FRDY             ;FIND RDY
6034  026612  104131                        ERROR   131                 ;NO RDY AFTER SEEK CMD.
6035
6036  026614  013737  001424  003372        MOV     T5000,TEMP1
6037  026622  004737  044606                JSR     PC,FATT2            ;FIND ATTN
6038  026626  104132                        ERROR   132                 ;NO ATTN AFTER SEEK CMD
6039  026630  032737  100000  003334        BIT     #CERR,HCS1
6040  026636  001401                        BEQ     69$
6041  026640  104210                        ERROR   210                 ;CERR AFTER SEEK CMD.
6042
6043  026642  004737  046106          69$:  JSR     PC,SUBCLR
6044  026646  104024                        ERROR   24                  ;CERR AFTER SCLR
6045
6046
6047
6048  026650  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
6049  026656  005037  003426                CLR     E.B0                ;EXPECTED MSG B0
6050  026662  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6051  026670  012737  000001  003432        MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
6052  026676  005037  003434                CLR     E.A2                ;EXPECTED MSG A2
6053  026702  012737  000002  003436        MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
6054  026710  012737  000003  003442        MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
6055  026716  012737  000022  003436        MOV     #<BIT4!2>,E.B2      ;EXPECTED MSG B2 & ID FOR CYL 1
```

```
6056
6057  026724  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6058  026730  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6059  026732  104133                  ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6060  026734  104134                  ERROR   134             ;MSG B0 ERROR
6061  026736  104135                  ERROR   135             ;MSG A1 ERROR
6062  026740  104136                  ERROR   136             ;MSG B1 ERROR
6063  026742  023727  001364  000001  CMP     CYLADD,#1
6064  026750  001401                  BEQ     9$
6065  026752  104043                  ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
6066  026754                  9$:
6067  026754  005037  001176          CLR     $ESCAPE
6068  026760  005737  001410          TST     LPFLG
6069  026764  001402                  BEQ     70$
6070  026766  000177  152116          JMP     @$LPERR         ;SW 9 WAS SET.
6071  026772  000177  152110  70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6072  026776                  10$:
6073  026776  005237  001410          INC     LPFLG
6074  027002  032777  001000  152130  BIT     #SW9,@SWR       ;LOOP ON ERROR?
6075  027010  001262                  BNE     8$              ;YES, RECONDITION DRIVE
6076  027012  000137  026110          JMP     2$              ;RETURN TO MAINLINE
6077  027016                  12$:
6078  027016  005237  001410          INC     LPFLG
6079  027022  032777  001000  152110  BIT     #SW9,@SWR       ;LOOP ON ERROR?
6080  027030  001252                  BNE     8$              ;YES, RECONDITION DRIVE
6081  027032  000137  026402          JMP     4$              ;RETURN TO MAINLINE
6082                  ;;*****************************************************************
6083                  ;*TEST 32        SINGLE INCREMENT SEEKS TO CYL 410
6084                  ;*
6085                  ;*      THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO CYL 410
6086                  ;*      WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY
6087                  ;*      DESTROY DATA.
6088                  ;*
6089                  ;;*****************************************************************
6090  027036  000004  TST32:  SCOPE
6091  027040  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
6092  027046  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
6093
6094  027052  004737  046106          JSR     PC,SUBCLR
6095  027056  104024                  ERROR   24              ;CERR AFTER SCLR
6096  027060  005037  001350          CLR     FRCYL           ;FROM CYL
6097  027064  012737  000001  001352  MOV     #1,TOCYL        ;TO CYL
6098  027072  012737  000001  001360  MOV     #1,CALDIF       ;CALCULATED DIFF.
6099
6100  027100                  1$:
6101  027100  104415                  SCOP1
6102  027102  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
6103
6104  027106  004737  046106          JSR     PC,SUBCLR
6105  027112  104024                  ERROR   24              ;CERR AFTER SCLR
6106
6107  027114  012737  027720  001176  MOV     #10$,$ESCAPE
6108  027122  013765  001352  000020  MOV     TOCYL,RKDC(R5)  ;CYL TO SEEK TO
6109
6110  027130  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD
6111  027136  013737  001414  003372  MOV     T10,TEMP1       ;SETUP  TIMEOUT
```

```
 6112  027144  004737  044176            JSR     PC,FRDY         ;FIND RDY
 6113  027150  104131                    ERROR   131             ;NO RDY AFTER SEEK CMD
 6114  027152  012737  030140  003424    MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 6115  027160  005037  003426            CLR     E.B0
 6116  027164  012737  003720  003430    MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 6117  027172  012737  000001  003432    MOV     #1,E.B1
 6118
 6119  027200  004737  044720            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6120  027204  000003                    .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6121  027206  104203                    ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 6122  027210  104204                    ERROR   204             ;MSG B0 ERROR
 6123  027212  104205                    ERROR   205             ;MSG A1 ERROR
 6124  027214  104206                    ERROR   206             ;MSG B1 ERROR
 6125
 6126  027216  023727  001362  000001    CMP     CYLDIF,#1
 6127  027224  001401                    BEQ     2$
 6128  027226  104212                    ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
 6129
 6130  027230  012737  027740  001176  2$: MOV     #12$,$ESCAPE
 6131  027236  013737  001422  003372    MOV     T2500,TEMP1     ;SETUP TIMEOUT
 6132
 6133  027244  004737  044606            JSR     PC,FATT2        ;FIND ATTN
 6134  027250  104132                    ERROR   132             ;NO ATTN AFTER SEEK CMD
 6135  027252  032737  100000  003334    BIT     #CERR,HCS1
 6136  027260  001401                    BEQ     64$
 6137  027262  104210                    ERROR   210             ;CERR AFTER SEEK CMD
 6138  027264                          64$:
 6139
 6140  027264  012737  050340  003424    MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 6141  027272  005037  003426            CLR     E.B0            ;EXPECTED MSG B0
 6142  027276  012737  001720  003430    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
 6143  027304  012737  000001  003432    MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6144  027312  005037  0C  "34          CLR     E.A2            ;EXPECTED MSG A2
 6145  027316  012737  000002  003436    MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6146  027324  012737  000003  003442    MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6147
 6148  027332  004737  044720            JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
 6149  027336  000003                    .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 6150  027340  104133                    ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 6151  027342  104134                    ERROR   134             ;MSG B0 ERROR
 6152  027344  104135                    ERROR   135             ;MSG A1 ERROR
 6153  027346  104136                    ERROR   136             ;MSG B1 ERROR
 6154  027350  005737  001362            TST     CYLDIF
 6155  027354  001401                    BEQ     65$
 6156  027356  104137                    ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
 6157
 6158  027360                          65$:
 6159
 6160  027360  012765  100000  000000    MOV     #CCLR,RKCS1(R5)
 6161  027366  013765  001222  000010    MOV     $UNIT,RKCS2(R5) ;DRIVE#
 6162  027374  012765  000005  000000    MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 6163  027402  013737  001414  003372    MOV     T10,TEMP1
 6164  027410  004737  044176            JSR     PC,FRDY         ;FIND RDY
 6165  027414  104151                    ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 6166  027416  004737  044460            JSR     PC,TSTATN       ;TEST FOR ATTN
 6167  027422  000401                    BR      66$
```

```
                                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                                66$:
                                        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
                                        CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
                                        CLR     E.A2            ;EXPECTED MSG A2
                                        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
                                        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
                                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
                                        ERROR   265             ;MSG B0 ERROR
                                        ERROR   274             ;MSG A1 ERROR
                                        ERROR   266             ;MSG B1 ERROR

                                        CMP     CYLADD,TOCYL
                                        BEQ     3$
                                        ERROR   207             ;CYL ADDR IN RKMR2 NOT=RKDC

                                3$:     CMP     TOCYL,#410.     ;ALL CYL DONE?
                                        BEQ     4$              ;BR IF YES
                                        INC     FRCYL           ;ELSE DO ANOTHER
                                        INC     TOCYL
                                        BEQ     4$              ;BR IF YES
                                        JMP     1$

                                4$:
                                        JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
                                        BR      TST33           ;;GO TO NEXT TEST
                                                                ;RETURN HERE IF SW 14 IS SET OR
                                                                ;SW 8 WITH SWR <7:0> APPLY


                                6$:

                                        JSR     PC,SUBCLR
                                        ERROR   24              ;CERR AFTER SCRL

                                67$:    MOV     TOCYL,RKDC(R5)  ;CYL#

                                        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
                                        MOV     T10,TEMP1       ;SETUP TIMEOUT
                                        JSR     PC,FRDY         ;FIND RDY
                                        ERROR   131             ;NO RDY AFTER SEEK CMD.

                                        MOV     T5000,TEMP1
                                        JSR     PC,FATT2        ;FIND ATTN
                                        ERROR   132             ;NO ATTN AFTER SEEK CMD
                                        BIT     #CERR,HCS1
                                        BEQ     69$
                                        ERROR   210             ;CERR AFTER SEEK CMD.
```

```
6224  027644  004737  046106          69$:    JSR     PC,SUBCLR
6225  027650  104024                          ERROR   24              ;CERR AFTER SCLR
6226
6227  027652  023727  001352  000000          CMP     TOCYL,#0        ;ALL CYL DONE?
6228  027660  001403                          BEQ     68$             ;BR IF YES
6229  027662  005337  001352                  DEC     TOCYL           ;ELSE DO ANOTHER
6230  027666  000737                          BR      67$
6231
6232  027670  004737  046106          68$:    JSR     PC,SUBCLR
6233  027674  104024                          ERROR   24              ;CERR AFTER SCLR
6234
6235  027676  005037  001176                  CLR     $ESCAPE
6236  027702  005737  001410                  TST     LPFLG
6237  027706  001402                          BEQ     70$
6238  027710  000177  151174                  JMP     @$LPERR         ;SW 9 WAS SET.
6239  027714  000177  151166          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6240
6241
6242
6243  027720                          10$:
6244  027720  005237  001410                  INC     LPFLG
6245  027724  032777  001000  151206          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6246  027732  001312                          BNE     6$              ;YES, RECONDITION DRIVE
6247  027734  000137  027230                  JMP     2$              ;RETURN TO MAINLINE
6248
6249  027740                          12$:
6250  027740  005237  001410                  INC     LPFLG
6251  027744  032777  001000  151166          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6252  027752  001302                          BNE     6$              ;YES, RECONDITION DRIVE
6253  027754  000137  027552                  JMP     4$              ;RETURN TO MAINLINE
6254
6255                                  ;;*****************************************************************
6256                                  ;*TEST 33       READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
6257                                  ;*
6258                                  ;*      THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
6259                                  ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
6260                                  ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
6261                                  ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
6262                                  ;*
6263                                  ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED FOR BAD
6264                                  ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
6265                                  ;*
6266                                  ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
6267                                  ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
6268                                  ;*      A MSG WILL BE TYPED INDICATING THAT ALL
6269                                  ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
6270                                  ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
6271                                  ;*
6272                                  ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
6273                                  ;*
6274                                  ;*       THIS IS THE FIRST TEST WHERE THE READ DATA CMD IS PERFORMED
6275                                  ;*
6276                                  ;;*****************************************************************
6277  027760  000004                  TST33:  SCOPE
6278  027762  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6279  027770  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
```

```
6280
6281  027774  004737  046106                  JSR     PC,SUBCLR
6282  030000  104024                          ERROR   24              ;CERR AFTER SCLR
6283  030002  005037  003374                  CLR     TEMP2           ;SECTOR CTR
6284  030006  005037  003376                  CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
6285                                                                  ;1=22 SECTOR SOFTWARE DETECTED TABLE
6286                                                                  ;2=DONE
6287  030012  012737  002304  003400          MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE INFO
6288  030020  013765  003400  000004          MOV     TEMP4,RKBA(R5)
6289  030026  012737  001000  003402          MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
6290  030034  013765  003402  000006          MOV     TEMP5,RKDA(R5)
6291
6292  030042  012765  000632  000020  1$:     MOV     #410.,RKDC(R5)  ;CYL 410
6293  030050  012765  177400  000002          MOV     #-256.,RKWC(R5) ;LOAD WORD CT
6294  030056  012765  000021  000000          MOV     #RDDATA,RKCS1(R5);READ DATA CMD
6295  030064  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP TIMEOUT
6296  030072  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6297  030076  104226                          ERROR   226             ;NO RDY AFTER READ DATA CMD
6298  030100  004737  045534                  JSR     PC,GSTAT        ;GET FRESH STATUS
6299  030104  032737  100000  003334          BIT     #CERR,HCS1
6300  030112  001470                          BEQ     8$
6301  030114  104227                          ERROR   227             ;CERR AFTER READ DATA CMD
6302
6303  030116  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
6304  030124  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
6305  030130  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6306  030136  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6307  030144  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
6308  030150  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6309  030156  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6310
6311  030164  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6312  030170  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6313  030172  104051                          ERROR   51              ;MSG A0 ERROR AFTER READ DATA CMD
6314  030174  104052                          ERROR   52              ;MSG B0 ERROR
6315  030176  104112                          ERROR   112             ;MSG A1 ERROR
6316  030200  104113                          ERROR   113             ;MSG B1 ERROR
6317
6318  030202  004737  046106                  JSR     PC,SUBCLR
6319  030206  104024                          ERROR   24              ;CERR AFTER SUBCLR
6320
6321  030210  005237  003374                  INC     TEMP2
6322  030214  023727  003374  000005          CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
6323  030222  001007                          BNE     5$
6324  030224  005737  003376                  TST     TEMP3
6325  030230  001002                          BNE     2$
6326  030232  104233                          ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
6327  030234  000414                          BR      3$
6328  030236  104230                  2$:     ERROR   230             ;CANT READ SECTORS 10,12,14,16,18,20
6329  030240  000412                          BR      3$
6330
6331  030242  013765  003400  000004  5$:     MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
6332  030250  062737  000002  003402          ADD     #2,TEMP5        ;SETUP TO READ 2 SECTORS FROM LAST
6333  030256  013765  003402  000006          MOV     TEMP5,RKDA(R5)
6334  030264  000666                          BR      1$
6335
```

```
6336  030266  005237  001456          3$:     INC     BSERR           ;SET BSE FLAG
6337  030272  000454                          BR      TST34           ;;GO TO NEXT TEST
6338
6339  030274  005737  002312          8$:     TST     BSE22H+6        ;TEST CARTRIDGE TYPE
6340  030300  001404                          BEQ     9$              ;BRANCH IF DATA CARTRIDGE
6341  030302  104235                          ERROR   235             ;ALIGNMENT CARTRIDGE USED
6342  030304  005237  001456                  INC     BSERR           ;SET BSE ERROR FLAG
6343  030310  000426                          BR      10$
6344
6345  030312  005237  003376          9$:     INC     TEMP3
6346  030316  023727  003376  000001          CMP     TEMP3,#1
6347  030324  001020                          BNE     10$
6348  030326  005037  003374                  CLR     TEMP2
6349  030332  012737  054752  003400          MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
6350  030340  013765  003400  000004          MOV     TEMP4,RKBA(R5)
6351  030346  012737  001012  003402          MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12
6352  030354  013765  003402  000006          MOV     TEMP5,RKDA(R5)
6353  030362  000137  030042                  JMP     1$              ;REPEAT
6354
6355  030366  005737  001216          10$:    TST     $PASS
6356  030372  001014                          BNE     TST34           ;;GO TO NEXT TST IF NOT 1'ST PASS
6357  030374  104401  056571                  TYPE    ,MSG17          ;CART SERIAL #
6358  030400  012746  002304                  MOV     #BSE22H,-(SP)
6359  030404  004737  054156                  JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
6360  030410  004737  054526                  JSR     PC,$SUPRS       ;TYPE SERIAL #
6361  030414  104401  001205                  TYPE    ,$CRLF
6362  030420  104401  001205                  TYPE    ,$CRLF
6363
6364                                  ;;*****************************************************************
6365                                  ;*TEST 34       DETECT INNER LIMIT
6366                                  ;*
6367                                  ;*      THIS TEST VERIFIES THAT THE LAST CYL IN THE ABOVE
6368                                  ;*      TEST WAS 410 BY DETECTING INNER LIMIT AS THE ADJACENT CYL.
6369                                  ;*      IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON CYL 410
6370                                  ;*      & THAT BSE INFO IS NOT VALID.  THE FORMAT PACK TEST
6371                                  ;*      & ALL READ-WRITE TESTS ARE BYPASSED
6372                                  ;*      TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
6373                                  ;*      SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.
6374                                  ;*
6375                                  ;;*****************************************************************
6376  030424  000004          TST34:  SCOPE
6377  030426  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6378  030434  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6379
6380  030440  004737  046106                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
6381  030444  104024                          ERROR   24              ;CERR AFTER SCLR
6382
6383  030446  005037  001410                  CLR     LPFLG
6384  030452  005237  001462                  INC     BYPCERR         ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
6385  030456  005237  003316                  INC     UNLD            ;USED FOR VALID HALT
6386
6387  030462  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
6388  030470  012765  000631  000020          MOV     #409.,RKDC(R5)  ;CYL 409.
6389  030476  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
6390  030504  013737  001414  003372          MOV     T10,TEMP1
6391  030512  004737  044176                  JSR     PC,FRDY         ;FIND RDY
```

```
6392  030516  104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
6393  030520  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
6394  030524  104125                          ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
6395  030526  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
6396  030534  012737  001200  003426          MOV     #<D.FLT!D.PAR>,E.B0
6397  030542  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D:SSP>,E.A1
6398  030550  012737  000001  003432          MOV     #1,E.B1
6399
6400  030556  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6401  030562  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6402  030564  104110                          ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
6403  030566  104111                          ERROR   111             ;MSG B0 ERROR
6404  030570  104146                          ERROR   146             ;MSG A1 ERROR
6405  030572  104147                          ERROR   147             ;MSG B1 ERROR
6406
6407  030574  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6408  030602  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6409  030610  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6410  030616  013737  001414  003372          MOV     T10,TEMP1
6411  030624  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6412  030630  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6413  030632  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
6414  030636  000401                          BR      64$
6415  030640  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6416  030642                          64$:
6417
6418  030642  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
6419  030650  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
6420  030654  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
6421  030662  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6422  030670  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
6423  030674  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6424  030702  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6425
6426  030710  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6427  030714  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6428  030716  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6429  030720  104265                          ERROR   265             ;MSG B0 ERROR
6430  030722  104274                          ERROR   274             ;MSG A1 ERROR
6431  030724  104266                          ERROR   266             ;MSG B1 ERROR
6432
6433
6434
6435  030726  012765  000632  000020          MOV     #410.,RKDC(R5)  ;CYL 410.
6436  030734  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL 410.
6437  030742  013737  001414  003372          MOV     T10,TEMP1
6438  030750  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6439  030754  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
6440  030756  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6441  030764  004737  045534                  JSR     PC,GSTAT
6442  030770  004737  046742                  JSR     PC,FLIM         ;FIND LIMIT DETECT
6443  030774  104160                          ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
6444
6445
6446  030776  032737  040000  003362          BIT     #D.UNLD,HMR2
6447  031004  001003                          BNE     1$
```

```
031006  104305                                  ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
031010  000137  031646                          JMP     30$             ;BYPASS REST OF TEST

















031014  012737  031562  001176  1$:             MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
031022  012737  070140  003424                  MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0   ;EXPECTED A0
031030  012737  002200  003426                  MOV     #<D.SKI!D.FLT>,E.B0
031036  012737  045720  003430                  MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
031044  012737  030001  003432                  MOV     #<D.LIMD!D.NMOV!1>,E.B1

031052  004737  044720                          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
031056  000000                                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
031060  104161                                  ERROR   161             ;MSG A0 ERROR AFTER INNER LIMIT DETECT
031062  104162                                  ERROR   162             ;MSG B0 ERROR
031064  104163                                  ERROR   163             ;MSG A1 ERROR
031066  104164                                  ERROR   164             ;MSG B1 ERROR

031070  004737  044460                          JSR     PC,TSTATN
031074  104165                                  ERROR   165             ;NO ATTN AFTER INNER LIMIT DETECT
031076  005037  001462                          CLR     BYPCERR         ;ALLOW CHECKING CERR IN GSTAT1

031102  004737  046106                          JSR     PC,SUBCLR       ;SUBSYS CLR
031106  104024                                  ERROR   24              ;CERR AFTER SCLR
031110  013737  001414  003374                  MOV     T10,TEMP2       ;SET UP TIMEOUT
031116  004737  047020                          JSR     PC,FHDHM        ;FIND HEAD HOME
031122  104166                                  ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
031124  004737  047074                          JSR     PC,FLOAD        ;FIND LOAD HEADS
031130  104167                                  ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
031132  013737  001414  003374                  MOV     T10,TEMP2       ;SETUP TIMEOUT
031140  004737  044512                          JSR     PC,FATT1        ;FIND ATTN
031144  104067                                  ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
031146  005037  001176          2$:             CLR     $ESCAPE

031152  012737  050340  003424                  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
031160  005037  003426                          CLR     E.B0            ;EXPECTED MSG B0
031164  012737  001720  003430                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
031172  012737  000001  003432                  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
031200  005037  003434                          CLR     E.A2            ;EXPECTED MSG A2
031204  012737  000002  003436                  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
031212  012737  000003  003442                  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

```

```
6504  031220  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6505  031224  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6506  031226  104063                  ERROR   63              ;MSG A0 ERROR AT END OF HEAD LOADING
6507  031230  104064                  ERROR   64              ;MSG B0 ERROR
6508  031232  104065                  ERROR   65              ;MSG A1 ERROR
6509  031234  104066                  ERROR   66              ;MSG B1 ERROR
6510
6511  031236  005737  001362          TST     CYLDIF          ;SEE IF MSG A2=0
6512  031242  001401                  BEQ     65$             ;BR IF YES
6513  031244  104175                  ERROR   175             ;MSG A2 NOT CLEARED AT END OF HEAD LOADING
6514  031246  005737  001364  65$:    TST     CYLADD          ;SEE IF MSG B2=0
6515  031252  001401                  BEQ     66$             ;BR IF YES
6516  031254  104176                  ERROR   176             ;MSG B2 NOT CLEARED AT END OF HEAD LOADING
6517  031256          66$:
6518
6519  031256  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
6520  031264  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
6521  031272  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6522  031300  013737  001414  003372  MOV     T10,TEMP1
6523  031306  004737  044176          JSR     PC,FRDY         ;FIND RDY
6524  031312  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6525  031314  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
6526  031320  000401                  BR      67$
6527  031322  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6528  031324          67$:
6529
6530  031324  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
6531  031332  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
6532  031336  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6533  031344  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6534  031352  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
6535  031356  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6536  031364  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6537
6538  031372  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6539  031376  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6540  031400  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6541  031402  104265                  ERROR   265             ;MSG B0 ERROR
6542  031404  104274                  ERROR   274             ;MSG A1 ERROR
6543  031406  104266                  ERROR   266             ;MSG B1 ERROR
6544
6545  031410  005037  003316          CLR     UNLD            ;USED FOR VALID HALT
6546  031414  004737  050036          JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6547  031420  000512                  BR      TST35           ;;GO TO NEXT TEST
6548                                                          ;RETURN HERE IF SW 14 IS SET OR
6549                                                          ;SW 8 WITH SWR <7:0> APPLY
6550
6551
6552  031422          10$:
6553
6554  031422  004737  046106          JSR     PC,SUBCLR
6555  031426  104024                  ERROR   24              ;CERR AFTER SCRL
6556
6557  031430  013765  001352  000020  68$:    MOV     TOCYL,RKDC(R5)  ;CYL#
6558
6559  031436  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
```

```
6560  031444  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
6561  031452  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6562  031456  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
6563
6564  031460  013737  001424  003372          MOV     T5000,TEMP1
6565  031466  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
6566  031472  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6567  031474  032737  100000  003334          BIT     #CERR,HCS1
6568  031502  001401                          BEQ     70$
6569  031504  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
6570
6571  031506  004737  046106          70$:    JSR     PC,SUBCLR
6572  031512  104024                          ERROR   24              ;CERR AFTER SCLR
6573
6574  031514  023727  001352  000632          CMP     TOCYL,#410.     ;ALL CYL DONE?
6575  031522  001403                          BEQ     69$             ;BR IF YES
6576  031524  005237  001352                  INC     TOCYL           ;ELSE DO ANOTHER
6577  031530  000737                          BR      68$
6578
6579  031532  004737  046106          69$:    JSR     PC,SUBCLR
6580  031536  104024                          ERROR   24              ;CERR AFTER SCLR
6581
6582  031540  005037  001176                  CLR     $ESCAPE
6583  031544  005737  001410                  TST     LPFLG
6584  031550  001402                          BEQ     71$
6585  031552  000177  147332                  JMP     @$LPERR         ;SW 9 WAS SET.
6586  031556  000177  147324          71$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6587
6588  031562                          20$:
6589
6590  031562  004737  046106                  JSR     PC,SUBCLR
6591  031566  104024                          ERROR   24              ;CERR AFTER SCLR
6592
6593  031570  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
6594  031576  013737  001414  003372          MOV     T10,TEMP1       ;SET TIMEOUT
6595  031604  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6596  031610  104121                          ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
6597
6598  031612  013737  001416  003374          MOV     T100,TEMP2      ;SETUP  TIMEOUT
6599  031620  004737  044512                  JSR     PC,FATT1        ;FIND ATTN
6600  031624  104067                          ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
6601  031626  005237  001410                  INC     LPFLG
6602  031632  032777  001000  147300          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6603  031640  001270                          BNE     10$             ;YES, RECONDITION DRIVE
6604  031642  000137  031146                  JMP     2$              ;RETURN TO MAINLINE
6605  031646                          30$:
6606
6607  031646                          FORM:
6608                                  ;;*****************************************************************
6609                                  ;*TEST 35       FORMAT PACK
6610                                  ;*
6611                                  ;*      THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
6612                                  ;*      DOING 1 CYL INCREMENTAL SEEKS
6613                                  ;*      FROM 0 TO 410 WITH WRITE HEADER CMDS (ALL TRACKS).
6614                                  ;*      HEADERS WILL BE READ IN THE NEXT TEST
6615                                  ;*
```

```
                                ;;*****************************************************************
031646  000004                  TST35:  SCOPE
031650  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION

031656  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
031662  005737  001342          TST     MODTST          ;SEE IF MODULE TESTING
031666  001402                  BEQ     22$             ;BR IF NO
031670  104401  056672          TYPE    ,MSG20          ;RUNNING MODIFIED VERSION OF TEST

031674  005737  001460  22$:    TST     LIMERROR                ;CHECK IF FOUND LIMIT DETECT ERROR
031700  001403                  BEQ     1$
031702  104170                  ERROR   170             ;FORMAT TEST BYPASSED, LIMIT DETECT ERROR
031704  000137  033750          JMP     13$
031710  005737  001456  1$:     TST     BSERR           ;CHECK IF FOUND BSE INFO OK
031714  001403                  BEQ     2$
031716  104177                  ERROR   177             ;FORMAT TEST BYPASSED-BSE ERROR
031720  000137  033750          JMP     13$

031724  004737  046106  2$:     JSR     PC,SUBCLR
031730  104024                  ERROR   24              ;CERR AFTER SCLR

031732  104401  056407          TYPE    ,MSG12          ;FORMATTING PACK, PLEASE WAIT

031736  005037  001352          CLR     TOCYL

031742  013737  001352  001366  MOV     TOCYL,CALADD    ;SETUP
031750  012737  000000  001430  MOV     #0,HEAD         ;TO FILL
031756  012737  000000  001436  MOV     #0,FORMAT       ;HEADER
031764  004737  047142          JSR     PC,FHDTAB       ;TABLE

031770  012765  001470  000004  9$:     MOV     #HDTAB,RKBA(R5) ;THIS SECTION
031776  012765  177676  000002  MOV     #-66.,RKWC(R5)  ;OF CODE
032004  000337  001430          SWAB    HEAD            ;IS TO RESTORE STANDARD FORMAT
032010  013765  001430  000006  MOV     HEAD,RKDA(R5)   ;TO CYL 0
032016  000337  001430          SWAB    HEAD            ;HEAD 0,1 & 2

032022  012765  000027  000000  MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
032030  013737  001426  003372  MOV     T50000,TEMP1    ;SETUP TIMEOUT
032036  004737  044176          JSR     PC,FRDY         ;FIND RDY
032042  104200                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
032044  004737  045534          JSR     PC,GSTAT        ;GET FRESH STATUS
032050  032737  100000  003334  BIT     #CERR,HCS1
032056  001405                  BEQ     64$
032060  104201                  ERROR   201             ;CERR AFTER WRITE HEADER CMD
032062  104401  056606          TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
032066  000137  043470          JMP     $EOP            ;ABORT DRIVE
032072                  64$:
032072  005237  001430          INC     HEAD
032076  023727  001430  000003  CMP     HEAD,#3
032104  001403                  BEQ     11$             ;BR IF ALL HEADS DONE

032106  004737  047142          JSR     PC,FHDTAB
032112  000726                  BR      9$
```

```
6672
6673  032114  012737  000001  001366  11$:    MOV     #1,CALADD       ;SETUP
6674  032122  005037  001430                  CLR     HEAD            ;FOR
6675  032126  005037  001436                  CLR     FORMAT          ;FHDTAB ROUTINE
6676
6677  032132  012737  000001  001360          MOV     #1,CALDIF       ;SETUP
6678  032140  005037  001350                  CLR     FRCYL           ;FOR
6679  032144  012737  000001  001352          MOV     #1,TOCYL        ;ERROR REPORT
6680                                                                  ;START FORMATTING CYL 1 TO 410 HERE
6681
6682
6683  032152                          3$:
6684  032152  104415                          SCOP1
6685  032154  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6686
6687  032160  004737  046106                  JSR     PC,SUBCLR
6688  032164  104024                          ERROR   24              ;CERR AFTER SCLR
6689
6690  032166  005737  001342                  TST     MODTST          ;SEE IF MODULE TESTING
6691  032172  001404                          BEQ     18$             ;BR IF NO
6692  032174  012737  033710  001176          MOV     #16$,$ESCAPE
6693  032202  000403                          BR      19$
6694  032204  012737  033410  001176  18$:    MOV     #10$,$ESCAPE
6695  032212  013765  001366  000020  19$:    MOV     CALADD,RKDC(R5) ;CYL #
6696  032220  000337  001430                  SWAB    HEAD
6697  032224  013765  001430  000006          MOV     HEAD,RKDA(R5)   ;HEAD #
6698  032232  000337  001430                  SWAB    HEAD
6699
6700  032236  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
6701  032244  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
6702  032252  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6703  032256  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
6704  032260  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
6705  032266  005037  003426                  CLR     E.B0
6706  032272  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6707  032300  012737  000001  003432          MOV     #1,E.B1
6708
6709  032306  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6710  032312  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6711  032314  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
6712  032316  104204                          ERROR   204             ;MSG B0 ERROR
6713  032320  104205                          ERROR   205             ;MSG A1 ERROR
6714  032322  104206                          ERROR   206             ;MSG B1 ERROR
6715
6716  032324  023727  001362  000001          CMP     CYLDIF,#1
6717  032332  001401                          BEQ     4$
6718  032334  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
6719
6720  032336  005737  001342          4$:     TST     MODTST          ;SEE IF MODULE TESTING
6721  032342  001404                          BEQ     20$             ;BR IF NO
6722  032344  012737  033730  001176          MOV     #17$,$ESCAPE
6723  032352  000403                          BR      21$
6724
6725  032354  012737  033430  001176  20$:    MOV     #12$,$ESCAPE
6726  032362  012737  004704  003372  21$:    MOV     #2500.,TEMP1    ;SETUP TIMEOUT
6727
```

```
6728  032370  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
6729  032374  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6730  032376  032737  100000  003334          BIT     #CERR,HCS1
6731  032404  001401                          BEQ     65$
6732  032406  104210                          ERROR   210             ;CERR AFTER SEEK CMD
6733  032410                          65$:
6734
6735  032410  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
6736  032416  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
6737  032422  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
6738  032430  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6739  032436  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
6740  032442  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6741  032450  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6742
6743  032456  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6744  032462  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6745  032464  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6746  032466  104134                          ERROR   134             ;MSG B0 ERROR
6747  032470  104135                          ERROR   135             ;MSG A1 ERROR
6748  032472  104136                          ERROR   136             ;MSG B1 ERROR
6749  032474  005737  001362                  TST     CYLDIF
6750  032500  001401                          BEQ     66$
6751  032502  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6752
6753  032504                          66$:
6754
6755  032504  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6756  032512  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6757  032520  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6758  032526  013737  001414  003372          MOV     T10,TEMP1
6759  032534  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6760  032540  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6761  032542  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
6762  032546  000401                          BR      67$
6763  032550  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6764  032552                          67$:
6765
6766  032552  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
6767  032560  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
6768  032564  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
6769  032572  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6770  032600  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
6771  032604  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6772  032612  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6773
6774  032620  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6775  032624  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6776  032626  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6777  032630  104265                          ERROR   265             ;MSG B0 ERROR
6778  032632  104274                          ERROR   274             ;MSG A1 ERROR
6779  032634  104266                          ERROR   266             ;MSG B1 ERROR
6780
6781  032636  023737  001364  001366          CMP     CYLADD,CALADD
6782  032644  001401                          BEQ     5$
6783  032646  104232                          ERROR   232             ;CYL ADDR IN RKMR2 NOT=RKDC
```

```
6784
6785  032650                                  5$:
6786  032650  104415                            SCOP1
6787  032652  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
6788
6789  032656  004737  046106                    JSR     PC,SUBCLR
6790  032662  104024                            ERROR   24              ;CERR AFTER SCLR
6791
6792  032664  005037  001176                    CLR     $ESCAPE
6793  032670  004737  047142                    JSR     PC,FHDTAB       ;FILL HEADER TABLE
6794  032674  000337  001430                    SWAB    HEAD
6795  032700  013765  001430  000006            MOV     HEAD,RKDA(R5)   ;SET TRACK #
6796  032706  000337  001430                    SWAB    HEAD
6797  032712  012765  001470  000004            MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
6798  032720  012765  177676  000002            MOV     #-66.,RKWC(R5)  ;WORD CT
6799  032726  013765  001366  000020            MOV     CALADD,RKDC(R5) ;CYL #
6800
6801
6802  032734  012765  000027  000000            MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
6803  032742  013737  001426  003372            MOV     T50000,TEMP1    ;SETUP TIMEOUT
6804  032750  004737  044176                    JSR     PC,FRDY         ;FIND RDY
6805  032754  104200                            ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
6806  032756  004737  045534                    JSR     PC,GSTAT        ;GET FRESH STATUS
6807  032762  032737  100000  003334            BIT     #CERR,HCS1
6808  032770  001405                            BEQ     68$
6809  032772  104201                            ERROR   201             ;CERR AFTER WRITE HEADER CMD
6810  032774  104401  056606                    TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6811  033000  000137  043470                    JMP     $EOP            ;ABORT DRIVE
6812  033004                                  68$:
6813
6814  033004  012737  010340  003424            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
6815  033012  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
6816  033016  012737  001720  003430            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
6817  033024  012737  000001  003432            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6818  033032  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
6819  033036  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6820  033044  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6821
6822  033052  004737  044720                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
6823  033056  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
6824  033060  104277                            ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
6825  033062  104267                            ERROR   267             ;MSG B0 ERROR
6826  033064  104300                            ERROR   300             ;MSG A1 ERROR
6827  033066  104270                            ERROR   270             ;MSG B1 ERROR
6828
6829
6830  033070  005737  001342                    TST     MODTST          ;SEE IF MODULE TESTING
6831  033074  001402                            BEQ     23$             ;BR IF NO
6832  033076  000137  033450                    JMP     14$             ;ELSE RESTORE HEADERS ONLY
6833
6834  033102  005237  001430            23$:    INC     HEAD
6835  033106  023727  001430  000002            CMP     HEAD,#2
6836  033114  001006                            BNE     6$
6837  033116  023727  001366  000632            CMP     CALADD,#410.    ;HEAD 2, SEE IF CYL 410
6838  033124  001002                            BNE     6$              ;DO NOT WRITE ON CYL 410 HEAD 2
6839  033126  000137  033360                    JMP     7$
```

```
6840
6841  033132  023727  001430  000003  6$:   CMP   HEAD,#3          ;ALL HEADS DONE?
6842  033140  001243                        BNE   5$              ;BR IF NO
6843  033142  005037  001430                CLR   HEAD            ;ALL HEADS ON CYL DONE
6844  033146  005237  001366                INC   CALADD          ;GO TO NEXT CYL
6845  033152  005237  001350                INC   FRCYL           ;FOR ERROR REPORT
6846  033156  005237  001352                INC   TOCYL           ;FOR ERROR REPORT
6847  033162  005737  003322                TST   HPEND           ;SEE IF HALT PENDING
6848  033166  001002                        BNE   24$             ;BR IF YES
6849  033170  000137  032152                JMP   3$              ;ELSE KEEP FORMATTING
6850
6851  033174  005037  003322          24$:  CLR   HPEND           ;CLEAR FOR FUTURE FORMATTING
6852  033200  005037  003320                CLR   BADHDR          ;HEADERS NOW OK
6853  033204  000137  050106                JMP   STOP            ;GO & HALT THE CPU
6854
6855  033210  005037  001366          8$:   CLR   CALADD
6856  033214  005037  001350                CLR   FRCYL
6857
6858  033220  004737  046106                JSR   PC,SUBCLR
6859  033224  104024                        ERROR 24              ;CERR AFTER SCRL
6860
6861  033226  013765  001352  000020  69$:  MOV   TOCYL,RKDC(R5)  ;CYL#
6862
6863  033234  012765  000017  000000        MOV   #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6864  033242  013737  001414  003372        MOV   T10,TEMP1       ;SETUP TIMEOUT
6865  033250  004737  044176                JSR   PC,FRDY         ;FIND RDY
6866  033254  104131                        ERROR 131             ;NO RDY AFTER SEEK CMD.
6867
6868  033256  013737  001424  003372        MOV   T5000,TEMP1
6869  033264  004737  044606                JSR   PC,FATT2        ;FIND ATTN
6870  033270  104132                        ERROR 132             ;NO ATTN AFTER SEEK CMD
6871  033272  032737  100000  003334        BIT   #CERR,HCS1
6872  033300  001401                        BEQ   71$
6873  033302  104210                        ERROR 210             ;CERR AFTER SEEK CMD.
6874
6875  033304  004737  046106          71$:  JSR   PC,SUBCLR
6876  033310  104024                        ERROR 24              ;CERR AFTER SCLR
6877
6878  033312  023727  001352  000000        CMP   TOCYL,#0        ;ALL CYL DONE?
6879  033320  001403                        BEQ   70$             ;BR IF YES
6880  033322  005337  001352                DEC   TOCYL           ;ELSE DO ANOTHER
6881  033326  000737                        BR    69$
6882
6883  033330  004737  046106          70$:  JSR   PC,SUBCLR
6884  033334  104024                        ERROR 24              ;CERR AFTER SCLR
6885
6886  033336  005037  001176                CLR   $ESCAPE
6887  033342  005737  001410                TST   LPFLG
6888  033346  001402                        BEQ   72$
6889  033350  000177  145534                JMP   @$LPERR         ;SW 9 WAS SET.
6890  033354  000177  145526          72$:  JMP   @$LPADR         ;SW 14 OR 8 WAS SET
6891
6892
6893  033360                          7$:
6894  033360  004737  050036                JSR   PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6895  033364  000571                        BR    TST36           ;;GO TO NEXT TEST
```

```
6896                                                    ;RETURN HERE IF SW 14 IS SET OR
6897                                                    ;SW 8 WITH SWR <7:0> APPLY
6898  033366  005037  001176                  CLR     $ESCAPE
6899  033372  005737  001410                  TST     LPFLG
6900  033376  001402                          BEQ     73$
6901  033400  000177  145504                  JMP     @$LPERR         ;SW 9 WAS SET.
6902  033404  000177  145476          73$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6903
6904  033410                          10$:
6905  033410  005237  001410                  INC     LPFLG
6906  033414  032777  001000  145516          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6907  033422  001272                          BNE     8$              ;YES, RECONDITION DRIVE
6908  033424  000137  032336                  JMP     4$              ;RETURN TO MAINLINE
6909
6910  033430                          12$:
6911  033430  005237  001410                  INC     LPFLG
6912  033434  032777  001000  145476          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6913  033442  001262                          BNE     8$              ;YES, RECONDITION DRIVE
6914  033444  000137  032650                  JMP     5$              ;RETURN TO MAINLINE
6915
6916
6917  033450  005237  001430          14$:    INC     HEAD
6918  033454  023727  001430  000003          CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
6919  033462  001402                          BEQ     15$             ;BR IF YES TO GO BACK TO CYL 0
6920  033464  000137  032650                  JMP     5$              ;ELSE REPEAT FOR NEXT HEAD
6921
6922  033470  005065  000006          15$:    CLR     RKDA(R5)        ;SEEK TO CYL 0 & READ HEADERS
6923  033474  005037  001352                  CLR     TOCYL           ;TO RECONDITION DRIVE
6924
6925  033500  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6926  033506  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
6927  033514  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6928  033520  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
6929
6930  033522  013737  001424  003372          MOV     T5000,TEMP1
6931  033530  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
6932  033534  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6933  033536  032737  100000  003334          BIT     #CERR,HCS1
6934  033544  001401                          BEQ     74$
6935  033546  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
6936
6937  033550  004737  046106          74$:    JSR     PC,SUBCLR
6938  033554  104024                          ERROR   24              ;CERR AFTER SCLR
6939
6940
6941
6942  033556  012700  001674                  MOV     #RHTAB,R0
6943  033562  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
6944  033570  013737  001426  003372          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
6945  033576  004737  044176                  JSR     PC,FRDY         ;FIND RDY
6946  033602  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
6947  033604  032737  100000  003334          BIT     #CERR,HCS1
6948  033612  001405                          BEQ     76$
6949  033614  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
6950  033616  104401  056606                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
6951  033622  000137  043470                  JMP     $EOP            ;ABORT DRIVE
```

```
                                
033626  016520  000024          76$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
033632  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
033636  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
                                
                                
033642  032765  100000  000010        BIT     #DLT,RKCS2(R5)
033650  001407                        BEQ     77$
033652  004737  045534                JSR     PC,GSTAT
033656  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
033660  104401  056606                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
033664  000137  043470                JMP     $EOP            ;ABORT DRIVE
033670                          77$:
                                
033670  023737  001674  001352        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
033676  001401                        BEQ     75$             ;BR IF SAME
033700  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
033702                          75$:
                                
033702  004737  050036                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
033706  000420                        BR      TST36           ;;GO TO NEXT TEST
                                                            ;RETURN HERE IF SW 14 IS SET OR
                                                            ;SW 8 WITH SWR <7:0> APPLY
033710                          16$:
033710  005237  001410                INC     LPFLG
033714  032777  001000  145216        BIT     #SW9,@SWR       ;LOOP ON ERROR?
033722  001262                        BNE     15$             ;YES, RECONDITION DRIVE
033724  000137  032336                JMP     4$              ;RETURN TO MAINLINE
033730                          17$:
033730  005237  001410                INC     LPFLG
033734  032777  001000  145176        BIT     #SW9,@SWR       ;LOOP ON ERROR?
033742  001252                        BNE     15$             ;YES, RECONDITION DRIVE
033744  000137  032650                JMP     5$              ;RETURN TO MAINLINE
                                
                                
033750                          13$:
                                
                                
                                
                                ;;*****************************************************************
                                ;*TEST 36       DECREMENT FROM CYL 410 TO 0 & READ HEADERS
                                ;*
                                ;*      THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
                                ;*      SINGLE CYL INCREMENTAL SEEKS.
                                ;*
                                ;;*****************************************************************
033750  000004                  TST36:  SCOPE
033752  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
033760  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
                                
033764  104401  057121                TYPE    ,MSG22          ;FORMATTING FINISHED
033770  005737  001342                TST     MODTST          ;SEE IF MODULE TESTING
033774  001404                        BEQ     5$              ;BR IF NO
033776  104401  057041                TYPE    ,MSG21          ;BYP TESTS 36,40,41
034002  000137  035044                JMP     13$
034006  012737  000632  001350  5$:   MOV     #410.,FRCYL     ;FROM CYL
```

```
7008  034014  012737  000631  001352          MOV     #409.,TOCYL     ;TO CYL
7009
7010  034022                          1$:
7011  034022  104415                          SCOP1
7012  034024  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7013
7014  034030  004737  046106                  JSR     PC,SUBCLR
7015  034034  104024                          ERROR   24              ;CERR AFTER SCLR
7016
7017  034036  012737  035004  001176          MOV     #10$,$ESCAPE
7018  034044  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
7019
7020  034052  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7021  034060  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
7022  034066  004737  044176                  JSR     PC,FRDY         ;FIND RDY
7023  034072  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7024  034074  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7025  034102  005037  003426                  CLR     E.B0
7026  034106  012737  005720  003430          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7027  034114  012737  000001  003432          MOV     #1,E.B1
7028
7029  034122  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7030  034126  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7031  034130  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7032  034132  104204                          ERROR   204             ;MSG B0 ERROR
7033  034134  104205                          ERROR   205             ;MSG A1 ERROR
7034  034136  104206                          ERROR   206             ;MSG B1 ERROR
7035
7036  034140  023727  001362  000001          CMP     CYLDIF,#1
7037  034146  001401                          BEQ     2$
7038  034150  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
7039
7040  034152  012737  035024  001176  2$:     MOV     #12$,$ESCAPE
7041  034160  012737  004704  003372          MOV     #2500.,TEMP1    ;SETUP TIMEOUT
7042
7043  034166  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
7044  034172  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7045  034174  032737  100000  003334          BIT     #CERR,HCS1
7046  034202  001401                          BEQ     64$
7047  034204  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7048  034206                          64$:
7049
7050  034206  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7051  034214  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7052  034220  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7053  034226  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7054  034234  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7055  034240  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7056  034246  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7057
7058  034254  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7059  034260  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7060  034262  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7061  034264  104134                          ERROR   134             ;MSG B0 ERROR
7062  034266  104135                          ERROR   135             ;MSG A1 ERROR
7063  034270  104136                          ERROR   136             ;MSG B1 ERROR
```

```
7064  034272  005737  001362          TST     CYLDIF
7065  034276  001401                  BEQ     65$
7066  034300  104137                  ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7067
7068  034302                  65$:
7069
7070  034302  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
7071  034310  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
7072  034316  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7073  034324  013737  001414  003372  MOV     T10,TEMP1
7074  034332  004737  044176          JSR     PC,FRDY         ;FIND RDY
7075  034336  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7076  034340  004737  044460          JSR     PC,TSTATN       ;TEST FOR ATTN
7077  034344  000401                  BR      66$
7078  034346  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7079  034350                  66$:
7080
7081  034350  012737  010340  003424  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7082  034356  005037  003426          CLR     E.B0            ;EXPECTED MSG B0
7083  034362  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7084  034370  012737  000001  003432  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7085  034376  005037  003434          CLR     E.A2            ;EXPECTED MSG A2
7086  034402  012737  000002  003436  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7087  034410  012737  000003  003442  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7088
7089  034416  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7090  034422  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7091  034424  104273                  ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7092  034426  104265                  ERROR   265             ;MSG B0 ERROR
7093  034430  104274                  ERROR   274             ;MSG A1 ERROR
7094  034432  104266                  ERROR   266             ;MSG B1 ERROR
7095
7096  034434  023737  001364  001352  CMP     CYLADD,TOCYL
7097  034442  001401                  BEQ     3$
7098  034444  104207                  ERROR   207             ;CYL ADDR IN RKMR3 NOT = RKDC
7099
7100  034446                  3$:
7101  034446  104415                  SCOP1
7102  034450  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
7103
7104  034454  004737  046106          JSR     PC,SUBCLR
7105  034460  104024                  ERROR   24              ;CERR AFTER SCLR
7106
7107  034462  005037  001176          CLR     $ESCAPE
7108
7109  034466  013765  001352  000020  MOV     TOCYL,RKDC(R5)  ;CYL #
7110
7111
7112  034474  012700  001674          MOV     #RHTAB,R0
7113  034500  012765  000025  000000  MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7114  034506  013737  001426  003372  MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7115  034514  004737  044176          JSR     PC,FRDY         ;FIND RDY
7116  034520  104171                  ERROR   171             ;NO RDY AFTER READ HEADER CMD
7117  034522  032737  100000  003334  BIT     #CERR,HCS1
7118  034530  001405                  BEQ     68$
7119  034532  104174                  ERROR   174             ;CERR AFTER READ HEADER CMD
```

```
                                                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
                                                JMP     $EOP            ;ABORT DRIVE

                                68$:            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     69$
                                                JSR     PC,GSTAT
                                                ERROR   173             ;DLT AFTER READ HEADER CMD
                                                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                                                JMP     $EOP            ;ABORT DRIVE
                                69$:

                                                CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
                                                BEQ     67$             ;BR IF SAME
                                                ERROR   310             ;READ CYL WORD HEADER ERROR
                                67$:

                                                DEC     FRCYL
                                                BEQ     4$
                                                DEC     TOCYL
                                                JMP     1$

                                4$:
                                                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
                                                BR      TST37          ;;GO TO NEXT TEST
                                                                        ;RETURN HERE IF SW 14 IS SET OR
                                                                        ;SW 8 WITH SWR <7:0> APPLY


                                6$:

                                                JSR     PC,SUBCLR
                                                ERROR   24              ;CERR AFTER SCRL

                                70$:            MOV     TOCYL,RKDC(R5)  ;CYL#

                                                MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
                                                MOV     T10,TEMP1       ;SETUP TIMEOUT
                                                JSR     PC,FRDY         ;FIND RDY
                                                ERROR   131             ;NO RDY AFTER SEEK CMD.

                                                MOV     T5000,TEMP1
                                                JSR     PC,FATT2        ;FIND ATTN
                                                ERROR   132             ;NO ATTN AFTER SEEK CMD
                                                BIT     #CERR,HCS1
                                                BEQ     72$
                                                ERROR   210             ;CERR AFTER SEEK CMD.

                                72$:            JSR     PC,SUBCLR
                                                ERROR   24              ;CERR AFTER SCLR

                                                CMP     TOCYL,#410.     ;ALL CYL DONE?
```

```
 7176  034744  001403                        BEQ     71$             ;BR IF YES
 7177  034746  005237  001352                INC     TOCYL           ;ELSE DO ANOTHER
 7178  034752  000737                        BR      70$
 7179
 7180  034754  004737  046106        71$:    JSR     PC,SUBCLR
 7181  034760  104024                        ERROR   24              ;CERR AFTER SCLR
 7182
 7183  034762  005037  001176                CLR     $ESCAPE
 7184  034766  005737  001410                TST     LPFLG
 7185  034772  001402                        BEQ     73$
 7186  034774  000177  144110                JMP     @$LPERR         ;SW 9 WAS SET.
 7187  035000  000177  144102        73$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 7188
 7189
 7190
 7191  035004                        10$:
 7192  035004  005237  001410                INC     LPFLG
 7193  035010  032777  001000  144122        BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7194  035016  001312                        BNE     6$              ;YES, RECONDITION DRIVE
 7195  035020  000137  034152                JMP     2$              ;RETURN TO MAINLINE
 7196
 7197  035024                        12$:
 7198  035024  005237  001410                INC     LPFLG
 7199  035030  032777  001000  144102        BIT     #SW9,@SWR       ;LOOP ON ERROR?
 7200  035036  001302                        BNE     6$              ;YES, RECONDITION DRIVE
 7201  035040  000137  034446                JMP     3$              ;RETURN TO MAINLINE
 7202
 7203  035044                        13$:
 7204
 7205
 7206                                  ;;*****************************************************************
 7207                                  ;*TEST 37       SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS
 7208                                  ;*
 7209                                  ;*      THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLS & READS HEADERS.
 7210                                  ;*      IT THEN SEEKS CYL 0 & READS HEADERS.
 7211                                  ;*
 7212                                  ;*      MAJOR CYLS ARE: 1  (DECIMAL)  =  1 (OCTAL)
 7213                                  ;*                      2                2
 7214                                  ;*                      4                4
 7215                                  ;*                      8                10
 7216                                  ;*                      16               20
 7217                                  ;*                      32               40
 7218                                  ;*                      64               100
 7219                                  ;*                      128              200
 7220                                  ;*                      256              400
 7221                                  ;*
 7222                                  ;;*****************************************************************
 7223  035044  000004                TST37:  SCOPE
 7224  035046  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
 7225  035054  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 7226
 7227  035060  012737  000000  001350        MOV     #0,FRCYL        ;SETUP FROM CYL
 7228  035066  012737  000001  001352        MOV     #1,TOCYL        ;SETUP TO CYL
 7229
 7230  035074                        1$:
 7231  035074  104415                        SCOP1
```

```
7232  035076  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
7233
7234  035102  004737  046106                JSR     PC,SUBCLR
7235  035106  104024                        ERROR   24              ;CERR AFTER SCLR
7236
7237  035110  012737  036670  001176        MOV     #10$,$ESCAPE
7238  035116  013737  001350  003376        MOV     FRCYL,TEMP3     ;SETUP
7239  035124  013737  001352  003400        MOV     TOCYL,TEMP4     ;CYL DIFF
7240  035132  163737  003376  003400        SUB     TEMP3,TEMP4     ;FOR
7241  035140  013737  003400  001360        MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
7242
7243  035146  013765  001352  000020        MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
7244
7245  035154  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7246  035162  013737  001414  003372        MOV     T10,TEMP1       ;SETUP  TIMEOUT
7247  035170  004737  044176                JSR     PC,FRDY         ;FIND RDY
7248  035174  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
7249  035176  012737  030140  003424        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7250  035204  005037  003426                CLR     E.B0
7251  035210  012737  003720  003430        MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7252  035216  012737  000001  003432        MOV     #1,E.B1
7253
7254  035224  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7255  035230  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7256  035232  104203                        ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7257  035234  104204                        ERROR   204             ;MSG B0 ERROR
7258  035236  104205                        ERROR   205             ;MSG A1 ERROR
7259  035240  104206                        ERROR   206             ;MSG B1 ERROR
7260
7261  035242  012737  036710  001176  2$:   MOV     #12$,$ESCAPE
7262  035250  013737  001424  003372        MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7263
7264  035256  004737  044606                JSR     PC,FATT2        ;FIND ATTN
7265  035262  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
7266  035264  032737  100000  003334        BIT     #CERR,HCS1
7267  035272  001401                        BEQ     64$
7268  035274  104210                        ERROR   210             ;CERR AFTER SEEK CMD
7269  035276                        64$:
7270
7271  035276  012737  050340  003424        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7272  035304  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
7273  035310  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7274  035316  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7275  035324  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
7276  035330  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7277  035336  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7278
7279  035344  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7280  035350  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7281  035352  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7282  035354  104134                        ERROR   134             ;MSG B0 ERROR
7283  035356  104135                        ERROR   135             ;MSG A1 ERROR
7284  035360  104136                        ERROR   136             ;MSG B1 ERROR
7285  035362  005737  001362                TST     CYLDIF
7286  035366  001401                        BEQ     65$
7287  035370  104137                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
```

```
7288
7289  035372                                  65$:
7290
7291  035372  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
7292  035400  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
7293  035406  012765  000005  000000            MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7294  035414  013737  001414  003372            MOV     T10,TEMP1
7295  035422  004737  044176                    JSR     PC,FRDY         ;FIND RDY
7296  035426  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7297  035430  004737  044460                    JSR     PC,TSTATN       ;TEST FOR ATTN
7298  035434  000401                            BR      66$
7299  035436  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7300  035440                                  66$:
7301
7302  035440  012737  010340  003424            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7303  035446  005037  003426                    CLR     E.B0            ;EXPECTED MSG B0
7304  035452  012737  001720  003430            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7305  035460  012737  000001  003432            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7306  035466  005037  003434                    CLR     E.A2            ;EXPECTED MSG A2
7307  035472  012737  000002  003436            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7308  035500  012737  000003  003442            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7309
7310  035506  004737  044720                    JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7311  035512  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7312  035514  104273                            ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7313  035516  104265                            ERROR   265             ;MSG B0 ERROR
7314  035520  104274                            ERROR   274             ;MSG A1 ERROR
7315  035522  104266                            ERROR   266             ;MSG B1 ERROR
7316
7317  035524  023737  001364  001352            CMP     CYLADD,TOCYL
7318  035532  001401                            BEQ     3$
7319  035534  104207                            ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
7320
7321  035536                                  3$:
7322  035536  104415                            SCOP1
7323  035540  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
7324
7325  035544  004737  046106                    JSR     PC,SUBCLR
7326  035550  104024                            ERROR   24              ;CERR AFTER SCLR
7327
7328  035552  005037  001176                    CLR     $ESCAPE
7329  035556  013765  001352  000020            MOV     TOCYL,RKDC(R5)  ;CYL #
7330
7331
7332  035564  012700  001674                    MOV     #RHTAB,R0
7333  035570  012765  000025  000000            MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7334  035576  013737  001426  003372            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7335  035604  004737  044176                    JSR     PC,FRDY         ;FIND RDY
7336  035610  104171                            ERROR   171             ;NO RDY AFTER READ HEADER CMD
7337  035612  032737  100000  003334            BIT     #CERR,HCS1
7338  035620  001405                            BEQ     68$
7339  035622  104174                            ERROR   174             ;CERR AFTER READ HEADER CMD
7340  035624  104401  056606                    TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7341  035630  000137  043470                    JMP     $EOP            ;ABORT DRIVE
7342
7343  035634  016520  000024          68$:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
```

```
7344  035640  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
7345  035644  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
7346
7347
7348  035650  032765  100000  000010          BIT     #DLT,RKCS2(R5)
7349  035656  001407                          BEQ     69$
7350  035660  004737  045534                  JSR     PC,GSTAT
7351  035664  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
7352  035666  104401  056606                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7353  035672  000137  043470                  JMP     $EOP            ;ABORT DRIVE
7354  035676                          69$:
7355
7356  035676  023737  001674  001352          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7357  035704  001401                          BEQ     67$             ;BR IF SAME
7358  035706  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
7359  035710                          67$:
7360
7361
7362  035710  104415                          SCOP1
7363  035712  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7364
7365  035716  004737  046106                  JSR     PC,SUBCLR
7366  035722  104024                          ERROR   24              ;CERR AFTER SCLR
7367
7368  035724  012737  036730  001176          MOV     #14$,$ESCAPE
7369  035732  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
7370  035740  013737  001350  001354          MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRJERROR ROUTINE
7371
7372  035746  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7373  035754  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
7374  035762  004737  044176                  JSR     PC,FRDY         ;FIND RDY
7375  035766  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7376  035770  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7377  035776  005037  003426                  CLR     E.B0
7378  036002  012737  005720  003430          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7379  036010  012737  000001  003432          MOV     #1,E.B1
7380
7381  036016  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7382  036022  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7383  036024  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7384  036026  104204                          ERROR   204             ;MSG B0 ERROR
7385  036030  104205                          ERROR   205             ;MSG A1 ERROR
7386  036032  104206                          ERROR   206             ;MSG B1 ERROR
7387
7388
7389  036034  012737  036750  001176  4$:     MOV     #16$,$ESCAPE
7390  036042  013737  001424  003372          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7391
7392  036050  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
7393  036054  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7394  036056  032737  100000  003334          BIT     #CERR,HCS1
7395  036064  001401                          BEQ     70$
7396  036066  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7397  036070                          70$:
7398
7399  036070  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
```

```
7400  036076  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7401  036102  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7402  036110  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7403  036116  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7404  036122  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7405  036130  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7406
7407  036136  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7408  036142  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7409  036144  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7410  036146  104134                          ERROR   134             ;MSG B0 ERROR
7411  036150  104135                          ERROR   135             ;MSG A1 ERROR
7412  036152  104136                          ERROR   136             ;MSG B1 ERROR
7413  036154  005737  001362                  TST     CYLDIF
7414  036160  001401                          BEQ     71$
7415  036162  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7416
7417  036164                          71$:
7418
7419  036164  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7420  036172  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7421  036200  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7422  036206  013737  001414  003372          MOV     T10,TEMP1
7423  036214  004737  044176                  JSR     PC,FRDY         ;FIND RDY
7424  036220  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7425  036222  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
7426  036226  000401                          BR      72$
7427  036230  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7428  036232                          72$:
7429
7430  036232  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
7431  036240  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7432  036244  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7433  036252  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7434  036260  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7435  036264  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7436  036272  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7437
7438  036300  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7439  036304  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7440  036306  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7441  036310  104265                          ERROR   265             ;MSG B0 ERROR
7442  036312  104274                          ERROR   274             ;MSG A1 ERROR
7443  036314  104266                          ERROR   266             ;MSG B1 ERROR
7444
7445  036316  023737  001364  001350          CMP     CYLADD,FRCYL
7446  036324  001401                          BEQ     5$
7447  036326  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7448
7449  036330                          5$:
7450  036330  104415                          SCOP1
7451  036332  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7452
7453  036336  004737  046106                  JSR     PC,SUBCLR
7454  036342  104024                          ERROR   24              ;CERR AFTER SCLR
7455
```

```
7456  036344  005037  001176                      CLR     $ESCAPE
7457  036350  013765  001350  000020              MOV     FRCYL,RKDC(R5)  ;CYL #
7458
7459
7460  036356  012700  001674                      MOV     #RHTAB,R0
7461  036362  012765  000025  000000              MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7462  036370  013737  001426  003372              MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7463  036376  004737  044176                      JSR     PC,FRDY         ;FIND RDY
7464  036402  104171                              ERROR   171             ;NO RDY AFTER READ HEADER CMD
7465  036404  032737  100000  003334              BIT     #CERR,HCS1
7466  036412  001405                              BEQ     74$
7467  036414  104174                              ERROR   174             ;CERR AFTER READ HEADER CMD
7468  036416  104401  056606                      TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7469  036422  000137  043470                      JMP     $EOP            ;ABORT DRIVE
7470
7471  036426  016520  000024          74$:        MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7472  036432  016520  000024                      MOV     RKDB(R5),(R0)+  ;2'ND WORD
7473  036436  016520  000024                      MOV     RKDB(R5),(R0)+  ;3'RD WORD
7474
7475
7476  036442  032765  100000  000010              BIT     #DLT,RKCS2(R5)
7477  036450  001407                              BEQ     75$
7478  036452  004737  045534                      JSR     PC,GSTAT
7479  036456  104173                              ERROR   173             ;DLT AFTER READ HEADER CMD
7480  036460  104401  056606                      TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7481  036464  000137  043470                      JMP     $EOP            ;ABORT DRIVE
7482  036470                          75$:
7483
7484  036470  023737  001674  001350              CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
7485  036476  001401                              BEQ     73$             ;BR IF SAME
7486  036500  104311                              ERROR   311             ;READ CYL WORD HEADER ERROR
7487  036502                          73$:
7488
7489
7490  036502  023727  001352  000400              CMP     TOCYL,#400      ;ALL CYL DONE?
7491  036510  001404                              BEQ     6$              ;BR IF YES
7492  036512  006337  001352                      ASL     TOCYL           ;ELSE DO ANOTHER
7493  036516  000137  035074                      JMP     1$
7494  036522                          6$:
7495  036522  004737  050036                      JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7496  036526  000520                              BR      TST40           ;;GO TO NEXT TEST
7497                                              ;RETURN HERE IF SW 14 IS SET OR
7498                                              ;SW 8 WITH SWR <7:0> APPLY
7499  036530                          8$:
7500
7501  036530  004737  046106                      JSR     PC,SUBCLR
7502  036534  104024                              ERROR   24              ;CERR AFTER SCRL
7503
7504  036536  013765  001352  000020  76$:        MOV     TOCYL,RKDC(R5)  ;CYL#
7505
7506  036544  012765  000017  000000              MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7507  036552  013737  001414  003372              MOV     T10,TEMP1       ;SETUP TIMEOUT
7508  036560  004737  044176                      JSR     PC,FRDY         ;FIND RDY
7509  036564  104131                              ERROR   131             ;NO RDY AFTER SEEK CMD.
7510
7511  036566  013737  001424  003372              MOV     T5000,TEMP1
```

```
7512  036574  004737  044606          JSR     PC,FATT2        ;FIND ATTN
7513  036600  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
7514  036602  032737  100000  003334  BIT     #CERR,HCS1
7515  036610  001401                  BEQ     78$
7516  036612  104210                  ERROR   210             ;CERR AFTER SEEK CMD.
7517
7518  036614  004737  046106  78$:    JSR     PC,SUBCLR
7519  036620  104024                  ERROR   24              ;CERR AFTER SCLR
7520
7521  036622  023727  001352  000000  CMP     TOCYL,#0        ;ALL CYL DONE?
7522  036630  001403                  BEQ     77$             ;BR IF YES
7523  036632  005337  001352          DEC     TOCYL           ;ELSE DO ANOTHER
7524  036636  000737                  BR      76$
7525
7526  036640  004737  046106  77$:    JSR     PC,SUBCLR
7527  036644  104024                  ERROR   24              ;CERR AFTER SCLR
7528
7529  036646  005037  001176          CLR     $ESCAPE
7530  036652  005737  001410          TST     LPFLG
7531  036656  001402                  BEQ     79$
7532  036660  000177  142224          JMP     @$LPERR         ;SW 9 WAS SET.
7533  036664  000177  142216  79$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7534
7535  036670          10$:
7536  036670  005237  001410          INC     LPFLG
7537  036674  032777  001000  142236  BIT     #SW9,@SWR       ;LOOP ON ERROR?
7538  036702  001312                  BNE     8$              ;YES, RECONDITION DRIVE
7539  036704  000137  035242          JMP     2$              ;RETURN TO MAINLINE
7540  036710          12$:
7541  036710  005237  001410          INC     LPFLG
7542  036714  032777  001000  142216  BIT     #SW9,@SWR       ;LOOP ON ERROR?
7543  036722  001302                  BNE     8$              ;YES, RECONDITION DRIVE
7544  036724  000137  035536          JMP     3$              ;RETURN TO MAINLINE
7545  036730          14$:
7546  036730  005237  001410          INC     LPFLG
7547  036734  032777  001000  142176  BIT     #SW9,@SWR       ;LOOP ON ERROR?
7548  036742  001272                  BNE     8$              ;YES, RECONDITION DRIVE
7549  036744  000137  036034          JMP     4$              ;RETURN TO MAINLINE
7550  036750          16$:
7551  036750  005237  001410          INC     LPFLG
7552  036754  032777  001000  142156  BIT     #SW9,@SWR       ;LOOP ON ERROR?
7553  036762  001262                  BNE     8$              ;YES, RECONDITION DRIVE
7554  036764  000137  036330          JMP     5$              ;RETURN TO MAINLINE
7555
7556
7557
7558                                  ;;*****************************************************************
7559                                  ;*TEST 40       SEEK TO ALL CYLS FROM 0 & READ HEADERS
7560                                  ;;*****************************************************************
7561  036770  000004          TST40:  SCOPE
7562  036772  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
7563  037000  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
7564
7565  037004  005737  001342          TST     MODTST          ;SEE IF MODULE TESTING
7566  037010  001402                  BEQ     DOSEEK          ;BR IF NO
7567  037012  000137  042744          JMP     CYLINV          ;ELSE BYPASS TESTS 40 & 41
```

```
7568  037016                                DOSEEK:
7569
7570  037016  012737  000000  001350              MOV     #0,FRCYL        ;SETUP FROM CYL
7571  037024  012737  000001  001352              MOV     #1,TOCYL        ;SETUP TO CYL
7572
7573  037032                                1$:
7574  037032  104415                              SCOP1
7575  037034  012706  001100                      MOV     #STACK,SP       ;RESTORE STK PTR
7576
7577  037040  004737  046106                      JSR     PC,SUBCLR
7578  037044  104024                              ERROR   24              ;CERR AFTER SCLR
7579
7580  037046  012737  040626  001176              MOV     #10$,$ESCAPE
7581  037054  013737  001350  003376              MOV     FRCYL,TEMP3     ;SETUP
7582  037062  013737  001352  003400              MOV     TOCYL,TEMP4     ;CYL DIFF
7583  037070  163737  003376  003400              SUB     TEMP3,TEMP4     ;FOR
7584  037076  013737  003400  001360              MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
7585
7586  037104  013765  001352  000020              MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
7587
7588  037112  012765  000017  000000              MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7589  037120  013737  001414  003372              MOV     T10,TEMP1       ;SETUP  TIMEOUT
7590  037126  004737  044176                      JSR     PC,FRDY         ;FIND RDY
7591  037132  104131                              ERROR   131             ;NO RDY AFTER SEEK CMD
7592  037134  012737  030140  003424              MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7593  037142  005037  003426                      CLR     E.B0
7594  037146  012737  003720  003430              MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7595  037154  012737  000001  003432              MOV     #1,E.B1
7596
7597  037162  004737  044720                      JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7598  037166  000003                              .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7599  037170  104203                              ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7600  037172  104204                              ERROR   204             ;MSG B0 ERROR
7601  037174  104205                              ERROR   205             ;MSG A1 ERROR
7602  037176  104206                              ERROR   206             ;MSG B1 ERROR
7603
7604  037200  012737  040646  001176  2$:         MOV     #12$,$ESCAPE
7605  037206  013737  001424  003372              MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7606
7607  037214  004737  044606                      JSR     PC,FATT2        ;FIND ATTN
7608  037220  104132                              ERROR   132             ;NO ATTN AFTER SEEK CMD
7609  037222  032737  100000  003334              BIT     #CERR,HCS1
7610  037230  001401                              BEQ     64$
7611  037232  104210                              ERROR   210             ;CERR AFTER SEEK CMD
7612  037234                                64$:
7613
7614  037234  012737  050340  003424              MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7615  037242  005037  003426                      CLR     E.B0            ;EXPECTED MSG B0
7616  037246  012737  001720  003430              MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7617  037254  012737  000001  003432              MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7618  037262  005037  003434                      CLR     E.A2            ;EXPECTED MSG A2
7619  037266  012737  000002  003436              MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7620  037274  012737  000003  003442              MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7621
7622  037302  004737  044720                      JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7623  037306  000003                              .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
```

```
7624  037310  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7625  037312  104134                        ERROR   134             ;MSG B0 ERROR
7626  037314  104135                        ERROR   135             ;MSG A1 ERROR
7627  037316  104136                        ERROR   136             ;MSG B1 ERROR
7628  037320  005737  001362                TST     CYLDIF
7629  037324  001401                        BEQ     65$
7630  037326  104137                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7631
7632  037330                          65$:
7633
7634  037330  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
7635  037336  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
7636  037344  012765  000005  000000        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7637  037352  013737  001414  003372        MOV     T10,TEMP1
7638  037360  004737  044176                JSR     PC,FRDY         ;FIND RDY
7639  037364  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7640  037366  004737  044460                JSR     PC,TSTATN       ;TEST FOR ATTN
7641  037372  000401                        BR      66$
7642  037374  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7643  037376                          66$:
7644
7645  037376  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7646  037404  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
7647  037410  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7648  037416  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7649  037424  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
7650  037430  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7651  037436  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7652
7653  037444  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7654  037450  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7655  037452  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7656  037454  104265                        ERROR   265             ;MSG B0 ERROR
7657  037456  104274                        ERROR   274             ;MSG A1 ERROR
7658  037460  104266                        ERROR   266             ;MSG B1 ERROR
7659
7660  037462  023737  001364  001352        CMP     CYLADD,TOCYL
7661  037470  001401                        BEQ     3$
7662  037472  104207                        ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
7663
7664  037474                          3$:
7665  037474  104415                        SCOP1
7666  037476  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
7667
7668  037502  004737  046106                JSR     PC,SUBCLR
7669  037506  104024                        ERROR   24              ;CERR AFTER SCLR
7670
7671  037510  005037  001176                CLR     $ESCAPE
7672  037514  013765  001352  000020        MOV     TOCYL,RKDC(R5)  ;CYL #
7673
7674
7675  037522  012700  001674                MOV     #RHTAB,R0
7676  037526  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7677  037534  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7678  037542  004737  044176                JSR     PC,FRDY         ;FIND RDY
7679  037546  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
```

```
 7680  037550  032737  100000  003334          BIT     #CERR,HCS1
 7681  037556  001405                          BEQ     68$
 7682  037560  104174                          ERROR   174                 ;CERR AFTER READ HEADER CMD
 7683  037562  104401  056606                  TYPE    ,MSG18              ;ABORT BALANCE OF TESTS
 7684  037566  000137  043470                  JMP     $EOP                ;ABORT DRIVE
 7685
 7686  037572  016520  000024          68$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
 7687  037576  016520  000024                  MOV     RKDB(R5),(R0)+      ;2'ND WORD
 7688  037602  016520  000024                  MOV     RKDB(R5),(R0)+      ;3'RD WORD
 7689
 7690
 7691  037606  032765  100000  000010          BIT     #DLT,RKCS2(R5)
 7692  037614  001407                          BEQ     69$
 7693  037616  004737  045534                  JSR     PC,GSTAT
 7694  037622  104173                          ERROR   173                 ;DLT AFTER READ HEADER CMD
 7695  037624  104401  056606                  TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
 7696  037630  000137  043470                  JMP     $EOP                ;ABORT DRIVE
 7697  037634                          69$:
 7698
 7699  037634  023737  001674  001352          CMP     RHTAB,TOCYL         ;CHECK WORD 0 (CYL#) ONLY
 7700  037642  001401                          BEQ     67$                 ;BR IF SAME
 7701  037644  104310                          ERROR   310                 ;READ CYL WORD HEADER ERROR
 7702  037646                          67$:
 7703
 7704
 7705  037646  104415                          SCOP1
 7706  037650  012706  001100                  MOV     #STACK,SP           ;RESTORE STK PTR
 7707
 7708  037654  004737  046106                  JSR     PC,SUBCLR
 7709  037660  104024                          ERROR   24                  ;CERR AFTER SCLR
 7710
 7711  037662  012737  040666  001176          MOV     #14$,$ESCAPE
 7712  037670  013765  001350  000020          MOV     FRCYL,RKDC(R5)      ;RETURN TO CYL #
 7713  037676  013737  001350  001354          MOV     FRCYL,CCYL          ;CURRENT CYL FOR TRUERROR ROUTINE
 7714
 7715  037704  012765  000017  000000          MOV     #SEEK,RKCS1(R5)     ;SEEK CMD
 7716  037712  013737  001414  003372          MOV     T10,TEMP1           ;SETUP  TIMEOUT
 7717  037720  004737  044176                  JSR     PC,FRDY             ;FIND RDY
 7718  037724  104131                          ERROR   131                 ;NO RDY AFTER SEEK CMD
 7719  037726  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
 7720  037734  005037  003426                  CLR     E.B0
 7721  037740  012737  005720  003430          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 7722  037746  012737  000001  003432          MOV     #1,E.B1
 7723
 7724  037754  004737  044720                  JSR     PC,CHKMSG           ;CHECK MSGS A0,B0,A1,B1
 7725  037760  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
 7726  037762  104203                          ERROR   203                 ;MSG A0 ERROR DURING SEEK CMD
 7727  037764  104204                          ERROR   204                 ;MSG B0 ERROR
 7728  037766  104205                          ERROR   205                 ;MSG A1 ERROR
 7729  037770  104206                          ERROR   206                 ;MSG B1 ERROR
 7730
 7731
 7732  037772  012737  040706  001176  4$:     MOV     #16$,$ESCAPE
 7733  040000  013737  001424  003372          MOV     T5000,TEMP1         ;SETUP  TIMEOUT
 7734
 7735  040006  004737  044606                  JSR     PC,FATT2            ;FIND ATTN
```

```
7736  040012  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
7737  040014  032737  100000  003334        BIT     #CERR,HCS1
7738  040022  001401                        BEQ     70$
7739  040024  104210                        ERROR   210             ;CERR AFTER SEEK CMD
7740  040026                          70$:
7741
7742  040026  012737  050340  003424        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7743  040034  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
7744  040040  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7745  040046  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7746  040054  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
7747  040060  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7748  040066  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7749
7750  040074  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7751  040100  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7752  040102  104133                        ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7753  040104  104134                        ERROR   134             ;MSG B0 ERROR
7754  040106  104135                        ERROR   135             ;MSG A1 ERROR
7755  040110  104136                        ERROR   136             ;MSG B1 ERROR
7756  040112  005737  001362                TST     CYLDIF
7757  040116  001401                        BEQ     71$
7758  040120  104137                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7759
7760  040122                          71$:
7761
7762  040122  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
7763  040130  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
7764  040136  012765  000005  000000        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7765  040144  013737  001414  003372        MOV     T10,TEMP1
7766  040152  004737  044176                JSR     PC,FRDY         ;FIND RDY
7767  040156  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7768  040160  004737  044460                JSR     PC,TSTATN       ;TEST FOR ATTN
7769  040164  000401                        BR      72$
7770  040166  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7771  040170                          72$:
7772
7773  040170  012737  010340  003424        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7774  040176  005037  003426                CLR     E.B0            ;EXPECTED MSG B0
7775  040202  012737  001720  003430        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7776  040210  012737  000001  003432        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7777  040216  005037  003434                CLR     E.A2            ;EXPECTED MSG A2
7778  040222  012737  000002  003436        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7779  040230  012737  000003  003442        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7780
7781  040236  004737  044720                JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7782  040242  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7783  040244  104273                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7784  040246  104265                        ERROR   265             ;MSG B0 ERROR
7785  040250  104274                        ERROR   274             ;MSG A1 ERROR
7786  040252  104266                        ERROR   266             ;MSG B1 ERROR
7787
7788  040254  023737  001364  001350        CMP     CYLADD,FRCYL
7789  040262  001401                        BEQ     5$
7790  040264  104243                        ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7791
```

```
 7792  040266                          5$:
 7793  040266  104415                        SCOP1
 7794  040270  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 7795
 7796  040274  004737  046106                JSR     PC,SUBCLR
 7797  040300  104024                        ERROR   24              ;CERR AFTER SCLR
 7798
 7799  040302  005037  001176                CLR     $ESCAPE
 7800  040306  013765  001350  000020        MOV     FRCYL,RKDC(R5)  ;CYL #
 7801
 7802
 7803  040314  012700  001674                MOV     #RHTAB,R0
 7804  040320  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
 7805  040326  013737  001426  003372        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 7806  040334  004737  044176                JSR     PC,FRDY         ;FIND RDY
 7807  040340  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
 7808  040342  032737  100000  003334        BIT     #CERR,HCS1
 7809  040350  001405                        BEQ     74$
 7810  040352  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
 7811  040354  104401  056606                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 7812  040360  000137  043470                JMP     $EOP            ;ABORT DRIVE
 7813
 7814  040364  016520  000024        74$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 7815  040370  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
 7816  040374  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
 7817
 7818
 7819  040400  032765  100000  000010        BIT     #DLT,RKCS2(R5)
 7820  040406  001407                        BEQ     75$
 7821  040410  004737  045534                JSR     PC,GSTAT
 7822  040414  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
 7823  040416  104401  056606                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 7824  040422  000137  043470                JMP     $EOP            ;ABORT DRIVE
 7825  040426                          75$:
 7826
 7827  040426  023737  001674  001350        CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
 7828  040434  001401                        BEQ     73$             ;BR IF SAME
 7829  040436  104311                        ERROR   311             ;READ CYL WORD HEADER ERROR
 7830  040440                          73$:
 7831
 7832
 7833  040440  023727  001352  000632        CMP     TOCYL,#410.     ;ALL CYL DONE?
 7834  040446  001404                        BEQ     6$              ;BR IF YES
 7835  040450  005237  001352                INC     TOCYL           ;ELSE DO ANOTHER
 7836  040454  000137  037032                JMP     1$
 7837  040460                          6$:
 7838  040460  004737  050036                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 7839  040464  000520                        BR      TST41           ;:GO TO NEXT TEST
 7840                                                                ;RETURN HERE IF SW 14 IS SET OR
 7841                                                                ;SW 8 WITH SWR <7:0> APPLY
 7842  040466                          8$:
 7843
 7844  040466  004737  046106                JSR     PC,SUBCLR
 7845  040472  104024                        ERROR   24              ;CERR AFTER SCRL
 7846
 7847  040474  013765  001352  000020  76$:  MOV     TOCYL,RKDC(R5)  ;CYL#
```

```
7848
7849  040502  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7850  040510  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
7851  040516  004737  044176                  JSR     PC,FRDY         ;FIND RDY
7852  040522  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
7853
7854  040524  013737  001424  003372          MOV     T5000,TEMP1
7855  040532  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
7856  040536  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7857  040540  032737  100000  003334          BIT     #CERR,HCS1
7858  040546  001401                          BEQ     78$
7859  040550  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
7860
7861  040552  004737  046106          78$:    JSR     PC,SUBCLR
7862  040556  104024                          ERROR   24              ;CERR AFTER SCLR
7863
7864  040560  023727  001352  000000          CMP     TOCYL,#0        ;ALL CYL DONE?
7865  040566  001403                          BEQ     77$             ;BR IF YES
7866  040570  005337  001352                  DEC     TOCYL           ;ELSE DO ANOTHER
7867  040574  000737                          BR      76$
7868
7869  040576  004737  046106          77$:    JSR     PC,SUBCLR
7870  040602  104024                          ERROR   24              ;CERR AFTER SCLR
7871
7872  040604  005037  001176                  CLR     $ESCAPE
7873  040610  005737  001410                  TST     LPFLG
7874  040614  001402                          BEQ     79$
7875  040616  000177  140266                  JMP     @$LPERR         ;SW 9 WAS SET.
7876  040622  000177  140260          79$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7877
7878  040626                          10$:
7879  040626  005237  001410                  INC     LPFLG
7880  040632  032777  001000  140300          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7881  040640  001312                          BNE     8$              ;YES, RECONDITION DRIVE
7882  040642  000137  037200                  JMP     2$              ;RETURN TO MAINLINE
7883  040646                          12$:
7884  040646  005237  001410                  INC     LPFLG
7885  040652  032777  001000  140260          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7886  040660  001302                          BNE     8$              ;YES, RECONDITION DRIVE
7887  040662  000137  037474                  JMP     3$              ;RETURN TO MAINLINE
7888  040666                          14$:
7889  040666  005237  001410                  INC     LPFLG
7890  040672  032777  001000  140240          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7891  040700  001272                          BNE     8$              ;YES, RECONDITION DRIVE
7892  040702  000137  037772                  JMP     4$              ;RETURN TO MAINLINE
7893  040706                          16$:
7894  040706  005237  001410                  INC     LPFLG
7895  040712  032777  001000  140220          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7896  040720  001262                          BNE     8$              ;YES, RECONDITION DRIVE
7897  040722  000137  040266                  JMP     5$              ;RETURN TO MAINLINE
7898
7899                                  ;;*****************************************************************
7900                                  ;*TEST 41       SEEK TO ALL CYLS FROM CYL 410 & READ HEADERS
7901                                  ;;*****************************************************************
7902  040726  000004                  TST41:  SCOPE
7903  040730  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
7904  040736  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
7905
7906
7907  040742  004737  046106          JSR     PC,SUBCLR
7908  040746  104024                  ERROR   24              ;CERR AFTER SCLR
7909
7910  040750  012765  000632  000020  MOV     #410.,RKDC(R5)  ;QUICK SEEK TO CYL 410
7911
7912  040756  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7913  040764  013737  001414  003372  MOV     T10,TEMP1       ;SETUP TIMEOUT
7914  040772  004737  044176          JSR     PC,FRDY         ;FIND RDY
7915  040776  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD.
7916
7917  041000  013737  001424  003372  MOV     T5000,TEMP1
7918  041006  004737  044606          JSR     PC,FATT2        ;FIND ATTN
7919  041012  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
7920  041014  032737  100000  003334  BIT     #CERR,HCS1
7921  041022  001401                  BEQ     64$
7922  041024  104210                  ERROR   210             ;CERR AFTER SEEK CMD.
7923
7924  041026  004737  046106  64$:    JSR     PC,SUBCLR
7925  041032  104024                  ERROR   24              ;CERR AFTER SCLR
7926
7927
7928  041034  012737  000632  001350  MOV     #410.,FRCYL     ;SETUP FROM CYL
7929  041042  012737  000631  001352  MOV     #409.,TOCYL     ;SETUP TO CYL
7930
7931  041050                  1$:
7932  041050  104415                  SCOP1
7933  041052  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
7934
7935  041056  004737  046106          JSR     PC,SUBCLR
7936  041062  104024                  ERROR   24              ;CERR AFTER SCLR
7937
7938  041064  012737  042644  001176  MOV     #10$,$ESCAPE
7939  041072  013737  001350  003376  MOV     FRCYL,TEMP3     ;SETUP
7940  041100  013737  001352  003400  MOV     TOCYL,TEMP4     ;CYL DIFF
7941  041106  163737  003400  003376  SUB     TEMP4,TEMP3     ;FOR
7942  041114  013737  003376  001360  MOV     TEMP3,CALDIF    ;ERROR PRINTOUT
7943
7944  041122  013765  001352  000020  MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
7945
7946  041130  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7947  041136  013737  001414  003372  MOV     T10,TEMP1       ;SETUP  TIMEOUT
7948  041144  004737  044176          JSR     PC,FRDY         ;FIND RDY
7949  041150  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD
7950  041152  012737  030140  003424  MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7951  041160  005037  003426          CLR     E.B0
7952  041164  012737  005720  003430  MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7953  041172  012737  000001  003432  MOV     #1,E.B1
7954
7955  041200  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7956  041204  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7957  041206  104203                  ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
7958  041210  104204                  ERROR   204             ;MSG B0 ERROR
7959  041212  104205                  ERROR   205             ;MSG A1 ERROR
```

```
7960  041214  104206                          ERROR   206             ;MSG B1 ERROR
7961
7962  041216  012737  042664  001176  2$:     MOV     #12$,$ESCAPE
7963  041224  013737  001424  003372          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7964
7965  041232  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
7966  041236  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7967  041240  032737  100000  003334          BIT     #CERR,HCS1
7968  041246  001401                          BEQ     65$
7969  041250  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7970  041252                          65$:
7971
7972  041252  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
7973  041260  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
7974  041264  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
7975  041272  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7976  041300  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
7977  041304  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7978  041312  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7979
7980  041320  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
7981  041324  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7982  041326  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7983  041330  104134                          ERROR   134             ;MSG B0 ERROR
7984  041332  104135                          ERROR   135             ;MSG A1 ERROR
7985  041334  104136                          ERROR   136             ;MSG B1 ERROR
7986  041336  005737  001362                  TST     CYLDIF
7987  041342  001401                          BEQ     66$
7988  041344  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7989
7990  041346                          66$:
7991
7992  041346  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7993  041354  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7994  041362  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7995  041370  013737  001414  003372          MOV     T10,TEMP1
7996  041376  004737  044176                  JSR     PC,FRDY         ;FIND RDY
7997  041402  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7998  041404  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
7999  041410  000401                          BR      67$
8000  041412  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8001  041414                          67$:
8002
8003  041414  012737  010340  003424          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0       ;EXPECTED MSG A0
8004  041422  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
8005  041426  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1        ;EXPECTED A1
8006  041434  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8007  041442  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
8008  041446  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8009  041454  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8010
8011  041462  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8012  041466  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8013  041470  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8014  041472  104265                          ERROR   265             ;MSG B0 ERROR
8015  041474  104274                          ERROR   274             ;MSG A1 ERROR
```

```
 8016  041476  104266                          ERROR   266             ;MSG B1 ERROR
 8017
 8018  041500  023737  001364  001352          CMP     CYLADD,TOCYL
 8019  041506  001401                          BEQ     3$
 8020  041510  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
 8021
 8022  041512                          3$:
 8023  041512  104415                          SCOP1
 8024  041514  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 8025
 8026  041520  004737  046106                  JSR     PC,SUBCLR
 8027  041524  104024                          ERROR   24              ;CERR AFTER SCLR
 8028
 8029  041526  005037  001176                  CLR     $ESCAPE
 8030  041532  013765  001352  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
 8031
 8032
 8033  041540  012700  001674                  MOV     #RHTAB,R0
 8034  041544  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
 8035  041552  013737  001426  003372          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
 8036  041560  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 8037  041564  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
 8038  041566  032737  100000  003334          BIT     #CERR,HCS1
 8039  041574  001405                          BEQ     69$
 8040  041576  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
 8041  041600  104401  056606                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 8042  041604  000137  043470                  JMP     $EOP            ;ABORT DRIVE
 8043
 8044  041610  016520  000024          69$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 8045  041614  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
 8046  041620  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
 8047
 8048
 8049  041624  032765  100000  000010          BIT     #DLT,RKCS2(R5)
 8050  041632  001407                          BEQ     70$
 8051  041634  004737  045534                  JSR     PC,GSTAT
 8052  041640  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
 8053  041642  104401  056606                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 8054  041646  000137  043470                  JMP     $EOP            ;ABORT DRIVE
 8055  041652                          70$:
 8056
 8057  041652  023737  001674  001352          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
 8058  041660  001401                          BEQ     68$             ;BR IF SAME
 8059  041662  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
 8060  041664                          68$:
 8061
 8062
 8063  041664  104415                          SCOP1
 8064  041666  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 8065
 8066  041672  004737  046106                  JSR     PC,SUBCLR
 8067  041676  104024                          ERROR   24              ;CERR AFTER SCLR
 8068
 8069  041700  012737  042704  001176          MOV     #14$,$ESCAPE
 8070  041706  013765  001350  000020          MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
 8071  041714  013737  001350  001354          MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERROR ROUTINE
```

```
8072
8073  041722  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
8074  041730  013737  001414  003372          MOV     T10,TEMP1       ;SETUP  TIMEOUT
8075  041736  004737  044176                  JSR     PC,FRDY         ;FIND RDY
8076  041742  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
8077  041744  012737  030140  003424          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
8078  041752  005037  003426                  CLR     E.B0
8079  041756  012737  003720  003430          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8080  041764  012737  000001  003432          MOV     #1,E.B1
8081
8082  041772  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8083  041776  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8084  042000  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
8085  042002  104204                          ERROR   204             ;MSG B0 ERROR
8086  042004  104205                          ERROR   205             ;MSG A1 ERROR
8087  042006  104206                          ERROR   206             ;MSG B1 ERROR
8088
8089
8090  042010  012737  042724  001176  4$:     MOV     #16$,$ESCAPE
8091  042016  013737  001424  003372          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
8092
8093  042024  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
8094  042030  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
8095  042032  032737  100000  003334          BIT     #CERR,HCS1
8096  042040  001401                          BEQ     71$
8097  042042  104210                          ERROR   210             ;CERR AFTER SEEK CMD
8098  042044                          71$:
8099
8100  042044  012737  050340  003424          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
8101  042052  005037  003426                  CLR     E.B0            ;EXPECTED MSG B0
8102  042056  012737  001720  003430          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
8103  042064  012737  000001  003432          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
8104  042072  005037  003434                  CLR     E.A2            ;EXPECTED MSG A2
8105  042076  012737  000002  003436          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
8106  042104  012737  000003  003442          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
8107
8108  042112  004737  044720                  JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8109  042116  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8110  042120  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
8111  042122  104134                          ERROR   134             ;MSG B0 ERROR
8112  042124  104135                          ERROR   135             ;MSG A1 ERROR
8113  042126  104136                          ERROR   136             ;MSG B1 ERROR
8114  042130  005737  001362                  TST     CYLDIF
8115  042134  001401                          BEQ     72$
8116  042136  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8117
8118  042140                          72$:
8119
8120  042140  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
8121  042146  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
8122  042154  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8123  042162  013737  001414  003372          MOV     T10,TEMP1
8124  042170  004737  044176                  JSR     PC,FRDY         ;FIND RDY
8125  042174  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8126  042176  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
8127  042202  000401                          BR      73$
```

```
                                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                                73$:
                                        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
                                        CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1       ;EXPECTED A1
                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
                                        CLR     E.A2            ;EXPECTED MSG A2
                                        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
                                        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
                                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
                                        ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
                                        ERROR   265             ;MSG B0 ERROR
                                        ERROR   274             ;MSG A1 ERROR
                                        ERROR   266             ;MSG B1 ERROR

                                        CMP     CYLADD,FRCYL
                                        BEQ     5$
                                        ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC

                                5$:
                                        SCOP1
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                        JSR     PC,SUBCLR
                                        ERROR   24              ;CERR AFTER SCLR

                                        CLR     $ESCAPE
                                        MOV     FRCYL,RKDC(R5)  ;CYL #


                                        MOV     #RHTAB,R0
                                        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
                                        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
                                        JSR     PC,FRDY         ;FIND RDY
                                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
                                        BIT     #CERR,HCS1
                                        BEQ     75$
                                        ERROR   174             ;CERR AFTER READ HEADER CMD
                                        TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
                                        JMP     $EOP            ;ABORT DRIVE

                                75$:    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     76$
                                        JSR     PC,GSTAT
                                        ERROR   173             ;DLT AFTER READ HEADER CMD
                                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                                        JMP     $EOP            ;ABORT DRIVE
                                76$:
```

```
042444  023737  001674  001350          CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
042452  001401                          BEQ     74$             ;BR IF SAME
042454  104311                          ERROR   311             ;READ CYL WORD HEADER ERROR
042456                          74$:

042456  023727  001352  000000          CMP     TOCYL,#0        ;ALL CYL DONE?
042464  001404                          BEQ     6$              ;BR IF YES
042466  005337  001352                  DEC     TOCYL           ;ELSE DO ANOTHER
042472  000137  041050                  JMP     1$
042476                          6$:
042476  004737  050036                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
042502  000520                          BR      TST42           ;;GO TO NEXT TEST
                                                                ;RETURN HERE IF SW 14 IS SET OR
                                                                ;SW 8 WITH SWR <7:0> APPLY
042504                          8$:

042504  004737  046106                  JSR     PC,SUBCLR
042510  104024                          ERROR   24              ;CERR AFTER SCRL

042512  013765  001352  000020  77$:    MOV     TOCYL,RKDC(R5)  ;CYL#

042520  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
042526  013737  001414  003372          MOV     T10,TEMP1       ;SETUP TIMEOUT
042534  004737  044176                  JSR     PC,FRDY         ;FIND RDY
042540  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.

042542  013737  001424  003372          MOV     T5000,TEMP1
042550  004737  044606                  JSR     PC,FATT2        ;FIND ATTN
042554  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
042556  032737  100000  003334          BIT     #CERR,HCS1
042564  001401                          BEQ     79$
042566  104210                          ERROR   210             ;CERR AFTER SEEK CMD.

042570  004737  046106          79$:    JSR     PC,SUBCLR
042574  104024                          ERROR   24              ;CERR AFTER SCLR

042576  023727  001352  000632          CMP     TOCYL,#410.     ;ALL CYL DONE?
042604  001403                          BEQ     78$             ;BR IF YES
042606  005237  001352                  INC     TOCYL           ;ELSE DO ANOTHER
042612  000737                          BR      77$

042614  004737  046106          78$:    JSR     PC,SUBCLR
042620  104024                          ERROR   24              ;CERR AFTER SCLR

042622  005037  001176                  CLR     $ESCAPE
042626  005737  001410                  TST     LPFLG
042632  001402                          BEQ     80$
042634  000177  136250                  JMP     @$LPERR         ;SW 9 WAS SET.
042640  000177  136242          80$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET

042644                          10$:
042644  005237  001410                  INC     LPFLG
042650  032777  001000  136262          BIT     #SW9,@SWR       ;LOOP ON ERROR?
042656  001312                          BNE     8$              ;YES, RECONDITION DRIVE
```

```
 8240  042660  000137  041216                        JMP     2$              ;RETURN TO MAINLINE
 8241  042664                            12$:
 8242  042664  005237  001410                        INC     LPFLG
 8243  042670  032777  001000  136242                BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8244  042676  001302                                BNE     8$              ;YES, RECONDITION DRIVE
 8245  042700  000137  041512                        JMP     3$              ;RETURN TO MAINLINE
 8246  042704                            14$:
 8247  042704  005237  001410                        INC     LPFLG
 8248  042710  032777  001000  136222                BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8249  042716  001272                                BNE     8$              ;YES, RECONDITION DRIVE
 8250  042720  000137  042010                        JMP     4$              ;RETURN TO MAINLINE
 8251  042724                            16$:
 8252  042724  005237  001410                        INC     LPFLG
 8253  042730  032777  001000  136202                BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8254  042736  001262                                BNE     8$              ;YES, RECONDITION DRIVE
 8255  042740  000137  042304                        JMP     5$              ;RETURN TO MAINLINE
 8256
 8257  042744                            CYLINV:
 8258
 8259
 8260                                    ;;*****************************************************************
 8261                                    ;*TEST 42       SEEK TO ALL KEY INVALID CYLS
 8262                                    ;*
 8263                                    ;*      THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
 8264                                    ;*      PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
 8265                                    ;*      CYL IS SPECIFIED IN A SEEK.
 8266                                    ;*
 8267                                    ;*      INVALID CYLS ARE 411 THRU 511 (10)  IE. 633 THRU 777 (8)
 8268                                    ;*
 8269                                    ;*      THIS TEST CHECKS KEY INVALID CYLS 411,412,416,448 & 480
 8270                                    ;*      FOR A FULL LOGIC TEST
 8271                                    ;*
 8272                                    ;;*****************************************************************
 8273  042744  000004                    TST42:  SCOPE
 8274  042746  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
 8275  042754  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
 8276
 8277  042760  004737  046106                    JSR     PC,SUBCLR
 8278  042764  104024                            ERROR   24              ;CERR AFTER SCLR
 8279
 8280  042766  012765  000017  000000            MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
 8281  042774  013737  001414  003372            MOV     T10,TEMP1       ;SETUP TIMEOUT
 8282  043002  004737  044176                    JSR     PC,FRDY         ;FIND RDY
 8283  043006  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD.
 8284
 8285  043010  013737  001424  003372            MOV     T5000,TEMP1
 8286  043016  004737  044606                    JSR     PC,FATT2        ;FIND ATTN
 8287  043022  104132                            ERROR   132             ;NO ATTN AFTER SEEK CMD
 8288  043024  032737  100000  003334            BIT     #CERR,HCS1
 8289  043032  001401                            BEQ     64$
 8290  043034  104210                            ERROR   210             ;CERR AFTER SEEK CMD.
 8291
 8292  043036  004737  046106            64$:    JSR     PC,SUBCLR
 8293  043042  104024                            ERROR   24              ;CERR AFTER SCLR
 8294
 8295  043044  005000                            CLR     R0
```

```
8296  043046  005037  001350          CLR     FRCYL           ;FROM CYL 0
8297
8298  043052                  1$:
8299  043052  104415                  SCOP1
8300  043054  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
8301
8302  043060  004737  046106          JSR     PC,SUBCLR
8303  043064  104024                  ERROR   24              ;CERR AFTER SCLR
8304
8305  043066  016037  003304  001352  MOV     INVCYL(R0),TOCYL        ;GET INVALID CYL ADDR
8306  043074  013737  001352  001360  MOV     TOCYL,CALDIF
8307  043102  013765  001352  000020  MOV     TOCYL,RKDC(R5)
8308  043110  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD
8309  043116  012737  000005  003372  MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
8310  043124  004737  044176          JSR     PC,FRDY         ;FIND RDY
8311  043130  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD
8312  043132  004737  044460          JSR     PC,TSTATN
8313  043136  104245                  ERROR   245             ;NO ATTN AFTER SEEK TO INV CYL
8314
8315  043140  032737  000040  003364  BIT     #D.IDAE,HMR3
8316  043146  001001                  BNE     2$
8317  043150  104246                  ERROR   246             ;IDAE NOT SET AFTER SEEK TO INVALID ADDR
8318  043152  032737  000200  003364  2$:     BIT     #D.FLT,HMR3
8319  043160  001001                  BNE     4$
8320  043162  104247                  ERROR   247             ;FLT NOT SET AFTER SEEK TO INV ADDR
8321  043164  032737  020000  003362  4$:     BIT     #D.PIP,HMR2
8322  043172  001401                  BEQ     5$
8323  043174  104250                  ERROR   250             ;PIP SET AFTER SEEK TO INV ADDR
8324  043176  032737  040000  003362  5$:     BIT     #D.DSC,HMR2
8325  043204  001001                  BNE     6$
8326  043206  104251                  ERROR   251             ;DSC NOT SET AFTER SEEK TO INV ADDR
8327
8328  043210  005237  001462          6$:     INC     BYPCERR         ;BYPASS CHECKING FOR CERR IN GSTAT1
8329  043214  012737  050340  003424  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
8330  043222  012737  002240  003426  MOV     #<D.SKI!D.FLT!D.IDAE>,E.B0
8331  043230  012737  001720  003430  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
8332  043236  012737  000001  003432  MOV     #1,E.B1
8333
8334  043244  004737  044720          JSR     PC,CHKMSG       ;CHECK MSGS A0,B0,A1,B1
8335  043250  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
8336  043252  104252                  ERROR   252             ;MSG A0 ERROR AFTER SEEK TO INV CYL
8337  043254  104253                  ERROR   253             ;MSG B0 ERROR
8338  043256  104254                  ERROR   254             ;MSG A1 ERROR
8339  043260  104255                  ERROR   255             ;MSG B1 ERROR
8340  043262  023737  001362  001352  CMP     CYLDIF,TOCYL
8341  043270  001401                  BEQ     7$
8342  043272  104256                  ERROR   256             ;CYL DIFF IN RKMR2 NOT=CYL DIF
8343  043274  023737  001364  001352  7$:     CMP     CYLADD,TOCYL
8344  043302  001401                  BEQ     8$
8345  043304  104257                  ERROR   257             ;CYL ADDR IN RKMR3 NOT=RKDC
8346
8347  043306                  8$:
8348
8349  043306  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
8350  043314  013765  001222  000010  MOV     $UNIT,RKCS2(R5)
8351  043322  012765  000013  000000  MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
```

```
8352  043330  013737  001414  003372          MOV     T10,TEMP1
8353  043336  004737  044176                  JSR     PC,FRDY         ;FIND RDY
8354  043342  104124                          ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
8355
8356  043344  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
8357  043352  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
8358  043360  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8359  043366  013737  001414  003372          MOV     T10,TEMP1
8360  043374  004737  044176                  JSR     PC,FRDY         ;FIND RDY
8361  043400  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8362  043402  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
8363  043406  000401                          BR      66$
8364  043410  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8365  043412                          66$:
8366
8367
8368  043412  004737  045534                  JSR     PC,GSTAT
8369  043416  032737  000040  003364          BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
8370  043424  001401                          BEQ     65$             ;BR IF YES
8371  043426  104155                          ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
8372
8373  043430  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5)
8374  043436  013737  001412  003374          MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
8375  043444  004737  044512                  JSR     PC,FATT1
8376  043450  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
8377
8378
8379  043452  062700  000002                  ADD     #2,R0
8380  043456  020027  000012                  CMP     R0,#10.
8381  043462  001402                          BEQ     $EOP
8382  043464  000137  043052                  JMP     1$
8383
8384
```

```
                        .SBTTL  END OF PASS ROUTINE

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

043470                  $EOP:

043470  000004                  SCOPE
043472  012706  001100          MOV     #STACK,SP
043476  005237  001220          INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
043502  023737  003454  001220  CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
043510  001403                  BEQ     $EOP1+2         ;BR IF YES
043512  000137  011356          JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
043516  000004          $EOP1:  SCOPE
043520  005037  001102          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
043524  005037  001174          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
043530  005237  001216          INC     $PASS           ;;INCREMENT THE PASS NUMBER
043534  042737  100000  001216  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
043542  005327                  DEC     (PC)+           ;;LOOP?
043544  000001          $EOPCT: .WORD   1
043546  003022                  BGT     $DOAGN          ;;YES
043550  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
043552  000001          $ENDCT: .WORD   1
043554  043544                  $EOPCT
043556  104401  043623          TYPE    $ENDMG          ;;TYPE "END PASS #"
043562  013746  001216          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
043566  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
043570  104401  043620          TYPE    $ENULL          ;;TYPE A NULL CHARACTER
043574  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
043600  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
043602  000005                  RESET                   ;;CLEAR THE WORLD
043604  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
043606  000240                  NOP                     ;;SAVE ROOM
043610  000240                  NOP                     ;;FOR
043612  000240                  NOP                     ;;ACT11
043614                  $DOAGN:
043614  000137                  JMP     @(PC)+          ;;RETURN
043616  007740          $RTNAD: .WORD   ST5
043620     377     377  000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
043623     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
043630  050040  051501  020123
043636  000043
```

```
                                .SBTTL  SUBROUTINES

                                ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                ;
043640  012700  003444          CLRFLG: MOV     #DDUMP,R0
043644  012701  177757                  MOV     #-17.,R1
043650  005020                  1$:     CLR     (R0)+
043652  005201                          INC     R1
043654  001375                          BNE     1$
043656  000207                          RTS     PC

                                ;
                                ;TYPE PROGRAM ID IF FTITLE=0
                                ;
043660  005737  001344          TITLE:  TST     FTITLE
043664  001024                          BNE     1$
043666  005237  001344                  INC     FTITLE
043672  104401  055116                  TYPE    ,MSG1                   ;PROGRAM ID
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
043676  005737  000042                  TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
043702  001012                          BNE     64$                     ;;BRANCH IF YES
043704  123727  001230  000001          CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
043712  001406                          BEQ     64$                     ;;BRANCH IF YES
043714  023727  001140  000176          CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
043722  001005                          BNE     65$                     ;;BRANCH IF NO
043724  104406                          GTSWR                           ;;GET SOFT-SWR SETTINGS
043726  000403                          BR      65$
043730  112737  000001  001134  64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
043736                          65$:
043736  000207                  1$:     RTS     PC

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

043740  104411                  GDRVS:  RDLIN
043742  012600                          MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
043744  012701  177770                  MOV     #-8.,R1                 ;SET UP COUNT
043750  112002                  1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
043752  042702  177400                  BIC     #177400,R2              ;MASK HI BYTE
043756  012703  003456                  MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
043762  012704  000060                  MOV     #60,R4

043766  020402                  2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
043770  001415                          BEQ     3$                      ;BRANCH IF YES
043772  005723                          TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
043774  005204                          INC     R4
043776  020427  000070                  CMP     R4,#70
044002  001371                          BNE     2$                      ;S/B 0-7 OR TERMINATOR
044004  005702                          TST     R2
044006  001022                          BNE     4$
044010  020127  177770                  CMP     R1,#-8.
044014  001426                          BEQ     6$                      ;DEFAULT ALL DRIVES
```

```
 8486  044016  005037  003504          7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
 8487  044022  000207                          RTS     PC              ;FOUND TERMINATOR, EXIT
 8488
 8489  044024  005213                  3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
 8490  044026  005237  003454                  INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
 8491  044032  112002                          MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
 8492  044034  042702  177400                  BIC     #177400,R2      ;MASK
 8493  044040  022702  000054                  CMP     #54,R2          ;IS IT A COMMA?
 8494  044044  001407                          BEQ     5$              ;YES, GO TO NEXT WORD.
 8495  044046  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
 8496  044050  001001                          BNE     4$              ;IF NOT, SOMETHING WRONG.
 8497  044052  000761                          BR      7$              ;FOUND TERMINATOR, EXIT
 8498
 8499  044054  104401  057255          4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
 8500  044060  000137  007146                  JMP     PRGSRT          ;START ALL OVER
 8501
 8502  044064  005201                  5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
 8503  044066  001330                          BNE     1$              ;DRIVES TYPED IN.
 8504  044070  000771                          BR      4$              ;IF NORE, HAVE ERROR.
 8505
 8506  044072  005237  003504          6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
 8507  044076  000207                          RTS     PC              ;EXIT.
 8508
 8509                                  ;
 8510                                  ;ROUTINE TO INPUT RKBAS OR DEFAULT.
 8511                                  ;
 8512
 8513  044100  104412                  GBA:    RDOCT
 8514  044102  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 8515  044104  005700                          TST     R0
 8516  044106  001403                          BEQ     1$              ;BRANCH IF DEFAULT.
 8517  044110  010037  001264                  MOV     R0,$BASE
 8518  044114  000207                          RTS     PC
 8519  044116  012737  177440  001264  1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
 8520  044124  000207                          RTS     PC
 8521
 8522                                  ;
 8523                                  ;ROUTINE TO INPUT RKVEC OR DEFAULT
 8524                                  ;
 8525
 8526  044126  104412                  GINT:   RDOCT
 8527  044130  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 8528  044132  005700                          TST     R0
 8529  044134  001405                          BEQ     1$              ;BRANCH IF DEFAULT
 8530  044136  010037  001314                  MOV     R0,RKVEC
 8531  044142  004737  044160          2$:     JSR     PC,SETINT
 8532  044146  000207                          RTS     PC
 8533  044150  012737  000210  001314  1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
 8534  044156  000771                          BR      2$
 8535
 8536                                  ;
 8537                                  ;ROUTINE TO SETUP INTERRORUPT VECTOR & PRIORITY
 8538                                  ;
 8539
 8540  044160  013700  001314          SETINT: MOV     RKVEC,R0
 8541  044164  012720  050556                  MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
8542  044170  013710  001316                  MOV     RKPRI,(R0)      ;PRS TO RKVEC+2
8543  044174  000207                          RTS     PC
8544
8545
8546                                  ;
8547                                  ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
8548                                  ;ENTER WITH A COUNT IN TEMP1
8549                                  ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
8550                                  ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
8551                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8552                                  ;
8553  044176  032765  000200  000000  FRDY:   BIT     #RDY,RKCS1(R5)
8554  044204  001010                          BNE     1$
8555  044206  005337  003372                  DEC     TEMP1
8556  044212  001371                          BNE     FRDY
8557  044214  004737  044332                  JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
8558  044220  004737  045452                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
8559  044224  000207                          RTS     PC              ;NO RDY, EXIT
8560  044226  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
8561  044232  004737  044332                  JSR     PC,HOLD
8562  044236  004737  045452                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
8563  044242  000207                          RTS     PC
8564
8565                                  ;
8566                                  ; ROUTINE TO FIND CONTROLLER READY & STORE DRIVE REGS ONLY
8567                                  ;
8568  044244  032765  000200  000000  FRDY1:  BIT     #RDY,RKCS1(R5)
8569  044252  001014                          BNE     1$
8570  044254  005337  003372                  DEC     TEMP1
8571  044260  001371                          BNE     FRDY1
8572  044262  016537  000034  003362          MOV     RKMR2(R5),HMR2
8573  044270  016537  000036  003364          MOV     RKMR3(R5),HMR3
8574  044276  004737  045452                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
8575  044302  000207                          RTS     PC              ;NO RDY, EXIT
8576  044304  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
8577  044310  016537  000034  003362          MOV     RKMR2(R5),HMR2
8578  044316  016537  000036  003364          MOV     RKMR3(R5),HMR3
8579  044324  004737  045452                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
8580  044330  000207                          RTS     PC
8581                                  ;
8582                                  ;STORE ALL RK611 REGISTERS IN HOLDING REGS
8583                                  ;
8584
8585  044332  016537  000000  003334  HOLD:   MOV     RKCS1(R5),HCS1
8586  044340  016537  000010  003336          MOV     RKCS2(R5),HCS2
8587  044346  016537  000002  003340          MOV     RKWC(R5),HWC
8588  044354  016537  000004  003342          MOV     RKBA(R5),HBA
8589  044362  016537  000006  003344          MOV     RKDA(R5),HDA
8590  044370  016537  000012  003346          MOV     RKDS(R5),HDS
8591  044376  016537  000014  003350          MOV     RKER(R5),HER
8592  044404  016537  000016  003352          MOV     RKASOF(R5),HASOF
8593  044412  016537  000020  003354          MOV     RKDC(R5),HDC
8594  044420  016537  000026  003360          MOV     RKMR1(R5),HMR1
8595  044426  016537  000034  003362          MOV     RKMR2(R5),HMR2
8596  044434  016537  000036  003364          MOV     RKMR3(R5),HMR3
8597  044442  016537  000030  003366          MOV     RKECPS(R5),HPOS
```

```
 8598  044450  016537  000032  003370          MOV     RKECPT(R5),HPAT
 8599  044456  000207                          RTS     PC
 8600
 8601                                  ;
 8602                                  ;ROUTINE TO CHECK FOR CORRECT ATTN
 8603                                  ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
 8604                                  ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
 8605                                  ;
 8606  044460  010446                  ?STATN: MOV     R4,-(SP)        ;SAV R4
 8607  044462  013704  001222                  MOV     $UNIT,R4
 8608  044466  136437  003324  003353          BITB    ATTN(R4),HASOF+1
 8609  044474  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
 8610  044476  012604                          MOV     (SP)+,R4        ;RESTOR R4
 8611  044500  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
 8612  044504  000207                          RTS     PC
 8613  044506  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
 8614  044510  000207                          RTS     PC
 8615
 8616                                  ;
 8617                                  ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
 8618                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
 8619                                  ;RETURN IF NO ATTN (ERROR CONDITION)
 8620                                  ;RETURN +2 IF ATTN FOUND
 8621                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 8622                                  ;
 8623
 8624  044512  010446                  FATT1:  MOV     R4,-(SP)        ;SAV R4
 8625  044514  012737  177777  003372  3$:     MOV     #-1,TEMP1
 8626  044522  013704  001222                  MOV     $UNIT,R4
 8627  044526  136465  003324  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
 8628  044534  001014                          BNE     2$
 8629  044536  005337  003372                  DEC     TEMP1
 8630  044542  001371                          BNE     1$
 8631  044544  005337  003374                  DEC     TEMP2
 8632  044550  001361                          BNE     3$
 8633  044552  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
 8634  044556  004737  045534                  JSR     PC,GSTAT        ;GET LATEST STATUS
 8635  044562  012604                          MOV     (SP)+,R4        ;RESTOR R4
 8636  044564  000207                          RTS     PC
 8637  044566  005065  000026          2$:     CLR     RKMR1(R5)
 8638  044572  004737  045534                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
 8639  044576  012604                          MOV     (SP)+,R4        ;RESTOR R4
 8640  044600  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
 8641  044604  000207                          RTS     PC
 8642
 8643                                  ;
 8644                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
 8645                                  ;ENTER WITH COUNT IN TEMP1
 8646                                  ;RETURN IF NO ATTN (ERROR)
 8647                                  ;RETURN +2 IF ATTN FOUND
 8648                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 8649                                  ;
 8650
 8651  044606  010446                  FATT2:  MOV     R4,-(SP)        ;SAV R4
 8652  044610  013704  001222          2$:     MOV     $UNIT,R4
 8653  044614  136465  003324  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
```

```
 8654  044622  001011                        BNE     1$
 8655  044624  005337  003372                DEC     TEMP1
 8656  044630  001367                        BNE     2$
 8657  044632  005065  000026                CLR     RKMR1(R5)       ;SELECT WORD 0
 8658  044636  004737  045534                JSR     PC,GSTAT        ;GET LATEST STATUS.
 8659  044642  012604                        MOV     (SP)+,R4        ;RESTOR R4
 8660  044644  000207                        RTS     PC
 8661  044646  005065  000026          1$:   CLR     RKMR1(R5)
 8662  044652  004737  045534                JSR     PC,GSTAT
 8663  044656  012604                        MOV     (SP)+,R4        ;RESTOR R4
 8664  044660  062716  000002                ADD     #2,(SP)         ;SKIP OVER ERROR
 8665  044664  000207                        RTS     PC
 8666                                  ;
 8667                                  ;ENTER WITH A COUNT IN TEMP1
 8668                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
 8669                                  ;WHEN COUNT IS 0...BASED ON AN 11/05.
 8670                                  ;
 8671  044666  005737  003372          DLY:  TST     TEMP1           ;5.6 US
 8672  044672  001403                        BEQ     1$              ;2.5 US
 8673  044674  005337  003372                DEC     TEMP1           ;6.8 US
 8674  044700  000772                        BR      DLY             ;2.5 US
 8675  044702  000207                  1$:   RTS     PC              ;3.8 US
 8676                                  ;
 8677                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
 8678                                  ;
 8679
 8680  044704  104401  056516          BYP:  TYPE    ,MSG14          ;BYPASS DRIVE
 8681  044710  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 8682                                                                ;;TYPE DR#
 8683  044712  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
 8684  044714     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
 8685  044715     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 8686  044716  000207                        RTS     PC
 8687                                  ;
 8688                                  ; THIS ROUTINE READS ALL MSG A&B WORDS & CHECKS THEM AS REQ'D
 8689                                  ;
 8690  044720  017637  000000  001466  CHKMSG: MOV    @(SP),CHKFLG    ;PASS MSGS TO BE TESTED
 8691  044726  062716  000002                ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
 8692  044732  004737  045576                JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS
 8693
 8694  044736  053737  001222  003424        BIS     $UNIT,E.A0      ;SET UNIT #
 8695  044744  053737  001222  003430        BIS     $UNIT,E.A1
 8696  044752  053737  001222  003434        BIS     $UNIT,E.A2
 8697  044760  053737  001222  003440        BIS     $UNIT,E.A3
 8698
 8699  044766  013746  003372                MOV     TEMP1,-(SP)     ;SAVE TEMP 1
 8700
 8701  044772  013737  003424  003372        MOV     E.A0,TEMP1
 8702  045000  004737  047732                JSR     PC,SBPAR        ;GET PARITY FOR MSG A0
 8703  045004  013737  003372  003424        MOV     TEMP1,E.A0
 8704
 8705  045012  013737  003430  003372        MOV     E.A1,TEMP1
 8706  045020  004737  047732                JSR     PC,SBPAR        ;GET PARITY FOR MSG A1
 8707  045024  013737  003372  003430        MOV     TEMP1,E.A1
 8708
 8709  045032  013737  003434  003372        MOV     E.A2,TEMP1
```

```
 8710  045040  004737  047732                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
 8711  045044  013737  003372  003434          MOV     TEMP1,E.A2
 8712
 8713  045052  013737  003426  003372          MOV     E.B0,TEMP1
 8714  045060  004737  047732                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
 8715  045064  013737  003372  003426          MOV     TEMP1,E.B0
 8716
 8717  045072  013737  003432  003372          MOV     E.B1,TEMP1
 8718  045100  004737  047732                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
 8719  045104  013737  003372  003432          MOV     TEMP1,E.B1
 8720
 8721  045112  013737  003436  003372          MOV     E.B2,TEMP1
 8722  045120  004737  047732                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
 8723  045124  013737  003372  003436          MOV     TEMP1,E.B2
 8724
 8725  045132  013737  003442  003372          MOV     E.B3,TEMP1
 8726  045140  004737  047732                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
 8727  045144  013737  003372  003442          MOV     TEMP1,E.B3
 8728
 8729  045152  012637  003372                  MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
 8730  045156  013737  001176  001172          MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE
 8731
 8732  045164  023737  003404  003424          CMP     H.A0,E.A0       ;TEST MSG A0
 8733  045172  001411                          BEQ     2$              ;BR IF OK
 8734  045174  012737  045206  001176          MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
 8735  045202  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR
 8736  045204  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
 8737
 8738  045206  032777  001000  133724  1$:     BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
 8739  045214  001107                          BNE     20$             ;& BR IF LOOP ON ERROR
 8740  045216  062716  000002          2$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
 8741
 8742  045222  023737  003406  003426          CMP     H.B0,E.B0       ;TEST MSG B0
 8743  045230  001411                          BEQ     5$              ;BR IF OK
 8744  045232  012737  045244  001176          MOV     #4$,$ESCAPE     ;ELSE SETUP ESCAPE
 8745  045240  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR
 8746  045242  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
 8747
 8748  045244  032777  001000  133666  4$:     BIT     #SW9,@SWR       ;RETURN HERE FROM MAINLINE ERROR
 8749  045252  001070                          BNE     20$             ;& BR IF LOOP ON ERROR
 8750  045254  062716  000002          5$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
 8751
 8752  045260  023737  003410  003430          CMP     H.A1,E.A1       ;TEST MSG A1
 8753  045266  001411                          BEQ     8$              ;BR IF OK
 8754  045270  012737  045302  001176          MOV     #7$,$ESCAPE
 8755  045276  011646                          MOV     (SP),-(SP)
 8756  045300  000207                          RTS     PC
 8757
 8758  045302  032777  001000  133630  7$:     BIT     #SW9,@SWR
 8759  045310  001051                          BNE     20$
 8760  045312  062716  000002          8$:     ADD     #2,(SP)
 8761
 8762  045316  023737  003412  003432          CMP     H.B1,E.B1       ;TEST MSG B1
 8763  045324  001411                          BEQ     11$             ;BR IF OK
 8764  045326  012737  045340  001176          MOV     #10$,$ESCAPE
 8765  045334  011646                          MOV     (SP),-(SP)
```

```
 8766  045336  000207                          RTS     PC
 8767
 8768  045340  032777  001000  133572  10$:    BIT     #SW9,@SWR
 8769  045346  001032                          BNE     20$
 8770  045350  062716  000002          11$:    ADD     #2,(SP)
 8771
 8772  045354  032737  000001  001466  12$:    BIT     #T.A2,CHKFLG    ;TEST MSG A2?
 8773  045362  001402                          BEQ     13$             ;BR IF NO
 8774  045364  004737  046546                  JSR     PC,RCYLD        ;PUT INFO IN CYLDIF, DO NOT CHECK
 8775
 8776  045370  032737  000002  001466  13$:    BIT     #T.B2,CHKFLG    ;TEST MSG B2?
 8777  045376  001402                          BEQ     14$             ;BR IF NO
 8778  045400  004737  046620                  JSR     PC,RCYLA        ;PUT INFO IN CYLADD, DO NOT CHECK
 8779
 8780  045404  032737  000004  001466  14$:    BIT     #T.B3,CHKFLG    ;TEST MSG B3?
 8781  045412  001404                          BEQ     15$
 8782  045414  004737  046656                  JSR     PC,RSEC         ;PUT INFO IN SECTOR, DO NOT CHECK
 8783  045420  004737  046714                  JSR     PC,RHEAD        ;PUT INFO IN HEADA, DO NOT CHECK
 8784
 8785  045424  013737  001172  001176  15$:    MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
 8786  045432  000207                          RTS     PC
 8787
 8788  045434  012706  001100          20$:    MOV     #STACK,SP       ;RESET STACK PTR
 8789  045440  013737  001172  001176          MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
 8790  045446  000177  133436                  JMP     @$LPERR
```

```
 8791
```

```
                        ;
                        ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
```

```
 8794                                   ; IE: IF NED, CTO OR MDS SET, MSG A&B ARE INVALID
 8795                                   ;
 8796  045452  005737  001462           CKCERR: TST     BYPCERR
 8797  045456  001025                           BNE     4$
 8798  045460  032737  100000  003334           BIT     #CERR,HCS1
```

```
 8799  045466  001001                          BNE     1$              ;BR IF CERR
 8800  045470  000207                          RTS     PC
 8801
 8802  045472  032737  004000  003334  1$:     BIT     #CTO,HCS1
 8803  045500  001402                          BEQ     2$              ;BR IF NOT CTO
 8804  045502  104326                          ERROR   326             ;CTO ERROR, MSG A&B INVALID
 8805  045504  000207                          RTS     PC
 8806
 8807  045506  032737  010000  003336  2$:     BIT     #NED,HCS2
```

```
 8808  045514  001401                        BEQ     3$              ;BR IF NOT NED
 8809  045516  104327                        ERROR   327             ;NED ERROR, MSG A&B INVALID
 8810
 8811  045520  032737  001000  003336  3$:   BIT     #MDS,HCS2
 8812  045526  001401                        BEQ     4$
 8813  045530  104330                        ERROR   330             ;MDS ERROR, MSG A&B INVALID
 8814
 8815  045532  000207                  4$:   RTS     PC
 8816
 8817                                  ;
 8818                                  ;THIS ROUTINE DOES THE SELECT DRIVE CMD TO GET STATUS
 8819                                  ;IT THEN WAITS FOR CONTROLLER  READY.
 8820                                  ;IF RDY NOT RECEIVED BY A TIMEOUT, AN ERROR IS FLAGGED
 8821                                  ;
 8822  045534  013746  003372          GSTAT:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
 8823  045540  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 8824  045546  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;GET STATUS WITH SELECT DRIVE CMD
 8825  045554  013737  001414  003372          MOV     TIO,TEMP1
 8826  045562  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 8827  045566  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRIVE CMD
 8828  045570  012637  003372                  MOV     (SP)+,TEMP1     ;RESTOR TEMP1
 8829  045574  000207                          RTS     PC
 8830
```

```
                        ;
                        ; THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
                        ; & ALL CONTROLLER REGISTERS
```

```
8834                                    ;
8835  045576  013746  003372            GSTAT1: MOV     TEMP1,-(SP)     ;SAVE TEMP 1
8836  045602  004737  044332                    JSR     PC,HOLD         ;GET ALL CONTR REGS
8837  045606  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
8838  045614  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8839  045622  012765  000003  000026            MOV     #3,RKMR1(R5)    ;SELECT WORD 3
8840  045630  012765  000001  000000            MOV     #SELDRV,RKCS1(R5)       ;GET STATUS
8841  045636  013737  001414  003372            MOV     T10,TEMP1
8842  045644  004737  044244                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
8843  045650  104117                            ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
8844  045652  013737  003362  003420            MOV     HMR2,H.A3       ;STORE MSG A3
8845  045660  013737  003364  003422            MOV     HMR3,H.B3       ;STORE MSG B3
8846
8847  045666  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
8848  045674  013765  001222  000010            MOV     $UNIT,RKCS2(R5)
8849  045702  012765  000002  000026            MOV     #2,RKMR1(R5)    ;SELECT WORD 2
8850  045710  012765  000001  000000            MOV     #SELDRV,RKCS1(R5)
8851  045716  013737  001414  003372            MOV     T10,TEMP1
8852  045724  004737  044244                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
8853  045730  104117                            ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
8854  045732  013737  003362  003414            MOV     HMR2,H.A2       ;STORE MSG A2
8855  045740  013737  003364  003416            MOV     HMR3,H.B2       ;STORE MSG B2
8856
8857  045746  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
8858  045754  013765  001222  000010            MOV     $UNIT,RKCS2(R5)
8859  045762  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
8860  045770  012765  000001  000000            MOV     #SELDRV,RKCS1(R5)
8861  045776  013737  001414  003372            MOV     T10,TEMP1
8862  046004  004737  044244                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
8863  046010  104117                            ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
8864  046012  013737  003362  003410            MOV     HMR2,H.A1       ;STORE MSG A1
8865  046020  013737  003364  003412            MOV     HMR3,H.B1       ;STORE MSG B1
8866
8867  046026  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
8868  046034  013765  001222  000010            MOV     $UNIT,RKCS2(R5)
8869  046042  012765  000001  000000            MOV     #SELDRV,RKCS1(R5)       ;SELECT WORD 0
8870  046050  013737  001414  003372            MOV     T10,TEMP1
8871  046056  004737  044244                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
8872  046062  104117                            ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
8873  046064  013737  003362  003404            MOV     HMR2,H.A0       ;STORE MSG A0
8874  046072  013737  003364  003406            MOV     HMR3,H.B0       ;STORE MSG B0
8875
8876  046100  012637  003372                    MOV     (SP)+,TEMP1     ;RESTORE TEMP1
8877  046104  000207                            RTS     PC
8878
8879                                    ;
8880                                    ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
8881                                    ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
8882                                    ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
8883                                    ;RETURN IF CERR SET
8884                                    ;RETURN +2 IF CERR CLEAR
8885                                    ;
8886  046106  012765  000040  000010    SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
8887  046114  013737  001414  003372            MOV     T10,TEMP1
8888  046122  004737  044176                    JSR     PC,FRDY         ;FIND RDY
8889  046126  104120                            ERROR   120             ;RDY NOT SET BY END OF SCLR
```

```
8890  046130  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8891  046136  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
8892  046142  004737  045534                  JSR     PC,GSTAT        ;GET STATUS
8893  046146  032737  100000  003334          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
8894  046154  001401                          BEQ     1$
8895  046156  000207                          RTS     PC
8896  046160  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
8897  046164  000207                          RTS     PC
8898
8899                                          ;
8900                                          ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
8901                                          ;
8902  046166  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
8903  046174  004737  045534                  JSR     PC,GSTAT
8904  046200  013737  003364  001406          MOV     HMR3,SECTOR
8905  046206  042737  177017  001406          BIC     #^C<M.SECT>,SECTOR
8906  046214  006237  001406                  ASR     SECTOR          ;RIGHT JUSTIFY
8907  046220  006237  001406                  ASR     SECTOR          ;SECTOR
8908  046224  006237  001406                  ASR     SECTOR          ;INFO
8909  046230  006237  001406                  ASR     SECTOR
8910  046234  000207                          RTS     PC
8911
8912
8913                                          ;
8914                                          ;FIND SECTOR 0 IN 22 SECTOR FORMAT.
8915                                          ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
8916                                          ;
8917  046236  013746  003372          FS022:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
8918  046242  013737  001424  003372          MOV     T5000,TEMP1     ;SETUP TIMEOUT
8919  046250  004737  046166          1$:     JSR     PC,RDSEC                ;READ SECTOR
8920  046254  005737  001406                  TST     SECTOR          ;LOOK FOR SECTOR 0
8921  046260  001005                          BNE     2$
8922  046262  004737  046166                  JSR     PC,RDSEC
8923  046266  005737  001406                  TST     SECTOR
8924  046272  001406                          BEQ     3$              ;BR IF SAME TWICE
8925  046274  005337  003372          2$:     DEC     TEMP1
8926  046300  001363                          BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
8927  046302  012637  003372                  MOV     (SP)+,TEMP1     ;ELSE RESTORE TEMP1
8928  046306  000207                          RTS     PC              ;EXIT
8929  046310  012637  003372          3$:     MOV     (SP)+,TEMP1
8930  046314  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
8931  046320  000207                          RTS     PC
8932
8933                                          ;
8934                                          ;FIND NEXT SECTOR IN 22 SECTOR FORMAT
8935                                          ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
8936                                          ;
8937  046322  013746  003372          FNS22:  MOV     TEMP1,-(SP)     ;SAVE TEMP 1
8938  046326  013737  001420  003372          MOV     T500,TEMP1      ;SETUP TIMEOUT
8939  046334  004737  046166          1$:     JSR     PC,RDSEC                ;READ SECTOR
8940  046340  023737  001402  001406          CMP     PSEC,SECTOR
8941  046346  001406                          BEQ     3$              ;BR IF SAME
8942  046350  004737  046166                  JSR     PC,RDSEC                ;ELSE TRY READ DIFFERENT TWICE
8943  046354  023737  001402  001406          CMP     PSEC,SECTOR
8944  046362  001006                          BNE     2$              ;BR IF DIFFERENT TWICE
8945  046364  005337  003372          3$:     DEC     TEMP1           ;ELSE TRY AGAIN IF TIME LEFT
```

```
 8946  046370  001361                        BNE     1$
 8947  046372  012637  003372                MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
 8948  046376  000207                        RTS     PC
 8949  046400  012637  003372        2$:     MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
 8950  046404  062716  000002                ADD     #2,(SP)         ;SKIP OVER ERROR
 8951  046410  000207                        RTS     PC
 8952                                  ;
 8953                                  ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
 8954                                  ;
 8955  046412  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
 8956  046420  004737  045534                JSR     PC,GSTAT
 8957  046424  013737  003362  001362        MOV     HMR2,CYLDIF
 8958  046432  042737  160017  001362        BIC     #^C<M.CDIF>,CYLDIF
 8959  046440  006237  001362                ASR     CYLDIF          ;RIGHT JUSTIFY
 8960  046444  006237  001362                ASR     CYLDIF          ;CYL DIFF/OFFSET
 8961  046450  006237  001362                ASR     CYLDIF          ;INFO
 8962  046454  006237  001362                ASR     CYLDIF
 8963  046460  023727  001362  000777        CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
 8964  046466  001002                        BNE     1$              ;BR IF NOT
 8965  046470  005037  001362                CLR     CYLDIF          ;CLR IF YES
```

```
8966  046474  000207                    1$:     RTS     PC
8967                                    ;
8968                                    ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
8969                                    ;
8970  046476  012765  000002  000026    RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
8971  046504  004737  045534            JSR     PC,GSTAT
8972  046510  013737  003364  001364    MOV     HMR3,CYLADD
8973  046516  042737  160017  001364    BIC     #↑C<M.CADD>,CYLADD
8974  046524  006237  001364            ASR     CYLADD          ;RIGHT JUSTIFY
8975  046530  006237  001364            ASR     CYLADD          ;CYL ADDR
8976  046534  006237  001364            ASR     CYLADD          ;INFO
8977  046540  006237  001364            ASR     CYLADD
8978  046544  000207                    RTS     PC
8979                                    ;
8980                                    ; READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
8981                                    ;
8982  046546  013737  003414  001362    RCYLD:  MOV     H.A2,CYLDIF
8983  046554  042737  160017  001362    BIC     #↑C<M.CDIF>,CYLDIF      ;CLEAR UNWANTED INFO
8984  046562  006237  001362            ASR     CYLDIF          ;RIGHT JUSTIFY
8985  046566  006237  001362            ASR     CYLDIF
8986  046572  006237  001362            ASR     CYLDIF
8987  046576  006237  001362            ASR     CYLDIF
8988  046602  023727  001362  000777    CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
8989  046610  001002                    BNE     1$              ;BR IF NO
8990  046612  005037  001362            CLR     CYLDIF          ;ELSE CLEAR
8991  046616  000207                    1$:     RTS     PC
8992                                    ;
8993                                    ; READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
8994                                    ;
8995  046620  013737  003416  001364    RCYLA:  MOV     H.B2,CYLADD
8996  046626  042737  160017  001364    BIC     #↑C<M.CADD>,CYLADD      ;CLEAR UNWANTED INFO
8997  046634  006237  001364            ASR     CYLADD          ;RIGHT JUSTIFY
8998  046640  006237  001364            ASR     CYLADD
8999  046644  006237  001364            ASR     CYLADD
9000  046650  006237  001364            ASR     CYLADD
9001  046654  000207                    RTS     PC
9002                                    ;
9003                                    ; READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
9004                                    ;
9005  046656  013737  003422  001406    RSEC:   MOV     H.B3,SECTOR
9006  046664  042737  177017  001406    BIC     #↑C<M.SECT>,SECTOR      ;CLEAR UNWANTED INFO
9007  046672  006237  001406            ASR     SECTOR          ;RIGHT JUSTIFY
9008  046676  006237  001406            ASR     SECTOR
9009  046702  006237  001406            ASR     SECTOR
9010  046706  006237  001406            ASR     SECTOR
9011  046712  000207                    RTS     PC
9012                                    ;
9013                                    ; READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
9014                                    ;
9015  046714  013737  003422  001432    RHEAD:  MOV     H.B3,HEADA
9016  046722  042737  170777  001432    BIC     #↑C<M.HEAD>,HEADA       ;CLEAR UNWANTED INFO
9017  046730  006237  001432            ASR     HEADA           ;RIGHT JUSTIFY IT
9018  046734  000337  001432            SWAB    HEADA
9019  046740  000207                    RTS     PC
9020                                    ;
9021                                    ;FIND LIMIT DETECT ON SEEK IN RKMR3 BEFORE TIMEOUT
```

```
                                        ;RETURN IF NOT FOUND: ERROR
                                        ;RETURN+2 IF FOUND: SKIP OVER ERROR
                                        ;
046742  005037  001460          FLIM:   CLR     LIMERROR                ;LIMIT DETECT ERROR FLAG
046746  012737  000764  003372          MOV     #500.,TEMP1     ;SETUP TIMEOUT
046754  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
046762  004737  045534          1$:     JSR     PC,GSTAT
046766  032737  020000  003364          BIT     #D.LIMD,HMR3
046774  001006                          BNE     2$                      ;EXIT IF SET
046776  005337  003372                  DEC     TEMP1
047002  001367                          BNE     1$
047004  005237  001460                  INC     LIMERROR                ;SET LIMIT DETECT FLAG
047010  000207                          RTS     PC
047012  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
047016  000207                          RTS     PC
                                        ;
                                        ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE TIMEOUT
                                        ;ENTER WITH TIME IN SECONDS IN TEMP2
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND - SKIP OVER ERROR
                                        ;
047020  012737  177777  003372  FHDHM:  MOV     #-1,TEMP1       ;ALL 1'S
047026  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
047034  004737  045534          1$:     JSR     PC,GSTAT
047040  032737  000040  003362          BIT     #D.HDHM,HMR2
047046  001007                          BNE     2$
047050  005337  003372                  DEC     TEMP1
047054  001367                          BNE     1$
047056  005337  003374                  DEC     TEMP2
047062  001356                          BNE     FHDHM
047064  000207                          RTS     PC
047066  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
047072  000207                          RTS     PC
                                        ;
                                        ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE TIMEOUT
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND: SKIP OVER ERROR
                                        ;
047074  012737  177777  003372  FLOAD:  MOV     #-1,TEMP1       ;SETUP TIMEOUT
047102  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
047110  004737  045534          1$:     JSR     PC,GSTAT
047114  032737  010000  003362          BIT     #D.LOAD,HMR2
047122  001004                          BNE     2$
047124  005337  003372                  DEC     TEMP1
047130  001367                          BNE     1$
047132  000207                          RTS     PC
047134  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
047140  000207                          RTS     PC
                                        ;
                                        ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
                                        ;ENTER WITH CYL # IN 'CALADD'
                                        ;ENTER WITH HEAD # IN 'HEAD'
                                        ;ENTER WITH FORMAT IN 'FORMAT'
                                        ;
047142  010046                  FHDTAB: MOV     R0,-(SP)        ;SAV R0
047144  010146                          MOV     R1,-(SP)        ;SAV R1
```

```
 9078  047146  012700  001470            MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
 9079  047152  005001                    CLR     R1              ;SECTOR COUNTER
 9080  047154  013737  001430  001434    MOV     HEAD,HD1
 9081  047162  006337  001434            ASL     HD1
 9082  047166  006337  001434            ASL     HD1
 9083  047172  006337  001434            ASL     HD1
 9084  047176  006337  001434            ASL     HD1
 9085  047202  006337  001434            ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
 9086  047206  013737  001436  001440    MOV     FORMAT,FMT1
 9087  047214  000337  001440            SWAB    FMT1
 9088  047220  006337  001440            ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
 9089
 9090  047224  013720  001366    1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
 9091  047230  010110                    MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
 9092  047232  053710  001434            BIS     HD1,(R0)        ;               -HEAD NO
 9093  047236  053710  001440            BIS     FMT1,(R0)       ;               -FORMAT
 9094  047242  005737  001464            TST     BYPFMT
 9095  047246  001403                    BEQ     2$              ;BR IF TRUE FORMAT
 9096  047250  052710  140000            BIS     #<BIT14!BIT15>,(R0)     ;SET GOOD SECTOR FLAGS
 9097  047254  000402                    BR      3$
 9098  047256  004737  047336    2$:     JSR     PC,SECFLG       ;GET  SECTOR FLAGS
 9099
 9100  047262  013737  001366  003372  3$:     MOV     CALADD,TEMP1
 9101  047270  011037  003374            MOV     (R0),TEMP2
 9102  047274  043737  001366  003374    BIC     CALADD,TEMP2
 9103  047302  042037  003372            BIC     (R0)+,TEMP1
 9104  047306  053737  003372  003374    BIS     TEMP1,TEMP2
 9105  047314  013720  003374            MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
 9106
 9107  047320  005201                    INC     R1              ;SECTOR CTR
 9108  047322  020127  000026            CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
 9109  047326  001336                    BNE     1$              ;BR IF NO
 9110
 9111  047330  012601                    MOV     (SP)+,R1        ;RESTOR R1
 9112  047332  012600                    MOV     (SP)+,R0        ;RESTOR R0
 9113  047334  000207                    RTS     PC
 9114
 9115                            ;
 9116                            ; THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS TEST
 9117                            ; & SETS BITS 14 & 15 APPROPRIATELY.
 9118                            ;
 9119  047336  010246            SECFLG: MOV     R2,-(SP)        ;SAVE R2
 9120  047340  005737  001436            TST     FORMAT
 9121  047344  001016                    BNE     1$              ;BR IF 20 SECTOR FORMAT
 9122  047346  012702  002314            MOV     #BSE22H+8.,R2
 9123  047352  004737  047406            JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
 9124  047356  052710  100000            BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
 9125
 9126  047362  012702  054762            MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
 9127  047366  004737  047406            JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
 9128  047372  052710  040000            BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
 9129
 9130  047376  012602                    MOV     (SP)+,R2        ;ELSE RETURN HERE
 9131  047400  000207                    RTS     PC
 9132
 9133
```

```
 9134  047402  012602                1$:     MOV     (SP)+,R2        ;RESTORE R2
 9135  047404  000207                        RTS     PC
 9136
 9137                                ;
 9138                                ; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
 9139                                ; ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC) IN TEMP1
 9140                                ; RETURN IF NO COMPARE
 9141                                ; RETURN +4 IF COMPARE
 9142                                ;
 9143  047406  010346                FLGTST: MOV     R3,-(SP)        ;SAVE R3
 9144
 9145  047410  021227  177777        1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
 9146  047414  001002                        BNE     2$              ;BR IF NO
 9147  047416  012603                        MOV     (SP)+,R3        ;RESTORE R3
 9148  047420  000207                        RTS     PC
 9149
 9150  047422  022237  001366        2$:     CMP     (R2)+,CALADD    ;SEE IF = CYL #, & ADV PTR TO TRACK/SECTOR WORD
 9151  047426  001403                        BEQ     3$
 9152  047430  062702  000002                ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
 9153  047434  000765                        BR      1$
 9154
 9155  047436  013703  001430        3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
 9156  047442  000303                        SWAB    R3
 9157  047444  050103                        BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
 9158  047446  022203                        CMP     (R2)+,R3        ;SECTOR/HEAD COMPARE? & INCR TO NEXT CYL WORD
 9159  047450  001401                        BEQ     4$              ;BR IF YES
 9160  047452  000756                        BR      1$              ;TRY NEXT CYL
 9161
 9162  047454  012603                4$:     MOV     (SP)+,R3        ;RESTORE R3
 9163  047456  062716  000004                ADD     #4,(SP)         ;INCREMENT RET ADDR
 9164  047462  000207                        RTS     PC
 9165                                ;
 9166                                ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
 9167                                ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
 9168                                ;
 9169  047464  010046                SORT:   MOV     R0,-(SP)        ;SAVE R0
 9170  047466  010146                        MOV     R1,-(SP)        ;SAVE R1
 9171  047470  004737  046166                JSR     PC,RDSEC
 9172  047474  062737  000001  001406        ADD     #1,SECTOR
 9173  047502  004737  047572                JSR     PC,MULT6        ;MULT SECTOR BY 6
 9174
 9175  047506  012700  000204                MOV     #132.,R0
 9176  047512  163700  001406                SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
 9177  047516  010037  001406                MOV     R0,SECTOR
 9178  047522  062737  001674  001406        ADD     #RHTAB,SECTOR   ;SAVE INDEX
 9179
 9180  047530  062700  001674                ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
 9181  047534  012701  002100                MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
 9182
 9183  047540  012021                1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
 9184  047542  020027  002100                CMP     R0,#RHTAB+132.
 9185  047546  001374                        BNE     1$
 9186
 9187  047550  012700  001674                MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
 9188  047554  012021                2$:     MOV     (R0)+,(R1)+
 9189  047556  020037  001406                CMP     R0,SECTOR
```

```
 9190  047562  001374                        BNE     2$
 9191
 9192  047564  012601                        MOV     (SP)+,R1        ;RESTOR R1
 9193  047566  012600                        MOV     (SP)+,R0        ;RESTOR R0
 9194  047570  000207                        RTS     PC
 9195
 9196                                  ;
 9197                                  ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
 9198                                  ;
 9199  047572  006337  001406         MULT6:  ASL     SECTOR          ;2 X SECTOR
 9200  047576  013746  001406                 MOV     SECTOR,-(SP)
 9201  047602  006337  001406                 ASL     SECTOR          ;4 X SECTOR
 9202  047606  062637  001406                 ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
 9203  047612  000207                        RTS     PC
 9204
 9205                                  ;
 9206                                  ;ROUTINE TO TURN L OR P CLOCK INTERRORUPT ON
 9207                                  ;
 9208  047614  005037  001376         CLKON:  CLR     TIMUP
 9209  047620  005737  003500                 TST     PCLKF
 9210  047624  001004                        BNE     1$              ;BRANCH IF P-CLOCK PRESENT
 9211  047626  012777  000100  131472       MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
 9212  047634  000207                        RTS     PC
 9213  047636  012777  177777  131456 1$:    MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
 9214  047644  012777  000135  131446       MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
 9215  047652  000207                        RTS     PC              ;LINE FREQ & RUN
 9216
 9217                                  ;
 9218                                  ;KW11-L & KW11-P INTERRORUPT HANDLER
 9219  047654  005037  001376         CLOCK:  CLR     TIMUP
 9220  047660  005337  001372                 DEC     COUNT
 9221  047664  001010                        BNE     1$
 9222  047666  013737  001370  001372       MOV     HZ,COUNT
 9223  047674  005337  001374                 DEC     SEC
 9224  047700  001002                        BNE     1$
 9225  047702  005237  001376                 INC     TIMUP           ;SORRY, TIME IS UP
 9226  047706  000002                 1$:    RTI
 9227
 9228                                  ;ROUTINE TO TURN L OR P CLOCK INTERRORUPT OFF
 9229                                  ;
 9230  047710  005737  003500         CLKOF:  TST     PCLKF
 9231  047714  001003                        BNE     1$              ;BRACH IF P-CLOCK PRESENT
 9232  047716  005077  131404                 CLR     @LKS            ;L-CLOCK, CLEAR INTERRORUPT
 9233  047722  000207                        RTS     PC
 9234  047724  005077  131370         1$:    CLR     @PKS            ;P-CLOCK, CLEAR INTERRORUPT
 9235  047730  000207                        RTS     PC
 9236
 9237
 9238                                  ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MSGS
 9239                                  ;ENTER WITH THE EXPECTED WORD IN TEMP1
 9240                                  ;  TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
 9241                                  ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
 9242                                  ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
 9243                                  ;  THE PARITY BIT IS NOT SET IN B
 9244                                  ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
 9245                                  ;SET IN TEMP1
```

```
                                ;
047732  010046                  $BPAR:  MOV     R0,-(SP)        ;SAVE R0
047734  010146                          MOV     R1,-(SP)        ;SAVE R1
047736  012700  000021                  MOV     #17.,R0         ;SHIFT COUNTER
047742  005001                          CLR     R1              ;COUNT # OF 1'S IN TEMP1
047744  000241                          CLC                     ;CLEAR CARRY
                                ;
047746  006137  003372          1$:     ROL     TEMP1
047752  103001                          BCC     2$              ;BR IF CARRY CLEAR
047754  005201                          INC     R1              ;COUNT # OF 1'S
047756  005300                  2$:     DEC     R0              ;SHIFT COUNTER
047760  001372                          BNE     1$
                                ;
047762  032701  000001                  BIT     #BIT0,R1
047766  001003                          BNE     3$              ;BR IF ODD # IN R0
047770  052737  100000  003372          BIS     #M.PAR,TEMP1    ;SET PARITY BIT
047776  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
050000  012600                          MOV     (SP)+,R0        ;RESTORE R0
050002  000207                          RTS     PC
                                ;
                                ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                ;  IE: MY LOOP MACRO
                                ;
050004  032777  001000  131126  $SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
050012  001406                          BEQ     1$              ;BR IF NO
050014  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
050020  001403                          BEQ     1$              ;BR IF NO
050022  013716  001110                  MOV     $LPERR,(SP)
050026  000002                          RTI
                                ;
050030  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
050034  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
                                ;
050036  005037  001176          $WTST:  CLR     $ESCAPE
050042  005037  001410                  CLR     LPFLG
050046  032777  040000  131064          BIT     #SW14,@SWR      ;LOOP ON TEST?
050054  001403                          BEQ     3$              ;BR IF NO
050056  062716  000002          1$:     ADD     #2,(SP)
050062  000207                  2$:     RTS     PC
                                ;
050064  032777  000400  131046  3$:     BIT     #SW8,@SWR       ;LOOP ON SPECIFIC TEST?
050072  001773                          BEQ     2$              ;BR IF NO
050074  127737  131040  001102          CMPB    @SWR,$TSTNM     ;RIGHT TEST? SWR <7:0>
050102  001765                          BEQ     1$              ;BR IF YES
050104  000207                          RTS     PC
```

```
                                ;
                                ;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
                                ;IT IS USED TO ALLOW THE OPERATOR TO HALT THE CPU WHILE INSURING
                                ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
                                ;THE CPU.
                                ;
050106  022626          STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRORUPT

050110  004737  046106          JSR     PC,SUBCLR
050114  104024                  ERROR   24              ;CERR AFTER

050116  005737  003316          TST     UNLD            ;SEE IF HEADS UNLOADED
050122  001432                  BEQ     3$              ;BR IF NO
050124  005737  000042          TST     42              ;SEE IF MANUAL OR AUTO MODE
050130  001403                  BEQ     1$              ;BR IF MANUAL MODE
050132  104401  057151          TYPE    ,MSG74          ;PGM ABORT PENDING
050136  000402                  BR      2$
050140  104401  057200  1$:     TYPE    ,MSG75          ;HALT PENDING
050144                  2$:

050144  004737  046106          JSR     PC,SUBCLR
050150  104024                  ERROR   24              ;CERR AFTER SCLR

050152  012765  000011  000000  MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
050160  013737  001414  003372  MOV     T10,TEMP1       ;SET TIMEOUT
050166  004737  044176          JSR     PC,FRDY         ;FIND RDY
050172  104121                  ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.

050174  013737  001416  003374  MOV     T100,TEMP2      ;SETUP  TIMEOUT
050202  004737  044512          JSR     PC,FATT1        ;FIND ATTN
050206  104067                  ERROR   67              ;NO ATTN AFTER ST SPIN CMD.

050210  005737  003320  3$:     TST     BADHDR          ;SEE IF HEADERS VALID
050214  001466                  BEQ     4$              ;BR IF YES
050216  005237  003322          INC     HPEND

050222  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
050230  013765  001222  000010  MOV     $UNIT,RKCS2(R5)
050236  012765  000013  000000  MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                ;IN RKMR2 & RKMR3 RESP.
050244  013737  001414  003372  MOV     T10,TEMP1
050252  004737  044176          JSR     PC,FRDY         ;FIND RDY
050256  104124                  ERROR   124             ;RDY NOT SET AFTER RECAL CMD

050260  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
050266  004737  045534          JSR     PC,GSTAT
050272  032737  020000  003362  BIT     #D.RTZ,HMR2
050300  001001                  BNE     64$
050302  104244                  ERROR   244             ;RTZ NOT SET DURING RECAL CMD
050304  013737  001412  003374  64$:    MOV     T1,TEMP2        ;SETUP TIMEOUT
050312  004737  044512          JSR     PC,FATT1        ;FIND ATTN
050316  104055                  ERROR   55              ;NO ATTN AFTER RECAL CMD

050320  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
```

```
 9358  050326  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 9359  050334  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
 9360  050342  013737  001414  003372          MOV     T10,TEMP1
 9361  050350  004737  044176                  JSR     PC,FRDY         ;FIND RDY
 9362  050354  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 9363  050356  004737  044460                  JSR     PC,TSTATN       ;TEST FOR ATTN
 9364  050362  000401                          BR      65$
 9365  050364  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 9366  050366                          65$:
 9367
 9368
 9369  050366  000137  031646                  JMP     FORM            ;WRITE VALID FORMATS
 9370
 9371  050372  005737  000042          4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
 9372  050376  001406                          BEQ     5$              ;BR IF MANUAL MODE
 9373  050400  104401  057222                  TYPE    ,MSG76          ;PGM ABORTED
 9374  050404  005037  043544                  CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
 9375  050410  000137  043470                  JMP     $EOP
 9376
 9377  050414  104401  057240          5$:     TYPE    ,MSG77          ;CPU HALTED
 9378  050420  000000                          HALT
 9379  050422  000137  007740                  JMP     ST5             ;START OVER IF CONTINUE PRESSED
 9380
 9381                                  .SBTTL  UNEXPECTED TIMEOUT HANDLER
 9382
 9383                                  ;
 9384                                  ;THIS ROUTINE IS ENTERED IF THERE IS
 9385                                  ;       A.  NON EXISTANT MEMORY (NO SSYN)
 9386                                  ;       B.  BOUNDRY ERROR
 9387                                  ;       C.  STACK OVERFLOW
 9388                                  ;
 9389
 9390  050426  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
 9391  050430  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE
 9392  050432  032777  020000  130500          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPOUT?
 9393  050440  001005                          BNE     1$              ;YES, DON'T TYPE
 9394  050442  104401  057416                  TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
 9395  050446  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 9396                                                                  ;;TYPE PC
 9397  050450  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 9398  050452     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
 9399  050453     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 9400  050454  032777  001000  130456  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
 9401  050462  001403                          BEQ     2$              ;NO, BRANCH
 9402  050464  022626                          CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
 9403  050466  000177  130414                  JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
 9404                                                                  ;THAT GAVE BAD TIMEOUT
 9405  050472  032777  040000  130440  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
 9406  050500  001401                          BEQ     3$              ;NO BRANCH
 9407  050502  000002                          RTI                     ;YES
 9408
 9409  050504  000000                  3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
 9410                                                                  ;AS INDICATED. YOU CAN LOOP ON
 9411                                                                  ;ERROR, LOOP ON TEST OR INHIBIT
 9412                                                                  ;ERROR TYPEOUT BY SETTING THOSE
 9413                                                                  ;SWITCHES.
```

```
                                                                         ;
050506  022626                                  CMP     (SP)+,(SP)+      ;RESTORE STACK
050510  000137  043516                          JMP     $EOP1            ;ABORT TESTS

                                        .SBTTL  MEMORY CHECK ENABLE TRAP

050514  012737  050530  001176  MEMERR: MOV     #1$,$ESCAPE      ;LOAD ESCAPE
050522  011637  001334                          MOV     (SP),TRAPPC      ;STORE PC
050526  104236                                  ERROR   236              ;UNEXP MEM PARITY TRAP

050530  005037  001176          1$:     CLR     $ESCAPE
050534  032777  001000  130376          BIT     #SW9,@SWR        ;CHECK IF LOOP ON ERROR
050542  001001                          BNE     2$               ;YES, FORCE STACK AND TRY AGAIN
050544  000002                          RTI                      ;ELSE RETURN

050546  012706  001100          2$:     MOV     #STACK,SP        ;INIT STACK
050552  000177  130332                  JMP     @$LPERR          ;LOOP ON ERROR

                                        .SBTTL  RK06 INTERRUPT HANDLER

050556  011600                  INTER:  MOV     (SP),R0          ;SAVE PC WHERE INT OCCURRED.
050560  005740                          TST     -(R0)            ;GET PC BEFORE UPDATE.
050562  104401  056163                  TYPE    ,MSG6            ;INT AT PC=
050566  010046                          MOV     R0,-(SP)         ;;SAVE R0 FOR TYPEOUT
                                                                 ;;TYPE PC
050570  104403                          TYPOS                    ;;GO TYPE--OCTAL ASCII
050572     006                          .BYTE   6                ;;TYPE 6 DIGIT(S)
050573     000                          .BYTE   0                ;;SUPPRESS LEADING ZEROS
050574  000002                          RTI

                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;POWER DOWN ROUTINE

050576  012737  050610  000024  $PWRDN: MOV     #$PWRUP,PWRVEC   ;SET UP VECTOR
050604  000000                          HALT
050606  000776                          BR      .-2              ;HANG UP.

                                        ;POWER UP ROUTINE

050610  005037  050662          $PWRUP: CLR     $PWRCT           ;WAIT LOOP FOR TTY
050614  005237  050662          1$:     INC     $PWRCT           ;WAIT FOR THE INCR
050620  001375                          BNE     1$               ;OF WORD
050622  012737  050576  000024          MOV     #$PWRDN,PWRVEC   ;SET POWER DOWN VECTOR
050630  012737  000340  000026          MOV     #PR7,PWRVEC+2    ;PRIORITY 7
050636  012737  000340  000036          MOV     #PR7,TRAPVEC+2   ;LOCKOUT ALL INTERRORUPTS FOR TRAPS
050644  012706  001100                  MOV     #STACK,SP        ;INITIALIZE STACK
050650  104401  056347                  TYPE    ,MSG11           ;REPORT POWER FAIL
050654  000005                          RESET
050656  000137  011646                  JMP     PFSRT

050662  000000                  $PWRCT: 0                        ;WAIT COUNT FOR TTY
```

```
                                .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

050664                          $SCOPE:
050664  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
050666  032777  040000  130244  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
050674  001114                         BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
050676  000416                $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
050700  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
050704  012737  050724  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
050712  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
050716  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050722  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
050724  022626                5$:      CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
050726  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050732  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
050734                          6$:;#####END OF CODE FOR THE XOR TESTER#####
050734  032777  000400  130176         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
050742  001404                         BEQ     2$              ;;BR IF NO
050744  127737  130170  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
050752  001465                         BEQ     $OVER           ;;BR IF YES
050754  105737  001103         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
050760  001421                         BEQ     3$              ;;BR IF NO
050762  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
050770  101015                         BHI     3$              ;;BR IF NO
050772  032777  001000  130140         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
051000  001404                         BEQ     4$              ;;BR IF NO
051002  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
051010  000446                         BR      $OVER
051012  105037  001103         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
051016  005037  001174                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
051022  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
051024  032777  004000  130106  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
051032  001011                         BNE     1$              ;;BR IF YES
051034  005737  001216                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
051040  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
051042  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
051046  023737  001174  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
051054  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
051056  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
051064  013737  051142  001174         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
051072  105237  001102         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
051076  113737  001102  001214         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 9523  051104  011637  001106                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 9524  051110  011637  001110                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 9525  051114  005037  001176                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 9526  051120  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 9527  051126  013777  001102  130006  $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 9528  051134  013716  001106                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 9529  051140  000002                        RTI                     ;;FIXES PS
 9530  051142  003720                $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 9531                                 .SBTTL  ERROR HANDLER ROUTINE
 9532
 9533                                 ;;**************************************************************
 9534                                 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 9535                                 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 9536                                 ;*AND GO TO TYPERR ON ERROR
 9537                                 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 9538                                 ;*SW15=1        HALT ON ERROR
 9539                                 ;*SW13=1        INHIBIT ERROR TYPEOUTS
 9540                                 ;*SW10=1        BELL ON ERROR
 9541                                 ;*SW09=1        LOOP ON ERROR
 9542                                 ;*CALL
 9543                                 ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 9544
 9545  051144                         $ERROR:
 9546  051144  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9547  051146  105237  001103        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 9548  051152  001775                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 9549  051154  013777  001102  127760        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 9550  051162  032777  002000  127750        BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 9551  051170  001402                        BEQ     1$              ;;NO - SKIP
 9552  051172  104401  001200                TYPE    ,$BELL          ;;RING BELL
 9553  051176  005237  001112        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 9554  051202  011637  001116                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 9555  051206  162737  000002  001116        SUB     #2,$ERRPC
 9556  051214  117737  127676  001114        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 9557  051222  032777  020000  127710        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 9558  051230  001004                        BNE     20$             ;;SKIP TYPEOUTS
 9559  051232  004737  067610                JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 9560  051236  104401  001205                TYPE    ,$CRLF
 9561  051242                         20$:
 9562  051242  122737  000001  001230        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9563  051250  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
 9564  051252  113737  001114  051264        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 9565  051260  004737  052070                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 9566  051264     000         21$:    .BYTE   0
 9567  051265     000                 .BYTE   0
 9568  051266  000777         22$:    BR      22$             ;;APT ERROR LOOP
 9569  051270  005777  127644  2$:     TST     @SWR            ;;HALT ON ERROR
 9570  051274  100002                        BPL     3$              ;;SKIP IF CONTINUE
 9571  051276  000000                        HALT                    ;;HALT ON ERROR!
 9572  051300  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9573  051302  032777  001000  127630  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 9574  051310  001402                        BEQ     4$              ;;BR IF NO
 9575  051312  013716  001110                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 9576  051316  005737  001176        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 9577  051322  001402                        BEQ     5$              ;;BR IF NONE
 9578  051324  013716  001176                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
 9579  051330                        5$:
 9580  051330  022737  043604  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 9581  051336  001001                          BNE     6$              ;;BRANCH IF NO
 9582  051340  000000                          HALT                    ;;YES
 9583  051342                        6$:
 9584  051342  000002                          RTI                     ;;RETURN
 9585                                .SBTTL  TYPE ROUTINE
 9586
 9587                                ;;*****************************************************************
 9588                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 9589                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 9590                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 9591                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 9592                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 9593                                ;*
 9594                                ;*CALL:
 9595                                ;*1) USING A TRAP INSTRUCTION
 9596                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 9597                                ;*OR
 9598                                ;*      TYPE
 9599                                ;*      MESADR
 9600                                ;*
 9601
 9602  051344  105737  001157        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 9603  051350  100002                          BPL     1$              ;;BR IF YES
 9604  051352  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 9605  051354  000430                          BR      3$              ;;LEAVE
 9606  051356  010046                1$:       MOV     R0,-(SP)        ;;SAVE R0
 9607  051360  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 9608  051364  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9609  051372  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 9610  051374  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 9611  051402  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 9612  051404  010037  051414                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 9613  051410  004737  052060                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 9614  051414  000000                61$:      .WORD   0               ;;MESSAGE ADDRESS
 9615  051416  132737  000040  001231  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 9616  051424  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
 9617  051426  112046                2$:       MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 9618  051430  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 9619  051432  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 9620  051434  012600                60$:      MOV     (SP)+,R0        ;;RESTORE R0
 9621  051436  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 9622  051442  000002                          RTI                     ;;RETURN
 9623  051444  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 9624  051450  001430                          BEQ     8$
 9625  051452  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 9626  051456  001006                          BNE     5$
 9627  051460  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 9628  051462  104401                          TYPE                    ;;TYPE A CR AND LF
 9629  051464  001205                          $CRLF
 9630  051466  105037  051622                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 9631  051472  000755                          BR      2$              ;;GET NEXT CHARACTER
 9632  051474  004737  051556          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 9633  051500  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 9634  051504  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
 9635  051506  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 9636                                                                  ;;AND THE NULL CHAR.
 9637  051512  105366  000001          7$:   DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 9638  051516  002770                      BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 9639  051520  004737  051556              JSR     PC,$TYPEC       ;;GO TYPE A NULL
 9640  051524  105337  051622              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 9641  051530  000770                      BR      7$              ;;LOOP
 9642
 9643                                    ;HORIZONTAL TAB PROCESSOR
 9644
 9645  051532  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 9646  051536  004737  051556          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
 9647  051542  132737  000007  051622        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 9648  051550  001372                      BNE     9$              ;;TAB STOP
 9649  051552  005726                      TST     (SP)+           ;;POP SPACE OFF STACK
 9650  051554  000724                      BR      2$              ;;GET NEXT CHARACTER
 9651  051556  105777  127366        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 9652  051562  100375                      BPL     $TYPEC
 9653  051564  116677  000002  127360      MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 9654  051572  122766  000015  000002      CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 9655  051600  001003                      BNE     1$              ;;BRANCH IF NO
 9656  051602  105037  051622              CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 9657  051606  000406                      BR      $TYPEX          ;;EXIT
 9658  051610  122766  000012  000002  1$: CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 9659  051616  001402                      BEQ     $TYPEX          ;;BRANCH IF YES
 9660  051620  105227                      INCB    (PC)+           ;;COUNT THE CHARACTER
 9661  051622  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
 9662  051624  000207        $TYPEX: RTS     PC
 9663
 9664                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 9665
 9666                                    ;;******************************************************************
 9667                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 9668                                    ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 9669                                    ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 9670                                    ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 9671                                    ;*REPLACED WITH SPACES.
 9672                                    ;*CALL:
 9673                                    ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 9674                                    ;*      TYPDS                   ;;GO TO THE ROUTINE
 9675
 9676  051626                          $TYPDS:
 9677  051626  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9678  051630  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9679  051632  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9680  051634  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9681  051636  010546                      MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9682  051640  012746  020200              MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 9683  051644  016605  000020              MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 9684  051650  100004                      BPL     1$              ;;BR IF INPUT IS POS.
 9685  051652  005405                      NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 9686  051654  112766  000055  000001      MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 9687  051662  005000                  1$: CLR     R0              ;;ZERO THE CONSTANTS INDEX
 9688  051664  012703  052042              MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 9689  051670  112723  000040              MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 9690  051674  005002                  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 EXIY
                                        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.
                                $DBLK:  .BLKW   4
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;**********************************************************************
                                $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
                                $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
                                        BR      $ATYC
                                $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
                                $ATYC:
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                        TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
                                        BEQ     5$              ;;IF NOT:  BR
                                        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
                                        BNE     3$              ;;IF NOT:  BR
                                        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
                                        BEQ     3$              ;;IF NOT:  BR
```

```
 9747  052130  017600  000004                 MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 9748  052134  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9749  052142  005737  001210         1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 9750  052146  001375                 BNE     1$              ;;IF NOT:  WAIT
 9751  052150  010037  001224         MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 9752  052154  105720                 2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 9753  052156  001376                 BNE     2$
 9754  052160  163700  001224         SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 9755  052164  006200                 ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 9756  052166  010037  001226         MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 9757  052172  012737  000004  001210 MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 9758  052200  000413                 BR      5$
 9759  052202  017637  000004  052226 3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 9760  052210  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 9761  052216  013746  177776         MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 9762  052222  004737  051344         JSR     PC,$TYPE        ;;CALL TYPE MACRO
 9763  052226  000000                 4$:     .WORD   0
 9764  052230                         5$:
 9765  052230  105737  052316         10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9766  052234  001416                 BEQ     12$             ;;IF NOT:  BR
 9767  052236  005737  001230         TST     $ENV            ;;RUNNING UNDER APT?
 9768  052242  001413                 BEQ     12$             ;;IF NOT:  BR
 9769  052244  005737  001210         11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9770  052250  001375                 BNE     11$             ;;IF NOT:  WAIT
 9771  052252  017637  000004  001212 MOV     @4(SP),$FATAL   ;;GET ERROR #
 9772  052260  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9773  052266  005237  001210         INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9774  052272  105037  052316         12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 9775  052276  105037  052315         CLRB    $LFLG           ;;CLEAR LOG FLAG
 9776  052302  105037  052314         CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9777  052306  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
 9778  052310  012600                 MOV     (SP)+,R0        ;;POP STACK INTO R0
 9779  052312  000207                 RTS     PC              ;;RETURN
 9780  052314     000         $MFLG:  .BYTE   0               ;;MESSG. FLAG
 9781  052315     000         $LFLG:  .BYTE   0               ;;LOG FLAG
 9782  052316     000         $FFLG:  .BYTE   0               ;;FATAL FLAG
 9783          052320                 .EVEN
 9784          000200         APTSIZE=200
 9785          000001         APTENV=001
 9786          000100         APTSPOOL=100
 9787          000040         APTCSUP=040
 9788                         .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 9789
 9790                         ;;**************************************************************
 9791                         ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 9792                         ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 9793                         ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 9794                         ;*CALL:
 9795                         ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9796                         ;*      TYPOS                   ;;CALL FOR TYPEOUT
 9797                         ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 9798                         ;*      .BYTE   M               ;;M=1 OR 0
 9799                         ;*                                      ;;1=TYPE LEADING ZEROS
 9800                         ;*                                      ;;0=SUPPRESS LEADING ZEROS
 9801                         ;*
 9802                         ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
 9803                                   ;*$TYPOS OR $TYPOC
 9804                                   ;*CALL:
 9805                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9806                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
 9807                                   ;*
 9808                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 9809                                   ;*CALL:
 9810                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9811                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
 9812
 9813  052320  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 9814  052324  116637  000001  052543           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 9815  052332  112637  052545                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 9816  052336  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 9817  052342  000406                           BR      $TYPON
 9818  052344  112737  000001  052543  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 9819  052352  112737  000006  052545          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 9820  052360  112737  000005  052542  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 9821  052366  010346                           MOV     R3,-(SP)        ;;SAVE R3
 9822  052370  010446                           MOV     R4,-(SP)        ;;SAVE R4
 9823  052372  010546                           MOV     R5,-(SP)        ;;SAVE R5
 9824  052374  113704  052545                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 9825  052400  005404                           NEG     R4
 9826  052402  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 9827  052406  110437  052544                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 9828  052412  113704  052543                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 9829  052416  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 9830  052422  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
 9831  052424  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 9832  052426  000404                           BR      3$              ;;GO DO MSB
 9833  052430  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
 9834  052432  006105                           ROL     R5
 9835  052434  006105                           ROL     R5
 9836  052436  010503                           MOV     R5,R3
 9837  052440  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 9838  052442  105337  052544                   DECB    $OMODE          ;;TYPE THIS DIGIT?
 9839  052446  100016                           BPL     7$              ;;BR IF NO
 9840  052450  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
 9841  052454  001002                           BNE     4$              ;;TEST FOR 0
 9842  052456  005704                           TST     R4              ;;SUPPRESS THIS 0?
 9843  052460  001403                           BEQ     5$              ;;BR IF YES
 9844  052462  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 9845  052464  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 9846  052470  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 9847  052474  110337  052540                   MOVB    R3,8$           ;;SAVE FOR TYPING
 9848  052500  104401  052540                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
 9849  052504  105337  052542           7$:     DECB    $OCNT           ;;COUNT BY 1
 9850  052510  003347                           BGT     2$              ;;BR IF MORE TO DO
 9851  052512  002402                           BLT     6$              ;;BR IF DONE
 9852  052514  005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 9853  052516  000744                           BR      2$              ;;GO DO THE LAST DIGIT
 9854  052520  012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
 9855  052522  012604                           MOV     (SP)+,R4        ;;RESTORE R4
 9856  052524  012603                           MOV     (SP)+,R3        ;;RESTORE R3
 9857  052526  016666  000002  000004           MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 9858  052534  012616                           MOV     (SP)+,(SP)
```

```
 9859  052536  000002                  RTI                             ;;RETURN
 9860  052540     000  8$:     .BYTE   0                               ;;STORAGE FOR ASCII DIGIT
 9861  052541     000          .BYTE   0                               ;;TERMINATOR FOR TYPE ROUTINE
 9862  052542     000  $OCNT:  .BYTE   0                               ;;OCTAL DIGIT COUNTER
 9863  052543     000  $OFILL: .BYTE   0                               ;;ZERO FILL SWITCH
 9864  052544  000000  $OMODE: .WORD   0                               ;;NUMBER OF DIGITS TO TYPE
 9865                  .SBTTL  TTY INPUT ROUTINE
 9866
 9867                  ;;******************************************************************
 9868                  .ENABL  LSB
 9869  052546  000000  $TKCNT: .WORD   0                               ;;NUMBER OF ITEMS IN QUEUE
 9870  052550  000000  $TKQIN: .WORD   0                               ;;INPUT POINTER
 9871  052552  000000  $TKQOUT: .WORD  0                               ;;OUTPUT POINTER
 9872  052554  000001  $TKQSRT: .BLKB  1                               ;;TTY KEYBOARD QUEUE
 9873          052555  $TKQEND=.
 9874          052556  .EVEN
 9875
 9876                  ;*TK INITIALIZE ROUTINE
 9877                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
 9878                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
 9879                  ;
 9880                  ;*CALL:
 9881                  ;*      JSR     PC,$TKINT
 9882                  ;*      RETURN
 9883                  ;
 9884  052556  005037  052546  $TKINT: CLR     $TKCNT                  ;;CLEAR COUNT OF ITEMS IN QUEUE
 9885  052562  012737  052554  052550  MOV     #$TKQSRT,$TKQIN         ;;MOVE THE STARTING ADDRESS OF THE
 9886  052570  013737  052550  052552  MOV     $TKQIN,$TKQOUT          ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
 9887  052576  012737  052626  000060  MOV     #$TKSRV,@#TKVEC         ;;INITIALIZE THE KEYBOARD VECTOR
 9888  052604  012737  000200  000062  MOV     #200,@#TKVEC+2          ;;"BR" LEVEL 4
 9889  052612  005777  126330          TST     @$TKB                   ;;CLEAR DONE FLAG
 9890  052616  012777  000100  126320  MOV     #100,@$TKS              ;;ENABLE TTY KEYBOARD INTERRUPT
 9891  052624  000207                  RTS     PC                      ;;RETURN TO CALLER
 9892
 9893                  ;*TK SERVICE ROUTINE
 9894                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
 9895                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
 9896                  ;*IT IN THE QUEUE.
 9897                  ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
 9898                  ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
 9899                  ;
 9900  052626  117746  126314  $TKSRV: MOVB    @$TKB,-(SP)             ;;PICKUP THE CHARACTER
 9901  052632  042716  177600          BIC     #↑C177,(SP)             ;;STRIP THE JUNK
 9902  052636  021627  000003          CMP     (SP),#3                 ;;IS IT A CONTROL C?
 9903  052642  001007                  BNE     1$                      ;;BRANCH IF NO
 9904  052644  104401  053754          TYPE    ,$CNTLC                 ;;TYPE A CONTROL-C (↑C)
 9905  052650  004737  052556          JSR     PC,$TKINT               ;;INIT THE KEYBOARD
 9906  052654  005726                  TST     (SP)+                   ;;CLEAN UP STACK
 9907  052656  000137  050106          JMP     STOP                    ;;CONTROL C RESTART
 9908  052662  021627  000007  1$:     CMP     (SP),#7                 ;;IS IT A CONTROL G?
 9909  052666  001004                  BNE     2$                      ;;BRANCH IF NO
 9910  052670  022737  000176  001140  CMP     #SWREG,SWR              ;;IS SOFT-SWR SELECTED?
 9911  052676  001500                  BEQ     6$                      ;;GO TO SWR CHANGE
 9912
 9913  052700          2$:
 9914  052700  022737  000001  052546  CMP     #1,$TKCNT               ;;IS THE QUEUE FULL?
```

```
                                        BNE     3$              ;;BRANCH IF NO
                                        TYPE    ,$BELL          ;;RING THE TTY BELL
                                        IST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
                                        BR      5$              ;;EXIT
                                3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
                                        BNE     32$             ;;BRANCH IF NO
                                        CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
                                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
                                31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
                                        BPL     31$             ;;LOOP UNTIL ITS THERE
                                        MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
                                        BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
                                        BNE     31$             ;;BRANCH IF NO
                                        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
                                        RTI                     ;;RETURN
                                32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
                                        CMP     (SP),#140       ;;IS IT UPPER CASE?
                                        BLT     4$              ;;BRANCH IF YES
                                        CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
                                        BGT     4$              ;;BRANCH IF YES
                                        BIC     #40,(SP)        ;;MAKE IT UPPER CASE
                                4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
                                        INC     $TKQIN          ;;UPDATE THE POINTER
                                        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
                                        BNE     5$              ;;BRANCH IF NO
                                        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
                                5$:     RTI                     ;;RETURN

                        ;;**************************************************************
                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                        ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                        ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
                $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
                        BNE     15$             ;;EXIT IF NOT
                        TSTB    @$TKS           ;;IS A CHAR WAITING?
                        BPL     15$             ;;IF NOT, EXIT
                        MOVB    @$TKB,-(SP)     ;;YES
                        BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
                        CMP     (SP),#7         ;;IS IT A CONTROL-G?
                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
                                                ;;AND EXIT

                        ;;**************************************************************
                        ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                        ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                        ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
                6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
                        BEQ     2$              ;;BRANCH IF YES
                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
                        JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
                        CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
                        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

                        TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
```

```
 9971  053134  104401  053773          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 9972  053140  013746  000176                  MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
 9973  053144  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9974  053146  104401  054004                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 9975  053152  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 9976  053154  005046                          CLR     -(SP)           ;;THE NEW SWR
 9977  053156  105777  125762          7$:     TSTB    @$TKS           ;;CHAR THERE?
 9978  053162  100375                          BPL     7$              ;;IF NOT TRY AGAIN
 9979
 9980  053164  117746  125756                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 9981  053170  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9982
 9983  053174  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
 9984  053200  001015                          BNE     9$              ;;BRANCH IF NOT
 9985  053202  104401  053754                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
 9986  053206  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
 9987  053212  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
 9988  053220  001003                          BNE     8$              ;;BRANCH IF NO
 9989  053222  012777  000100  125714          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
 9990  053230  000137  050106          8$:     JMP     STOP            ;;CONTROL-C RESTART
 9991
 9992
 9993  053234  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 9994  053240  001005                          BNE     10$             ;;BRANCH IF NOT
 9995  053242  104401  053761                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 9996  053246  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 9997  053252  000737                          BR      19$             ;;LET'S TRY IT AGAIN
 9998
 9999
10000  053254  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
10001  053260  001022                          BNE     16$             ;;BRANCH IF NO
10002  053262  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
10003  053266  001403                          BEQ     11$             ;;BRANCH IF YES
10004  053270  016677  000002  125642          MOV     2(SP),@SWR      ;;SAVE NEW SWR
10005  053276  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
10006  053302  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
10007  053306  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
10008  053314  001003                          BNE     15$             ;;BRANCH IF NOT
10009  053316  012777  000100  125620          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
10010  053324  000002          15$:    RTI                             ;;RETURN
10011  053326  004737  051556          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
10012  053332  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
10013  053336  002420                          BLT     18$             ;;BRANCH IF YES
10014  053340  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
10015  053344  003015                          BGT     18$             ;;BRANCH IF YES
10016  053346  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
10017  053352  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
10018  053356  001403                          BEQ     17$             ;;BRANCH IF YES
10019  053360  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
10020  053362  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
10021  053364  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
10022  053366  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
10023  053372  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
10024  053376  000667                          BR      7$              ;;GET THE NEXT ONE
10025  053400  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
10026  053404  000720                          BR      20$             ;;SIMULATE CONTROL-U
```

```
10027                                   .DSABL  LSB
10028
10029
10030                                   ;;*****************************************************************
10031                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10032                                   ;*CALL:
10033                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
10034                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10035                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
10036                                   ;
10037
10038  053406  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
10039  053410  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
10040  053416  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
10041  053422  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
10042  053424  012746  053432                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
10043  053430  000002                          RTI                     ;;POP NEW PC AND PS
10044  053432                          64$:
10045  053432  005737  052546          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
10046  053436  001775                          BEQ     1$
10047  053440  005337  052546                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
10048  053444  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
10049  053452  005237  052552                  INC     $TKQOUT         ;;UPDATE THE POINTER
10050  053456  023727  052552  052555          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
10051  053464  001003                          BNE     2$              ;;BRANCH IF NO
10052  053466  012737  052554  052552          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
10053  053474  000002                  2$:     RTI                     ;;RETURN
10054                                   ;;*****************************************************************
10055                                   ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10056                                   ;*CALL:
10057                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10058                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10059                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10060
10061  053476  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
10062  053500  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
10063  053502  012703  053732          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
10064  053506  022703  053754          2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
10065  053512  101456                          BLOS    4$              ;;BR IF YES
10066  053514  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10067  053516  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
10068  053520  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
10069  053524  001022                          BNE     5$              ;;BR IF NO
10070  053526  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
10071  053530  001007                          BNE     6$              ;;BR IF NO
10072  053532  112737  000134  053730          MOVB    #'\,9$          ;;TYPE A BACK SLASH
10073  053540  104401  053730                  TYPE    ,9$
10074  053544  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
10075  053550  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
10076  053552  020327  053732                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
10077  053556  103434                          BLO     4$              ;;BR IF YES
10078  053560  111337  053730                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
10079  053564  104401  053730                  TYPE    ,9$             ;;GO TYPE
10080  053570  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
10081  053572  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
10082  053574  001406                          BEQ     7$              ;;BR IF NO
```

```
10083  053576  112737  000134  053730          MOVB    #'\,9$          ;;TYPE A BACK SLASH
10084  053604  104401  053730                  TYPE    9$
10085  053610  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
10086  053612  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
10087  053616  001003                          BNE     8$              ;;BR IF NO
10088  053620  104401  053761                  TYPE    $CNTLU          ;;TYPE A CONTROL "U"
10089  053624  000726                          BR      1$              ;;GO START OVER
10090  053626  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
10091  053632  001011                          BNE     3$              ;;BRANCH IF NO
10092  053634  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
10093  053636  104401  001205                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
10094  053642  104401  053732                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
10095  053646  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
10096  053650  104401  001204          4$:     TYPE    $QUES           ;;TYPE A '?'
10097  053654  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
10098  053656  111337  053730          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
10099  053662  104401  053730                  TYPE    ,9$
10100  053666  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10101  053672  001305                          BNE     2$              ;;LOOP IF NOT RETURN
10102  053674  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10103  053700  104401  001206                  TYPE    $LF             ;;TYPE A LINE FEED
10104  053704  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
10105  053706  012603                          MOV     (SP)+,R3        ;;RESTORE R3
10106  053710  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10107  053712  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10108  053720  012766  053732  000004          MOV     #$TTYIN,4(SP)
10109  053726  000002                          RTI                     ;;RETURN
10110  053730     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
10111  053731     000                  .BYTE   0                       ;;TERMINATOR
10112  053732  000022          $TTYIN: .BLKB   22                      ;;RESERVE 22 BYTES FOR TTY INPUT
10113  053754  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
10114  053761     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
10115  053766  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
10116  053773     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
10117  054000  036440  000040
10118  054004  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
10119  054012  020075     000
10120          054016                  .EVEN
10121                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
10122
10123                                  ;;*****************************************************************
10124                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
10125                                  ;*CHANGE IT TO BINARY.
10126                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
10127                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
10128                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
10129                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
10130                                  ;*CALL:
10131                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
10132                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
10133                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
10134
10135  054016  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
10136  054020  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
10137  054026  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10138  054030  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
10139  054032  010246                 1$:     MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10140  054034  104411                         RDLIN                   ;;READ AN ASCIZ LINE
10141  054036  012600                         MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
10142  054040  010037  054144                 MOV     R0,5$           ;;AND SAVE IT
10143  054044  005001                         CLR     R1              ;;CLEAR DATA WORD
10144  054046  005002                         CLR     R2
10145  054050  112046                 2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
10146  054052  001420                         BEQ     3$              ;;IF ZERO GET OUT
10147  054054  122716  000060                 CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
10148  054060  003026                         BGT     4$              ;;IS AN OCTAL DIGIT
10149  054062  122716  000067                 CMPB    #'7,(SP)
10150  054066  002423                         BLT     4$
10151  054070  006301                         ASL     R1              ;;*2
10152  054072  006102                         ROL     R2
10153  054074  006301                         ASL     R1              ;;*4
10154  054076  006102                         ROL     R2
10155  054100  006301                         ASL     R1              ;;*8
10156  054102  006102                         ROL     R2
10157  054104  042716  177770                 BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
10158  054110  062601                         ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
10159  054112  000756                         BR      2$              ;;LOOP
10160  054114  005726                 3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
10161  054116  010166  000012                 MOV     R1,12(SP)       ;;SAVE THE RESULT
10162  054122  010237  054154                 MOV     R2,$HIOCT
10163  054126  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
10164  054130  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
10165  054132  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
10166  054134  000002                         RTI                     ;;RETURN
10167  054136  005726                 4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
10168  054140  105010                         CLRB    (R0)            ;;SET A TERMINATOR
10169  054142  104401                         TYPE                    ;;TYPE UP THRU THE BAD CHAR.
10170  054144  000000                 5$:     .WORD   0
10171  054146  104401  001204                 TYPE    ,$QUES          ;;"?" "CR" & "LF"
10172  054152  000730                         BR      1$              ;;TRY AGAIN
10173  054154  000000                 $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
10174                                  .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
10175
10176                                  ;;*****************************************************************
10177                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
10178                                  ;*UNSIGNED OCTAL ASCIZ NUMBER.
10179                                  ;*CALL
10180                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
10181                                  ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
10182                                  ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
10183
10184
10185  054156  104413                 $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
10186  054160  016601  000002                 MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
10187  054164  012705  054275                 MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
10188  054170  012704  000014                 MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
10189  054174  012703  177770                 MOV     #^C7,R3         ;;MASK
10190  054200  012100                         MOV     (R1)+,R0        ;;LOWER WORD
10191  054202  012101                         MOV     (R1)+,R1        ;;HIGH WORD
10192  054204  005002                         CLR     R2              ;;TERMINATOR
10193  054206  110245                 1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
10194  054210  010002                         MOV     R0,R2           ;;GET THIS DIGIT
```

```
10195  054212  005304                          DEC     R4              ;;COUNT THIS CHARACTER
10196  054214  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
10197  054216  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
10198  054220  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
10199  054222  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
10200  054226  104414                          RESREG                  ;;RESTORE ALL REGISTERS
10201  054230  000207                          RTS     PC              ;;RETURN TO USER
10202  054232  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
10203  054234  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
10204  054236  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
10205  054240  006001                          ROR     R1
10206  054242  006000                          ROR     R0
10207  054244  006001                          ROR     R1
10208  054246  006000                          ROR     R0
10209  054250  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
10210  054252  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
10211  054256  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
10212  054260  000016                  $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
10213                                  .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
10214
10215                                  ;;*****************************************************************
10216                                  ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
10217                                  ;*UNSIGNED DECIMAL ASCIZ NUMBER.
10218                                  ;*CALL
10219                                  ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
10220                                  ;*      JSR     PC,@#$SB2D      ;;CALL
10221                                  ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
10222
10223
10224  054276  016637  000002  054326  $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
10225  054304  012746  054326                  MOV     #1$,-(SP)       ;;SET POINTER
10226  054310  004737  054332                  JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
10227  054314  062716  000005                  ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
10228  054320  012666  000002                  MOV     (SP)+,2(SP)     ;;PICKUP POINTER
10229  054324  000207                          RTS     PC              ;;RETURN
10230  054326  000000  000000          1$:     .WORD   0,0
10231                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
10232
10233                                  ;;*****************************************************************
10234                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
10235                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
10236                                  ;*POSITIVE.
10237                                  ;*CALL
10238                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
10239                                  ;*      JSR     PC,@#$DB2D
10240                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
10241                                                                  ;;IS ON THE STACK
10242
10243
10244  054332  104413                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
10245  054334  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
10246  054340  012700  054512                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
10247  054344  010066  000002                  MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
10248  054350  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
10249  054352  012202                          MOV     (R2)+,R2
10250  054354  012737  000012  054430          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
```

```
10251  054362  012704  054442          MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
10252  054366  012705  054444          MOV     #$TNPWR+2,R5
10253  054372  005003              1$: CLR     R3              ;;CLEAR PARTIAL
10254  054374  161401              2$: SUB     (R4),R1         ;;SUBTRACT TEN POWER
10255  054376  005602                  SBC     R2
10256  054400  161502                  SUB     (R5),R2
10257  054402  002402                  BLT     3$              ;;BR IF TEN POWER TO LARGE
10258  054404  005203                  INC     R3              ;;ADD 1 TO PARTIAL
10259  054406  000772                  BR      2$              ;;LOOP
10260  054410  062401              3$: ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
10261  054412  005502                  ADC     R2
10262  054414  062402                  ADD     (R4)+,R2
10263  054416  022525                  CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
10264  054420  052703  000060          BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
10265  054424  110320                  MOVB    R3,(R0)+        ;;SAVE IT
10266  054426  005327                  DEC     (PC)+           ;;DONE?
10267  054430  000000              4$: .WORD   0
10268  054432  001357                  BNE     1$              ;;BR IF NO
10269  054434  105020                  CLRB    (R0)+           ;;TERMINATOR
10270  054436  104414                  RESREG                  ;;RESTORE REGISTERS
10271  054440  000207                  RTS     PC              ;;RETURN
10272  054442  145000  $TNPWR: 145000                          ;;1.0E09
10273  054444  035632                  35632
10274  054446  160400                  160400                  ;;1.0E08
10275  054450  002765                  2765
10276  054452  113200                  113200                  ;;1.0E07
10277  054454  000230                  230
10278  054456  041100                  041100                  ;;1.0E06
10279  054460  000017                  17
10280  054462  103240                  103240                  ;;1.0E05
10281  054464  000001                  1
10282  054466  023420                  23420                   ;;1.0E04
10283  054470  000000                  0
10284  054472  001750                  1750                    ;;1.0E03
10285  054474  000000                  0
10286  054476  000144                  144                     ;;1.0E02
10287  054500  000000                  0
10288  054502  000012                  12                      ;;1.0E01
10289  054504  000000                  0
10290  054506  000001                  1                       ;;1.0E00
10291  054510  000000                  0
10292  054512  000014  $DECVL: .BLKB   12                      ;;RESERVE STORAGE FOR ASCIZ STRING
10293                  .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
10294
10295                  ;;**********************************************************************
10296                  ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
10297                  ;*LEADING NUMBERS.
10298                  ;*CALL
10299                  ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
10300                  ;*      JSR     PC,@#$SUPRS
10301
10302
10303  054526  010046  $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
10304  054530  016600  000004          MOV     4(SP),R0        ;;PICKUP THE POINTER
10305  054534  105710              1$: TSTB    (R0)            ;;TERMINATEOR?
10306  054536  001403                  BEQ     2$              ;;BR IF YES
```

```
10307  054540  122720  000060          CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
10308  054544  001773                  BEQ     1$              ;;BR IF YES
10309  054546  005300          2$:     DEC     R0              ;;BACKUP BY "1"
10310  054550  010037  054556          MOV     R0,3$           ;;SAVE FOR TYPING
10311  054554  104401                  TYPE                    ;;GO TYPE
10312  054556  000000          3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
10313  054560  012600                  MOV     (SP)+,R0        ;;RESTORE R0
10314  054562  012616                  MOV     (SP)+,(SP)      ;;RESTORE THE STACK
10315  054564  000207                  RTS     PC              ;;RETURN
10316                          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
10317
10318                          ;;**************************************************************
10319                          ;*SAVE R0-R5
10320                          ;*CALL:
10321                          ;*      SAVREG
10322                          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
10323                          ;*
10324                          ;*TOP---(+16)
10325                          ;* +2---(+18)
10326                          ;* +4---R5
10327                          ;* +6---R4
10328                          ;* +8---R3
10329                          ;*+10---R2
10330                          ;*+12---R1
10331                          ;*+14---R0
10332
10333  054566                  $SAVREG:
10334  054566  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10335  054570  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10336  054572  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10337  054574  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10338  054576  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10339  054600  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10340  054602  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
10341  054606  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
10342  054612  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
10343  054616  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
10344  054622  000002                  RTI
10345
10346                          ;*RESTORE R0-R5
10347                          ;*CALL:
10348                          ;*      RESREG
10349  054624                  $RESREG:
10350  054624  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
10351  054630  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
10352  054634  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
10353  054640  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
10354  054644  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
10355  054646  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
10356  054650  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
10357  054652  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10358  054654  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10359  054656  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10360  054660  000002                  RTI
10361                          .SBTTL  TRAP DECODER
10362
```

```
10363                                   ;;**********************************************************************
10364                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10365                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10366                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10367                                   ;*GO TO THAT ROUTINE.
10368
10369  054662  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
10370  054664  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
10371  054670  005740                          TST     -(R0)           ;;BACKUP BY 2
10372  054672  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
10373  054674  006300                          ASL     R0              ;;POSITION FOR INDEXING
10374  054676  016000  054716                 MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
10375  054702  000200                          RTS     R0              ;;GO TO ROUTINE
10376
10377
10378                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
10379
10380  054704  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
10381  054706  016666  000004  000002          MOV     4(SP),2(SF)     ;;MOVE THE PSW DOWN
10382  054714  000002                          RTI                     ;;RESTORE THE PSW
10383
10384                                   .SBTTL  TRAP TABLE
10385
10386                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10387                                   ;*BY THE "TRAP" INSTRUCTION.
10388
10389                                   ;       ROUTINE
10390                                   ;       -------
10391  054716  054704                  $TRPAD: .WORD   $TRAP2
10392  054720  051344                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
10393  054722  052344                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10394  054724  052320                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
10395  054726  052360                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
10396  054730  051626                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
10397
10398  054732  053134                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
10399
10400  054734  053044                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
10401  054736  053406                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
10402  054740  053476                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
10403  054742  054016                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
10404  054744  054566                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
10405  054746  054624                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
10406  054750  050004                          SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
10407
```

```
10408
10409                                   .SBTTL  SERVICE MSGS
10410                                   .EVEN
10411  054752  000062                   BSE22S: .BLKW   50.              ;22 SECTOR SOFTWARE INFO
10412
10413
10414  055116  005015  047125  041111   MSG1:   .ASCII  <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
10415  055124  051525  051040  030113
10416  055132  020066  051104  053111
10417  055140  020105  044504  043501
10418  055146  047516  052123  041511
10419  055154  005015  050011  051101           .ASCII  <CR><LF>/       PART 1  /
10420  055162  020124  004461
10421  055166  005015  040515  047111           .ASCII  <CR><LF>/MAINDEC-11-DZR6H-C-PB/<CR><LF>
10422  055174  042504  026503  030461
10423  055202  042055  051132  044066
10424  055210  041455  050055  006502
10425  055216      012
10426  055217      015  004412  025052           .ASCII  <CR><LF>/       *** CAUTION ***/<CR><LF>
10427  055224  020052  040503  052125
10428  055232  047511  020116  025052
10429  055240  006452      012
10430  055243      015  052012  044510           .ASCII  <CR><LF>/THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING CONTROL-C/
10431  055250  020123  051120  043517
10432  055256  040522  020115  044123
10433  055264  052517  042114  041040
10434  055272  020105  040510  052114
10435  055300  042105  047440  046116
10436  055306  020131  054502  052040
10437  055314  050131  047111  020107
10438  055322  047503  052116  047522
10439  055330  026514      103
10440  055333      015  047412  044124           .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND,OR THE DRIVE/
10441  055340  051105  044527  042523
10442  055346  020054  040503  052122
10443  055354  044522  043504  020105
10444  055362  047506  046522  052101
10445  055370  044524  043516  040440
10446  055376  042116  047454  020122
10447  055404  044124  020105  051104
10448  055412  053111      105
10449  055415      015  046412  054501           .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/<CR><LF>
10450  055422  041040  020105  042514
10451  055430  052106  044440  020116
10452  055436  047101  052440  042116
10453  055444  052105  051105  044515
10454  055452  042516  020104  052123
10455  055460  052101  006505      012
10456  055465      015  044412  044516           .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
10457  055472  044524  046101  054514
10458  055500  020054  051104  053111
10459  055506  051505  052040  020117
10460  055514  042502  052040  051505
10461  055522  042524  020104  044123
10462  055530  052517  042114  044040
10463  055536  053101  035105  005015
```

```
10464  055544  005015  027101  020040          .ASCII  <CR><LF>/A.  HEADS MANUALLY LOADED/
10465  055552  042510  042101  020123
10466  055560  040515  052516  046101
10467  055566  054514  046040  040517
10468  055574  042504     104
10469  055577     015  041012  020056          .ASCII  <CR><LF>/B.  CORRECT PORT SELECTED/
10470  055604  041440  051117  042522
10471  055612  052103  050040  051117
10472  055620  020124  042523  042514
10473  055626  052103  042105
10474  055632  005015  027103  020040          .ASCII  <CR><LF>/C.  WRITE LOCK DISABLED/
10475  055640  051127  052111  020105
10476  055646  047514  045503  042040
10477  055654  051511  041101  042514
10478  055662     104
10479  055663     015  042012  020056          .ASCII  <CR><LF>/D.  DRIVE READY INDICATOR ON/<CR><LF>
10480  055670  042040  044522  042526
10481  055676  051040  040505  054504
10482  055704  044440  042116  041511
10483  055712  052101  051117  047440
10484  055720  006516     012
10485  055723     015  042012  044522          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE BOTH/
10486  055730  042526  020123  047516
10487  055736  020124  047524  041040
10488  055744  020105  042524  052123
10489  055752  042105  046440  051525
10490  055760  020124  040510  042526
10491  055766  041040  052117     110
10492  055773     015  050012  051117          .ASCIZ  <CR><LF>/PORTS DESELECTED/<CR><LF>
10493  056000  051524  042040  051505
10494  056006  046105  041505  042524
10495  056014  006504  000012
10496
10497
10498
10499
10500  056020  005015  052520  020124  MSG2:   .ASCIZ  <CR><LF>/PUT SCRATCH PACK IN DRIVE 0/
10501  056026  041523  040522  041524
10502  056034  020110  040520  045503
10503  056042  044440  020116  051104
10504  056050  053111  020105  000060
10505  056056  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
10506  056064  024105  024523  052040
10507  056072  020117  042502  052040
10508  056100  051505  042524  035104
10509  056106  000040
10510  056110  005015  052502  020123  MSG4:   .ASCIZ  <CR><LF>/BUS ADDR (177440): /
10511  056116  042101  051104  024040
10512  056124  033461  032067  030064
10513  056132  035051  000040
10514  056136  005015  047503  052116  MSG5:   .ASCIZ  <CR><LF>/CONTR ADDR (210): /
10515  056144  020122  042101  051104
10516  056152  024040  030462  024460
10517  056160  020072     000
10518  056163     015  044412  052116  MSG6:   .ASCIZ  <CR><LF>/INTR AT PC=/
10519  056170  020122  052101  050040
```

```
10520  056176  036503     000
10521  056201     015  042012  044522  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
10522  056206  042526  030040  053440
10523  056214  046111  020114  047516
10524  056222  020124  042502  052040
10525  056230  051505  042524  000104
10526  056236  005015  042524  052123  MSG8:   .ASCIZ  <CR><LF>/TEST 16 TAKES 2 TO 4 MIN./<CR><LF>
10527  056244  030440  020066  040524
10528  056252  042513  020123  020062
10529  056260  047524  032040  046440
10530  056266  047111  006456  000012
10531  056274  005015  054502  040520  MSG9:   .ASCIZ  <CR><LF>/BYPASSING TEST 16/<CR><LF>
10532  056302  051523  047111  020107
10533  056310  042524  052123  030440
10534  056316  006466  000012
10535  056322  005015  053412  046111  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
10536  056330  020114  042524  052123
10537  056336  042040  044522  042526
10538  056344  035123     000
10539  056347     015  005012  047520  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
10540  056354  042527  020122  050125
10541  056362  051040  051505  040524
10542  056370  052122  052040  020117
10543  056376  042524  052123  030440
10544  056404  005015     000
10545  056407     015  050012  041501  MSG12:  .ASCIZ  <CR><LF>/PACK BEING FORMATTED/<CR><LF>
10546  056414  020123  042502  047111
10547  056422  020107  047506  046522
10548  056430  052101  042524  006504
10549  056436  000012
10550  056440  005015  047516  046040  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS/
10551  056446  047440  020122  020120
10552  056454  046103  041517  051513
10553  056462  005015  046101  020114          .ASCIZ  <CR><LF>/ALL TIMING TESTS BYPASSED/
10554  056470  044524  044515  043516
10555  056476  052040  051505  051524
10556  056504  041040  050131  051501
10557  056512  042523  000104
10558  056516  005015  054502  040520  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
10559  056524  051523  047111  020107
10560  056532  051104  053111  020105
10561  056540     000
10562  056541     015  005012  051104  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
10563  056546  053111  020105     000
10564  056553     015  042012  044522  MSG16:  .ASCIZ  <CR><LF>/DRIVE SER #/
10565  056560  042526  051440  051105
10566  056566  021440     000
10567  056571     015  041412  051101  MSG17:  .ASCIZ  <CR><LF>/CART SER #/
10568  056576  020124  042523  020122
10569  056604  000043
10570  056606  005015  040412  047502  MSG18:  .ASCIZ  <CR><LF><LF>/ABORTING BAL OF TESTS/<CR><LF><LF>
10571  056614  052122  047111  020107
10572  056622  040502  020114  043117
10573  056630  052040  051505  051524
10574  056636  005015  000012
10575  056642  005015  040412  046114  MSG19:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
```

```
10576  056650  042040  044522  042526
10577  056656  020123  042524  052123
10578  056664  042105  005015  000012
10579  056672  005015  047515  044504  MSG20:  .ASCII  <CR><LF>/MODIFIED VERSION OF FORMAT PACK TEST FOR MODULE TESTING/
10580  056700  044506  042105  053040
10581  056706  051105  044523  047117
10582  056714  047440  020106  047506
10583  056722  046522  052101  050040
10584  056730  041501  020113  042524
10585  056736  052123  043040  051117
10586  056744  046440  042117  046125
10587  056752  020105  042524  052123
10588  056760  047111      107
10589  056763      015  052012  020117          .ASCIZ  <CR><LF>/TO RESTORE HEADERS ON CYL 0 & 1, ALL TRACKS/
10590  056770  042522  052123  051117
10591  056776  020105  042510  042101
10592  057004  051105  020123  047117
10593  057012  041440  046131  030040
10594  057020  023040  030440  020054
10595  057026  046101  020114  051124
10596  057034  041501  051513      000
10597  057041      015  041012  050131  MSG21:  .ASCIZ  <CR><LF>/BYPASSING TESTS 36,40,41 FOR MODULE TESTING/<CR><LF>
10598  057046  051501  044523  043516
10599  057054  052040  051505  051524
10600  057062  031440  026066  030064
10601  057070  032054  020061  047506
10602  057076  020122  047515  052504
10603  057104  042514  052040  051505
10604  057112  044524, 043516  005015
10605  057120      000
10606  057121      015  043012  051117  MSG22:  .ASCIZ  <CR><LF>/FORMATTING FINISHED/<CR><LF>
10607  057126  040515  052124  047111
10608  057134  020107  044506  044516
10609  057142  044123  042105  005015
10610  057150      000
10611
10612  057151      015  050012  046507  MSG74:  .ASCIZ  <CR><LF>/PGM ABORT PENDING.../
10613  057156  040440  047502  052122
10614  057164  050040  047105  044504
10615  057172  043516  027056  000056
10616  057200  005015  040510  052114  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING.../
10617  057206  050040  047105  044504
10618  057214  043516  027056  000056
10619  057222  005015  043520  020115  MSG76:  .ASCIZ  <CR><LF>/PGM ABORTED/
10620  057230  041101  051117  042524
10621  057236  000104
10622  057240  005015  050103  020125  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
10623  057246  040510  052114  042105
10624  057254      000
10625
10626
10627                                  .SBTTL  ERR MSGS
10628
10629  057255      015  042412  051122  EM1:    .ASCIZ  <CR><LF>/ERR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
10630  057262  020054  047117  054514
10631  057270  030040  052040  051110
```

```
10632  057276  020125  020067  046101
10633  057304  047514  042527  026104
10634  057312  052040  054522  040440
10635  057320  040507  047111  005015
10636  057326     000
10637  057327     123  046105  041505  EM2:    .ASCIZ  /SELECTED DRIVE # IN RKCS2 CANNOT BE READ BACK IN RKMR2/
10638  057334  042524  020104  051104
10639  057342  053111  020105  020043
10640  057350  047111  051040  041513
10641  057356  031123  041440  047101
10642  057364  047516  020124  042502
10643  057372  051040  040505  020104
10644  057400  040502  045503  044440
10645  057406  020116  045522  051115
10646  057414  000062
10647  057416  005015  041101  051117  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXP TIME OUT AT PC=/
10648  057424  020124  042524  052123
10649  057432  027123  027056  047125
10650  057440  054105  020120  044524
10651  057446  042515  047440  052125
10652  057454  040440  020124  041520
10653  057462  000075
10654  057464  042115  020123  042523  EM5:    .ASCIZ  /MDS SET IN RKCS2/
10655  057472  020124  047111  051040
10656  057500  041513  031123     000
10657  057505     125  042506  051440  EM6:    .ASCIZ  /UFE SET IN RKCS2/
10658  057512  052105  044440  020116
10659  057520  045522  051503  000062
10660  057526  051104  020101  047111  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
10661  057534  051040  042113  020123
10662  057542  020046  042516  020104
10663  057550  047111  051040  041513
10664  057556  031123  051040  051505
10665  057564  052105  020073  051127
10666  057572  047117  020107  047520
10667  057600  052122  051440  046105
10668  057606  041505  042524  037504
10669  057614     000
10670  057615     104  044522  042526  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT TYPED BY OPERATOR/
10671  057622  050040  042522  042523
10672  057630  052116  041040  052125
10673  057636  047040  052117  052040
10674  057644  050131  042105  041040
10675  057652  020131  050117  051105
10676  057660  052101  051117     000
10677  057665     104  044522  042526  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT TYPED BY OPERATOR/
10678  057672  047040  052117  050040
10679  057700  042522  042523  052116
10680  057706  041040  052125  052040
10681  057714  050131  042105  041040
10682  057722  020131  050117  051105
10683  057730  052101  051117     000
10684  057735     101  047502  052122  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REF CONTR REG/
10685  057742  052040  051505  051524
10686  057750  027056  041456  047101
10687  057756  047516  020124  042522
```

```
10688  057764  020106  047503  052116
10689  057772  020122  042522  000107
10690  060000  051104  020101  047111  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
10691  060006  051040  042113  020123
10692  060014  020046  042516  020104
10693  060022  047111  051040  041513
10694  060030  031123  041040  052117
10695  060036  020110  042523  000124
10696  060044  047503  052116  020122  EM12:   .ASCIZ  /CONTR NOT READY IN RKCS1/
10697  060052  047516  020124  042522
10698  060060  042101  020131  047111
10699  060066  051040  041513  030523
10700  060074     000
10701  060075     116  020117  052101  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
10702  060102  047124  044440  020116
10703  060110  045522  051501  043117
10704  060116     000
10705  060117     127  047522  043516  EM14:   .ASCIZ  /WRONG ATTN IN RKASOF/
10706  060124  040440  052124  020116
10707  060132  047111  051040  040513
10708  060140  047523  000106
10709  060144  051104  054504  047040  EM15:   .ASCIZ  /DRDY NOT CLEARED IN RKMR2/
10710  060152  052117  041440  042514
10711  060160  051101  042105  044440
10712  060166  020116  045522  051115
10713  060174  000062
10714  060176  051504  020103  047516  EM16:   .ASCIZ  /DSC NOT SET IN RKMR2/
10715  060204  020124  042523  020124
10716  060212  047111  051040  046513
10717  060220  031122     000
10718  060223     115  043523  040440  EM17:   .ASCIZ  /MSG A0 ERR/
10719  060230  020060  051105  000122
10720  060236  051515  020107  030102  EM18:   .ASCIZ  /MSG B0 ERR/
10721  060244  042440  051122     000
10722  060251     115  043523  040440  EM19:   .ASCIZ  /MSG A1 ERR/
10723  060256  020061  051105  000122
10724  060264  051515  020107  030502  EM20:   .ASCIZ  /MSG B1 ERR/
10725  060272  042440  051122     000
10726  060277     103  051105  020122  EM21:   .ASCIZ  /CERR SET IN RKCS1/
10727  060304  042523  020124  047111
10728  060312  051040  041513  030523
10729  060320     000
10730  060321     122  051514  044440  EM22:   .ASCIZ  /RLS IN RKCS2 SET CERR IN RKCS1/
10731  060326  020116  045522  051503
10732  060334  020062  042523  020124
10733  060342  042503  051122  044440
10734  060350  020116  045522  051503
10735  060356  000061
10736  060360  043125  020105  047111  EM23:   .ASCIZ  /UFE IN RKCS2 SET (SACK) AFTER RLS IN RKCS2 SENT/
10737  060366  051040  041513  031123
10738  060374  051440  052105  024040
10739  060402  040523  045503  020051
10740  060410  043101  042524  020122
10741  060416  046122  020123  047111
10742  060424  051040  041513  031123
10743  060432  051440  047105  000124
```

```
10744  060440  053126  047040  052117  EM24:   .ASCIZ  /VV NOT SET IN RKMR2/
10745  060446  051440  052105  044440
10746  060454  020116  045522  051115
10747  060462  000062
10748  060464  051104  020126  054524  EM25:   .ASCIZ  /DRV TYPE SET IN RKMR2/
10749  060472  042520  051440  052105
10750  060500  044440  020116  045522
10751  060506  051115  000062
10752  060512  042104  020124  042523  EM26:   .ASCIZ  /DDT SET IN RKDS/
10753  060520  020124  047111  051040
10754  060526  042113  000123
10755  060532  052104  042531  051440  EM27:   .ASCIZ  /DTYE SET IN RKER/
10756  060540  052105  044440  020116
10757  060546  045522  051105     000
10758  060553     104  054524  020105  EM28:   .ASCIZ  /DTYE NOT SET IN RKER/
10759  060560  047516  020124  042523
10760  060566  020124  047111  051040
10761  060574  042513  000122
10762  060600  052104  042531  044440  EM29:   .ASCIZ  /DTYE IN RKER DID NOT SET CERR IN RKCS1/
10763  060606  020116  045522  051105
10764  060614  042040  042111  047040
10765  060622  052117  051440  052105
10766  060630  041440  051105  020122
10767  060636  047111  051040  041513
10768  060644  030523     000
10769  060647     103  042055  050040  EM30:   .ASCIZ  /C-D PAR ERR SET IN RKMR3/
10770  060654  051101  042440  051122
10771  060662  051440  052105  044440
10772  060670  020116  045522  051115
10773  060676  000063
10774  060700  026504  020103  040520  EM31:   .ASCIZ  /D-C PAR SET IN RKCS1/
10775  060706  020122  042523  020124
10776  060714  047111  051040  041513
10777  060722  030523     000
10778  060725     106  052114  047040  EM32:   .ASCIZ  /FLT NOT SET IN RKMR3/
10779  060732  052117  051440  052105
10780  060740  044440  020116  045522
10781  060746  051115  000063
10782  060752  026503  020104  040520  EM33:   .ASCIZ  /C-D PAR ERR NOT SET IN RKMR3/
10783  060760  020122  051105  020122
10784  060766  047516  020124  042523
10785  060774  020124  047111  051040
10786  061002  046513  031522     000
10787  061007     104  041455  050040  EM34:   .ASCIZ  /D-C PAR NOT SET IN RKCS1/
10788  061014  051101  047040  052117
10789  061022  051440  052105  044440
10790  061030  020116  045522  051503
10791  061036  000061
10792  061040  026504  020103  040520  EM35:   .ASCIZ  /D-C PAR IN RKCS1 DID NOT SET CERR IN RKCS1/
10793  061046  020122  047111  051040
10794  061054  041513  030523  042040
10795  061062  042111  047040  052117
10796  061070  051440  052105  041440
10797  061076  051105  020122  047111
10798  061104  051040  041513  030523
10799  061112     000
```

```
10800  061113  103     046131  040440  EM36:   .ASCIZ  /CYL ADDR IN B2 NOT SAME AS RKDC/
10801  061120  042104  020122  047111
10802  061126  041040  020062  047516
10803  061134  020124  040523  042515
10804  061142  040440  020123  045522
10805  061150  041504  000
10806  061153  103     046131  042040  EM37:   .ASCIZ  /CYL DIFF IN A2 NOT SAME AS RKDC/
10807  061160  043111  020106  047111
10808  061166  040440  020062  047516
10809  061174  020124  040523  042515
10810  061202  040440  020123  045522
10811  061210  041504  000
10812  061213  103     046131  042040  EM38:   .ASCIZ  /CYL DIFF IN RKMR2 NOT SAME AS 'CYL DIFF'/
10813  061220  043111  020106  047111
10814  061226  051040  046513  031122
10815  061234  047040  052117  051440
10816  061242  046501  020105  051501
10817  061250  023440  054503  020114
10818  061256  044504  043106  000047
10819  061264  054503  020114  044504  EM39:   .ASCIZ  /CYL DIFF & OFST IN A2 NOT =0/
10820  061272  043106  023040  047440
10821  061300  051506  020124  047111
10822  061306  040440  020062  047516
10823  061314  020124  030075  000
10824  061321  103     046131  040440  EM40:   .ASCIZ  /CYL ADDR IN B2 NOT =0/
10825  061326  042104  020122  047111
10826  061334  041040  020062  047516
10827  061342  020124  030075  000
10828  061347  103     046131  040440  EM41:   .ASCIZ  /CYL ADDR IN B2 DID NOT REMAIN =0/
10829  061354  042104  020122  047111
10830  061362  041040  020062  044504
10831  061370  020104  047516  020124
10832  061376  042522  040515  047111
10833  061404  036440  000060
10834  061410  042510  042101  040440  EM43:   .ASCIZ  /HEAD ADDR IN B3 NOT =0/
10835  061416  042104  020122  047111
10836  061424  041040  020063  047516
10837  061432  020124  030075  000
10838  061437  110     040505  020104  EM44:   .ASCIZ  /HEAD DECODE IN B3 INCORRECT/
10839  061444  042504  047503  042504
10840  061452  044440  020116  031502
10841  061460  044440  041516  051117
10842  061466  042522  052103  000
10843  061473  104     044522  042526  EM45:   .ASCII  /DRIVE READY IN RKMR2 NOT SET BY 1 SEC FROM FWD/
10844  061500  051040  040505  054504
10845  061506  044440  020116  045522
10846  061514  051115  020062  047516
10847  061522  020124  042523  020124
10848  061530  054502  030440  051440
10849  061536  041505  043040  047522
10850  061544  020115  053506  104
10851  061551  015     044412  020116          .ASCIZ  <CR><LF>/IN RTZ PORTION OF START SPIN CMD/
10852  061556  052122  020132  047520
10853  061564  052122  047511  020116
10854  061572  043117  051440  040524
10855  061600  052122  051440  044520
```

```
10856  061606  020116  046503  000104
10857  061614  051515  020107  031101  EM46:   .ASCIZ  /MSG A2 ERR/
10858  061622  042440  051122     000
10859  061627     115  043523  041040  EM47:   .ASCIZ  /MSG B2 ERR/
10860  061634  020062  051105  000122
10861  061642  051515  020107  031502  EM48:   .ASCIZ  /MSG B3 ERR/
10862  061650  042440  051122     000
10863  061655     106  042127  047040  EM49:   .ASCIZ  /FWD NOT SET IN RKMR2 IN RTZ PORTION OF START SPIN CMD/
10864  061662  052117  051440  052105
10865  061670  044440  020116  045522
10866  061676  051115  020062  047111
10867  061704  051040  055124  050040
10868  061712  051117  044524  047117
10869  061720  047440  020106  052123
10870  061726  051101  020124  050123
10871  061734  047111  041440  042115
10872  061742     000
10873  061743     106  042127  047040  EM50:   .ASCIZ  /FWD NOT SET IN RKMR2 FROM START SPIN CMD/
10874  061750  052117  051440  052105
10875  061756  044440  020116  045522
10876  061764  051115  020062  051106
10877  061772  046517  051440  040524
10878  062000  052122  051440  044520
10879  062006  020116  046503  000104
10880  062014  053506  020104  047516  EM51:   .ASCIZ  /FWD NOT CLEARED IN RKMR2 BY 5 SEC OF MOTION FROM START SPIN CMD/
10881  062022  020124  046103  040505
10882  062030  042522  020104  047111
10883  062036  051040  046513  031122
10884  062044  041040  020131  020065
10885  062052  042523  020103  043117
10886  062060  046440  052117  047511
10887  062066  020116  051106  046517
10888  062074  051440  040524  052122
10889  062102  051440  044520  020116
10890  062110  046503  000104
10891  062114  030062  051440  041505  EM52:   .ASCIZ  /20 SEC FORMAT NOT SET IN RKMR2/
10892  062122  043040  051117  040515
10893  062130  020124  047516  020124
10894  062136  042523  020124  047111
10895  062144  051040  046513  031122
10896  062152     000
10897  062153     123  041505  030040  EM53:   .ASCIZ  /SEC 0 NOT FOUND BY 50 MS/
10898  062160  047040  052117  043040
10899  062166  052517  042116  041040
10900  062174  020131  030065  046440
10901  062202  000123
10902  062204  044504  043106  051440  EM54:   .ASCIZ  /DIFF SEC NOT FOUND BY 3 MS/
10903  062212  041505  047040  052117
10904  062220  043040  052517  042116
10905  062226  041040  020131  020063
10906  062234  051515     000
10907  062237     101  052124  020116  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
10908  062244  047516  020124  046103
10909  062252  040505  042522  020104
10910  062260  047111  051040  040513
10911  062266  047523  000106
```

```
10912  062272  047125  054105  020120  EM56:   .ASCIZ  /UNEXP MEM PAR TRAP/
10913  062300  042515  020115  040520
10914  062306  020122  051124  050101
10915  062314     000
10916  062315     127  042503  040040  EM57:   .ASCIZ  /WCE @ CYL 411, TRK 2, SEC 2/
10917  062322  041440  046131  032040
10918  062330  030461  020054  051124
10919  062336  020113  026062  051440
10920  062344  041505  031040     000
10921  062351     015  051412  042520  EM58:   .ASCIZ  <CR><LF>/SPEED OK IN RKMR2 NOT =0 BY TIMEOUT/
10922  062356  042105  047440  020113
10923  062364  047111  051040  046513
10924  062372  031122  047040  052117
10925  062400  036440  020060  054502
10926  062406  052040  046511  047505
10927  062414  052125     000
10928  062417     114  046511  042040  EM59:   .ASCIZ  /LIM DET NOT SET IN RKMR3/
10929  062424  052105  047040  052117
10930  062432  051440  052105  044440
10931  062440  020116  045522  051115
10932  062446  000063
10933  062450  042510  042101  020123  EM60:   .ASCIZ  /HEADS HOME NOT SET IN RKMR2/
10934  062456  047510  042515  047040
10935  062464  052117  051440  052105
10936  062472  044440  020116  045522
10937  062500  051115  000062
10938  062504  047514  042101  044040  EM61:   .ASCIZ  /LOAD HEADS NOT SET IN RKMR2/
10939  062512  040505  051504  047040
10940  062520  052117  051440  052105
10941  062526  044440  020116  045522
10942  062534  051115  000062
10943  062540  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
10944  062546  020124  047111  051040
10945  062554  041513  031123     000
10946  062561     115  043523  041040  EM64:   .ASCIZ  /MSG B3 HEAD REG NOT =0/
10947  062566  020063  042510  042101
10948  062574  051040  043505  047040
10949  062602  052117  036440  000060
10950  062610  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERR/
10951  062616  040505  042504  020122
10952  062624  051105  000122
10953  062630  054503  020114  042101  EM66:   .ASCIZ  /CYL ADDR IN RKMR3 INCORRECT/
10954  062636  051104  044440  020116
10955  062644  045522  051115  020063
10956  062652  047111  047503  051122
10957  062660  041505  000124
10958  062664  042522  042101  047111  EM67:   .ASCIZ  /READING CYL 0 HEADERS ON CYL 1/
10959  062672  020107  054503  020114
10960  062700  020060  042510  042101
10961  062706  051105  020123  047117
10962  062714  041440  046131  030440
10963  062722     000
10964  062723     122  040505  044504  EM68:   .ASCIZ  /READING CYL 1 HEADERS ON CYL 0/
10965  062730  043516  041440  046131
10966  062736  030440  044040  040505
10967  062744  042504  051522  047440
```

```
10968   062752  020116  054503  020114
10969   062760  000060
10970   062762  046101  043511  020116  EM69:   .ASCIZ  /ALIGN CART USED/
10971   062770  040503  052122  052440
10972   062776  042523  000104
10973   063002  047125  054105  020120  EM70:   .ASCIZ  /UNEXP ATTN/
10974   063010  052101  047124     000
10975   063015     104  041523  051440  EM71:   .ASCIZ  /DSC SET IN RKMR2/
10976   063022  052105  044440  020116
10977   063030  045522  051115  000062
10978   063036  047506  046522  052101  EM72:   .ASCIZ  /FORMAT TEST BYPASSED/
10979   063044  052040  051505  020124
10980   063052  054502  040520  051523
10981   063060  042105     000
10982   063063     103  047524  051440  EM73:   .ASCIZ  /CTO SET IN RKCS1/
10983   063070  052105  044440  020116
10984   063076  045522  051503  000061
10985   063104  052122  020132  047516  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
10986   063112  020124  042523  020124
10987   063120  047111  051040  046513
10988   063126  031122     000
10989   063131     111  040504  020105  EM75:   .ASCIZ  /IDAE NOT SET IN RKMR3/
10990   063136  047516  020124  042523
10991   063144  020124  047111  051040
10992   063152  046513  031522     000
10993   063157     120  050111  051440  EM76:   .ASCIZ  /PIP SET IN RKMR2/
10994   063164  052105  044440  020116
10995   063172  045522  051115  000062
10996   063200  040506  046125  020124  EM77:   .ASCIZ  /FAULT NOT =0 IN RKMR3/
10997   063206  047516  020124  030075
10998   063214  044440  020116  045522
10999   063222  051115  000063
11000   063226  054503  020114  044504  EM78:   .ASCIZ  /CYL DIFF IN RKMR2 DID NOT REMAIN = 1 IN SEEK TO SELF/
11001   063234  043106  044440  020116
11002   063242  045522  051115  020062
11003   063250  044504  020104  047516
11004   063256  020124  042522  040515
11005   063264  047111  036440  030440
11006   063272  044440  020116  042523
11007   063300  045505  052040  020117
11008   063306  042523  043114     000
11009   063313     116  042105  051440  EM79:   .ASCIZ  /NED SET IN RKCS2/
11010   063320  052105  044440  020116
11011   063326  045522  051503  000062
11012   063334  047125  047514  042101  EM80:   .ASCIZ  /UNLOAD NOT SET IN RKMR2/
11013   063342  047040  052117  051440
11014   063350  052105  044440  020116
11015   063356  045522  051115  000062
11016   063364  050123  047111  047040  EM81:   .ASCIZ  /SPIN NOT SET IN RKMR2/
11017   063372  052117  051440  052105
11018   063400  044440  020116  045522
11019   063406  051115  000062
11020   063412  052122  020132  047516  EM82:   .ASCIZ  /RTZ NOT SET IN RKMR2/
11021   063420  020124  042523  020124
11022   063426  047111  051040  046513
11023   063434  031122     000
```

```
11024  063437     122  040505  020104  EM83:   .ASCIZ  /READ HEADER ERR   WORD 0 (CYL#)/
11025  063444  042510  042101  051105
11026  063452  042440  051122  020040
11027  063460  053440  051117  020104
11028  063466  020060  041450  046131
11029  063474  024443     000
11030  063477     106  051117  040515  EM84:   .ASCIZ  /FORMAT IN RKMR3 NOT SET/
11031  063504  020124  047111  051040
11032  063512  046513  031522  047040
11033  063520  052117  051440  052105
11034  063526     000
11035  063527     111  046114  040440  EM85:   .ASCIZ  /ILL ADDR IN RKMR3 NOT =0/
11036  063534  042104  020122  047111
11037  063542  051040  046513  031522
11038  063550  047040  052117  036440
11039  063556  000060
11040  063560  044127  046111  020105  EM86:   .ASCIZ  /WHILE WAITING FOR CONTR RDY OR AFTER CONTR RDY REC'D/
11041  063566  040527  052111  047111
11042  063574  020107  047506  020122
11043  063602  047503  052116  020122
11044  063610  042122  020131  051117
11045  063616  040440  052106  051105
11046  063624  041440  047117  051124
11047  063632  051040  054504  051040
11048  063640  041505  042047     000
11049  063645     103  047101  047516  EM87:   .ASCIZ  /CANNOT READ BSE INFO/
11050  063652  020124  042522  042101
11051  063660  041040  042523  044440
11052  063666  043116  000117
11053  063672  047516  042040  044522  EM88:   .ASCII  /NO DRIVES FOUND ON BUS/<CR><LF>
11054  063700  042526  020123  047506
11055  063706  047125  020104  047117
11056  063714  041040  051525  005015
11057  063722  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY & PRESS 'CONT'/<CR><LF>
11058  063730  047503  051122  041505
11059  063736  046124  020131  020046
11060  063744  051120  051505  020123
11061  063752  041447  047117  023524
11062  063760  005015     000
11063  063763     116  020117  051104  EM89:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/<CR><LF>
11064  063770  053111  051505  043040
11065  063776  052517  042116  044440
11066  064004  020116  042504  044526
11067  064012  042503  046440  050101
11068  064020  024040  042044  053105
11069  064026  024515  005015
11070  064032  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY & RESTART/<CR><LF>
11071  064040  047503  051122  041505
11072  064046  046124  020131  020046
11073  064054  042522  052123  051101
11074  064062  006524  000012
11075
11076                                  .SBTTL  DATA HEADERS
11077
11078  064066  042524  052123  047040  DH1:    .ASCIZ  /TEST NO.  PC/
11079  064074  027117  020040  041520
```

```
11080  064102     000
11081  064103     122  046513  031122  DH2:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKCS1   RKCS2   RKASOF/
11082  064110  051011  046513  031522
11083  064116  051011  042513  004522
11084  064124  045522  051504  051011
11085  064132  041513  030523  051011
11086  064140  041513  031123  051011
11087  064146  040513  047523  000106
11088  064154  045522  051115  004462  DH5:    .ASCIZ  /RKMR2  RKMP3   RKER    RKDS    RKDC    RKCS1   RKCS2/
11089  064162  045522  051115  004463
11090  064170  045522  051105  051011
11091  064176  042113  004523  045522
11092  064204  041504  051011  041513
11093  064212  030523  051011  041513
11094  064220  031123     000
11095  064223     122  046513  031122  DH6:    .ASCIZ  /RKMR2   RKMR3   RKDC   FROM CYL  TO CYL  CYL DIFF/
11096  064230  020040  051040  046513
11097  064236  031522  020040  051040
11098  064244  042113  020103  020040
11099  064252  051106  046517  041440
11100  064260  046131  020040  047524
11101  064266  041440  046131  020040
11102  064274  054503  020114  044504
11103  064302  043106     000
11104  064305     122  046513  031122  DH7:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDA    RKCS1   RKCS2/
11105  064312  051011  046513  031522
11106  064320  051011  042513  004522
11107  064326  045522  051504  051011
11108  064334  042113  004501  045522
11109  064342  051503  004461  045522
11110  064350  051503  000062
11111  064354  044127  047105  042040  DH8:    .ASCIZ  /WHEN DRIVE UNLOADED/
11112  064362  044522  042526  052440
11113  064370  046116  040517  042504
11114  064376  000104
11115  064400  043101  042524  020122  DH9:    .ASCIZ  /AFTER START SPIN CMD REC'D BY DRIVE/
11116  064406  052123  051101  020124
11117  064414  050123  047111  041440
11118  064422  042115  051040  041505
11119  064430  042047  041040  020131
11120  064436  051104  053111  000105
11121  064444  052101  042440  042116  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
11122  064452  047440  020106  042510
11123  064460  042101  046040  040517
11124  064466  044504  043516     000
11125  064473     101  052106  051105  DH11:   .ASCIZ  /AFTER START SPIN CMD & FWD SET/
11126  064500  051440  040524  052122
11127  064506  051440  044520  020116
11128  064514  046503  020104  020046
11129  064522  053506  020104  042523
11130  064530  000124
11131  064532  052101  044440  047116  DH12:   .ASCIZ  /AT INNER LIM FROM START SPIN CMD/
11132  064540  051105  046040  046511
11133  064546  043040  047522  020115
11134  064554  052123  051101  020124
11135  064562  050123  047111  041440
```

```
11136  064570  042115     000
11137  064573     106  047522  020115  DH13:   .ASCIZ  /FROM OUTER LIM TO CYL 0 DURING LOADING/
11138  064600  052517  042524  020122
11139  064606  044514  020115  047524
11140  064614  041440  046131  030040
11141  064622  042040  051125  047111
11142  064630  020107  047514  042101
11143  064636  047111  000107
11144  064642  043101  042524  020122  DH14:   .ASCIZ  /AFTER SEEK WITH BAD PAR/
11145  064650  042523  045505  053440
11146  064656  052111  020110  040502
11147  064664  020104  040520  000122
11148  064672  043101  042524  020122  DH16:   .ASCIZ  /AFTER LOADING HEAD REG & SEEK CMD/
11149  064700  047514  042101  047111
11150  064706  020107  042510  042101
11151  064714  051040  043505  023040
11152  064722  051440  042505  020113
11153  064730  046503  000104
11154  064734  043101  042524  020122  DH17:   .ASCIZ  /AFTER RECAL CMD/
11155  064742  042522  040503  020114
11156  064750  046503  000104
11157  064754  043101  042524  020122  DH18:   .ASCIZ  /AFTER UNLOAD CMD/
11158  064762  047125  047514  042101
11159  064770  041440  042115     000
11160  064775     101  052106  051105  DH19:   .ASCIZ  /AFTER PACK CMD/
11161  065002  050040  041501  020113
11162  065010  046503  000104
11163  065014  043101  042524  020122  DH20:   .ASCIZ  /AFTER SELECT DRIVE CMD/
11164  065022  042523  042514  052103
11165  065030  042040  044522  042526
11166  065036  041440  042115     000
11167  065043     101  052106  051105  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
11168  065050  051440  041125  054523
11169  065056  052123  046505  041440
11170  065064  042514  051101     000
11171  065071     101  052106  051105  DH22:   .ASCIZ  /AFTER DRIVE CLEAR CMD/
11172  065076  042040  044522  042526
11173  065104  041440  042514  051101
11174  065112  041440  042115     000
11175  065117     124  051505  020124  DH23:   .ASCIZ  /TEST NO.       TRAP PC/
11176  065124  047516  004456  051124
11177  065132  050101  050040  000103
11178  065140  043101  042524  020122  DH25:   .ASCIZ  /AFTER SEEK CMD/
11179  065146  042523  045505  041440
11180  065154  042115     000
11181  065157     101  052106  051105  DH26:   .ASCIZ  /AFTER READ DATA CMD/
11182  065164  051040  040505  020104
11183  065172  040504  040524  041440
11184  065200  042115     000
11185  065203     105  050130  041505  DH28:   .ASCIZ  /EXPECT EXPECT  EXPECT  EXPECT  EXPECT  EXPECT  EXPECT/
11186  065210  004524  054105  042520
11187  065216  052103  042411  050130
11188  065224  041505  004524  054105
11189  065232  042520  052103  042411
11190  065240  050130  041505  004524
11191  065246  054105  042520  052103
```

```
11192  065254  042411  050130  041505
11193  065262  000124
11194  065264  041501  052524  046101  DH29:   .ASCIZ  /ACTUAL ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL/
11195  065272  040411  052103  040525
11196  065300  004514  041501  052524
11197  065306  046101  040411  052103
11198  065314  040525  004514  041501
11199  065322  052524  046101  040411
11200  065330  052103  040525  004514
11201  065336  041501  052524  046101
11202  065344     000
11203  065345     101  052106  051105  DH30:   .ASCIZ  /AFTER READ HEADER CMD/
11204  065352  051040  040505  020104
11205  065360  042510  042101  051105
11206  065366  041440  042115     000
11207  065373     122  046513  031122  DH31:   .ASCIZ  /RKMR2  RKMR3   RKCS1   RKCS2   RKDC    RKDA/
11208  065400  051011  046513  031522
11209  065406  051011  041513  030523
11210  065414  051011  041513  031123
11211  065422  051011  042113  004503
11212  065430  045522  040504     000
11213  065435     101  004460  030102  DH32:   .ASCIZ  /A0      B0      A1      B1      A2      B2      B3/
11214  065442  040411  004461  030502
11215  065450  040411  004462  031102
11216  065456  041011  000063
11217  065462  052504  044522  043516  DH33:   .ASCIZ  /DURING SEEK CMD/
11218  065470  051440  042505  020113
11219  065476  046503  000104
11220  065502  042523  052103  051117  DH34:   .ASCIZ  /SECTOR REG UNSTABLE/
11221  065510  051040  043505  052440
11222  065516  051516  040524  046102
11223  065524  000105
11224  065526  042502  053524  042505  DH35:   .ASCIZ  /BETWEEN SECTOR COUNTS/
11225  065534  020116  042523  052103
11226  065542  051117  041440  052517
11227  065550  052116  000123
11228  065554  045522  051115  020062  DH36:   .ASCIZ  /RKMR2   RKMR3   RKCS1   RKCS2  FROM SECT  TO SECT/
11229  065562  020040  045522  051115
11230  065570  020063  020040  045522
11231  065576  051503  020061  020040
11232  065604  045522  051503  020062
11233  065612  043040  047522  020115
11234  065620  042523  052103  020040
11235  065626  047524  051440  041505
11236  065634  000124
11237  065636  050123  042505  020104  DH37:   .ASCIZ  /SPEED OK NOT =0 10 SEC AFTER UNLOAD/
11238  065644  045517  047040  052117
11239  065652  036440  020060  030061
11240  065660  051440  041505  040440
11241  065666  052106  051105  052440
11242  065674  046116  040517  000104
11243  065702  043101  042524  020122  DH38:   .ASCIZ  /AFTER LIM DET/
11244  065710  044514  020115  042504
11245  065716  000124
11246  065720  043101  042524  020122  DH39:   .ASCIZ  /AFTER WRITE HEADER CMD/
11247  065726  051127  052111  020105
```

```
11248  065734  042510  042101  051105
11249  065742  041440  042115     000
11250  065747     122  046513  031122  DH40:   .ASCIZ  /RKMR2  RKMR3   RKDA    WORD#   HEADER WAS  SHOULD BE/
11251  065754  051011  046513  031522
11252  065762  051011  042113  004501
11253  065770  047527  042122  004443
11254  065776  042510  042101  051105
11255  066004  053440  051501  020040
11256  066012  044123  052517  042114
11257  066020  041040  000105
11258  066024  052504  044522  043516  DH41:   .ASCIZ  /DURING RECAL CMD/
11259  066032  051040  041505  046101
11260  066040  041440  042115     000
11261  066045     117  020116  042523  DH42:   .ASCIZ  /ON SEC 0,2,4,6,8  CYL 410 TRK 2/
11262  066052  020103  026060  026062
11263  066060  026064  026066  020070
11264  066066  041440  046131  032040
11265  066074  030061  052040  045522
11266  066102  031040     000
11267  066105     106  051117  040515  DH44:   .ASCIZ  /FORMAT & ALL R-W TESTS WILL BE BYPASSED/
11268  066112  020124  020046  046101
11269  066120  020114  026522  020127
11270  066126  042524  052123  020123
11271  066134  044527  046114  041040
11272  066142  020105  054502  040520
11273  066150  051523  042105     000
11274  066155     102  041505  052501  DH45:   .ASCIZ  /BECAUSE OF LIM DET ERR ON PREV TEST/
11275  066162  042523  047440  020106
11276  066170  044514  020115  042504
11277  066176  020124  051105  020122
11278  066204  047117  050040  042522
11279  066212  020126  042524  052123
11280  066220     000
11281  066221     103  052517  042114  DH46:   .ASCIZ  /COULD NOT READ BSE INFO ON PREV TEST/
11282  066226  047040  052117  051040
11283  066234  040505  020104  051502
11284  066242  020105  047111  047506
11285  066250  047440  020116  051120
11286  066256  053105  052040  051505
11287  066264  000124
11288  066266  045522  051503  004461  DH47:   .ASCIZ  /RKCS1  RKCS2   RKASOF  RKER    RKDS    RKDC/
11289  066274  045522  051503  004462
11290  066302  045522  051501  043117
11291  066310  051011  042513  004522
11292  066316  045522  051504  051011
11293  066324  042113  000103
11294  066330  043101  042524  020122  DH48:   .ASCIZ  /AFTER SEEK CMD TO INV CYL/
11295  066336  042523  045505  041440
11296  066344  042115  052040  020117
11297  066352  047111  020126  054503
11298  066360  000114
11299  066362  051515  020107  023101  DH49:   .ASCIZ  /MSG A&B IN RKMR2 & RKMR3 RESP., ARE INVALID/
11300  066370  020102  047111  051040
11301  066376  046513  031122  023040
11302  066404  051040  046513  031522
11303  066412  051040  051505  027120
```

```
11304  066420  020054  051101  020105
11305  066426  047111  040526  044514
11306  066434  000104
11307  066436  043101  042524  020122  DH51:   .ASCIZ  /AFTER SEEK TO SELF CMD/
11308  066444  042523  045505  052040
11309  066452  020117  042523  043114
11310  066460  041440  042115     000
11311  066465     122  046513  031122  DH52:   .ASCIZ  /RKMR2  RKMR3   RKDA    EXP CYL#        CYL HEADER WAS/
11312  066472  051011  046513  031522
11313  066500  051011  042113  004501
11314  066506  054105  020120  054503
11315  066514  021514  041411  046131
11316  066522  044040  040505  042504
11317  066530  020122  040527  000123
11318  066536  047117  051440  041505  DH53:   .ASCIZ  /ON SEC 10,12,14,16,18,20 CYL 410 TRK 2/
11319  066544  030440  026060  031061
11320  066552  030454  026064  033061
11321  066560  030454  026070  030062
11322  066566  041440  046131  032040
11323  066574  030061  052040  045522
11324  066602  031040     000
11325                          .SBTTL  ERR OUTPUT DATA
11326
11327          066606          .EVEN
11328  066606  001214  001116  003362  DT1:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HCS1,HCS2,HASOF
11329  066614  003364  003350  003346
11330  066622  003334  003336  003352
11331  066630  001214  001116  003362  DT3:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HDC,HCS1,HCS2
11332  066636  003364  003350  003346
11333  066644  003354  003334  003336
11334  066652  001214  001116  003362  DT4:    $TESTN,$ERRPC,HMR2,HMR3,HDC,FRCYL,TOCYL,CALDIF
11335  066660  003364  003354  001350
11336  066666  001352  001360
11337  066672  001214  001116  003362  DT5:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HDA,HCS1,HCS2
11338  066700  003364  003350  003346
11339  066706  003344  003334  003336
11340  066714  001214  001116  003362  DT6:    $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,PSEC,ESEC
11341  066722  003364  003334  003336
11342  066730  001402  001404
11343  066734  001214  001116  003362  DT7:    $TESTN,$ERRPC,HMR2,HMR3,HDA,WDCNT,HDWD,TEMP1
11344  066742  003364  003344  001442
11345  066750  001454  003372
11346  066754  001214  001116  003362  DT8:    $TESTN,$ERRPC,HMR2,HMR3,HDC,TOCYL,FRCYL,CALDIF
11347  066762  003364  003354  001352
11348  066770  001350  001360
11349  066774  001214  001116  003362  DT9:    $TESTN,$ERRPC,HMR2,HMR3,HDA,TOCYL,RHTAB
11350  067002  003364  003344  001352
11351  067010  001674
11352  067012  001214  001116  003362  DT10:   $TESTN,$ERRPC,HMR2,HMR3,HDA,FRCYL,RHTAB
11353  067020  003364  003344  001350
11354  067026  001674
11355  067030  001214  001334          DT11:   $TESTN,TRAPPC
11356  067034  001214  001116  003362  DT12:   $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,HDC,HDA
11357  067042  003364  003334  003336
11358  067050  003354  003344
11359  067054  001214  001116  003424  DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
```

```
11360  067062  003426  003430  003432
11361  067070  003404  003406  003410
11362  067076  003412
11363  067100  003334  003336  003352          HCS1,HCS2,HASOF,HER,HDS,HDC
11364  067106  003350  003346  003354
11365
11366  067114  001214  001116  003424  DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
11367  067122  003426  003430  003432
11368  067130  003434  003436
11369  067134  003404  003406  003410          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
11370  067142  003412  003414  003416
11371  067150  003334  003336  003352          HCS1,HCS2,HASOF,HER,HDS,HDC
11372  067156  003350  003346  003354
11373
11374  067164  001214  001116  003424  DT15:   $TESTN,$ERRPC,E.A0,E B0,E.A1,E.B1,E.A2,E.B2,E.B3
11375  067172  003426  003430  003432
11376  067200  003434  003436  003442
11377  067206  003404  003406  003410          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
11378  067214  003412  003414  003416
11379  067222  003422
11380  067224  003334  003336  003352          HCS1,HCS2,HASOF,HER,HDS,HDC
11381  067232  003350  003346  003354
11382
11383                                  .SBTTL  ERR DATA FORMATS
11384
11385  067240  000002                  DF1:    2
11386  067242     002        000               .BYTE 2,0
11387  067244  064103                          DH2
11388  067246     007        000               .BYTE 7,0
11389
11390  067250  000004                  DF2:    4
11391  067252     000        000               .BYTE 0,0
11392  067254  066362                          DH49
11393  067256     000        000               .BYTE 0,0
11394  067260  064066                          DH1
11395  067262     002        000               .BYTE 2,0
11396  067264  064103                          DH2
11397  067266     007        000               .BYTE 7,0
11398
11399  067270  000001                  DF3:    1
11400  067272     002        000               .BYTE   2,0
11401  067274  000002                  DF4:    2
11402  067276     002        000               .BYTE   2,0
11403  067300  065373                          DH31
11404  067302     006        000               .BYTE   6,0
11405
11406  067304  000003                  DF5:    3
11407  067306     000        000               .BYTE 0,0
11408  067310  064066                          DH1
11409  067312     002        000               .BYTE 2,0
11410  067314  064154                          DH5
11411  067316     007        000               .BYTE 7,0
11412
11413  067320  000003                  DF6:    3
11414  067322     000        000               .BYTE 0,0
11415  067324  064066                          DH1
```

```
11416  067326  002  000                .BYTE 2,0
11417  067330  064223                  DH6
11418  067332  006  000                .BYTE 6,0
11419
11420
11421
11422  067334  000003                  DF10:  3
11423  067336  000  000                .BYTE 0,0
11424  067340  064066                  DH1
11425  067342  002  000                .BYTE 2,0
11426  067344  064103                  DH2
11427  067346  007  000                .BYTE 7,0
11428
11429  067350  000003                  DF12:  3
11430  067352  000  000                .BYTE 0,0
11431  067354  064066                  DH1
11432  067356  002  000                .BYTE 2,0
11433  067360  065554                  DH36
11434  067362  006  000                .BYTE 6,0
11435  067364  000003                  DF13:  3
11436  067366  000  000                .BYTE  0,0
11437  067370  064066                  DH1
11438  067372  002  000                .BYTE  2,0
11439  067374  064305                  DH7
11440  067376  007  000                .BYTE  7,0
11441
11442  067400  000002                  DF14:  2
11443  067402  002  000                .BYTE  2,0
11444  067404  065747                  DH40
11445  067406  006  000                .BYTE  6,0
11446
11447
11448  067410  000003                  DF15:  3
11449  067412  000  000                .BYTE  0,0
11450  067414  064066                  DH1
11451  067416  002  000                .BYTE  2,0
11452  067420  064305                  DH7
11453  067422  007  000                .BYTE  7,0
11454
11455  067424  000002                  DF16:  2
11456  067426  000  000                .BYTE 0,0
11457  067430  064066                  DH1
11458  067432  002  000                .BYTE 2,0
11459
11460  067434  000004                  DF17:  4
11461  067436  000  000                .BYTE  0,0
11462  067440  066105                  DH44
11463  067442  000  000                .BYTE  0,0
11464  067444  064066                  DH1
11465  067446  002  000                .BYTE  2,0
11466  067450  064103                  DH2
11467  067452  007  000                .BYTE  7,0
11468  067454  000007                  DF20:  7
11469  067456  000  000                .BYTE  0,0
11470  067460  064066                  DH1
11471  067462  002  000                .BYTE  2,0
```

```
11472  067464  065203                          DH28
11473  067466     000     000                  .BYTE   0,0
11474  067470  065435                          DH32
11475  067472     004     000                  .BYTE   4,0
11476  067474  065264                          DH29
11477  067476     000     000                  .BYTE   0,0
11478  067500  065435                          DH32
11479  067502     004     000                  .BYTE   4,0
11480  067504  066266                          DH47
11481  067506     006     000                  .BYTE   6,0
11482  067510  000002                  DF21:   2
11483  067512     002     000                  .BYTE   2,0
11484  067514  066465                          DH52
11485  067516     005     000                  .BYTE   5,0
11486  067520  000007                  DF22:   7
11487  067522     000     000                  .BYTE   0,0
11488  067524  064066                          DH1
11489  067526     002     000                  .BYTE   2,0
11490  067530  065203                          DH28
11491  067532     000     000                  .BYTE   0,0
11492  067534  065435                          DH32
11493  067536     006     000                  .BYTE   6,0
11494  067540  065264                          DH29
11495  067542     000     000                  .BYTE   0,0
11496  067544  065435                          DH32
11497  067546     006     000                  .BYTE   6,0
11498  067550  066266                          DH47
11499  067552     006     000                  .BYTE   6,0
11500
11501  067554  000007                  DF23:   7
11502  067556     000     000                  .BYTE   0,0
11503  067560  064066                          DH1
11504  067562     002     000                  .BYTE   2,0
11505  067564  065203                          DH28
11506  067566     000     000                  .BYTE   0,0
11507  067570  065435                          DH32
11508  067572     007     000                  .BYTE   7,0
11509  067574  065264                          DH29
11510  067576     000     000                  .BYTE   0,0
11511  067600  065435                          DH32
11512  067602     007     000                  .BYTE   7,0
11513  067604  066266                          DH47
11514  067606     006     000                  .BYTE   6,0
```

```
11515                                   ;;*****************************************************************
11516                                   .SBTTL  TYPE ERR ROUTINE
11517                                   ;*ENTRY JSR PC,TYP ERR
11518                                   ;*RETURN RTS PC
11519                                   ;*
11520                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
11521                                   ;*ERR IS TO BE REPORTED. IT THEN USES THE "ERR TABLE" ($ERRTB)
11522                                   ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
11523                                   ;*THE ERR.
11524                                   ;;*****************************************************************
11525   067610  104413                  TYPERR: SAVREG
11526   067612  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERR NUMBER
11527   067616  042700  177400                  BIC     #177400,R0      ;CLEAR SIGN EXTENSION
11528   067622  005300                          DEC     R0              ;FORM INDEX FOR ERR TABLE
11529   067624  006300                          ASL     R0
11530   067626  006300                          ASL     R0
11531   067630  006300                          ASL     R0
11532   067632  062700  003506          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERR ENTRY
11533   067636  012037  067652                  MOV     (R0)+,2$        ;GET EM POINTER
11534   067642  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
11535   067644  104401  001205                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
11536   067650  104401                          TYPE                    ;TYPE ERR MSG (EM)
11537   067652  000000          2$:     .WORD   0                       ;EM POINTER GOES HERE
11538   067654  012037  067670  3$:     MOV     (R0)+,4$        ;GET DH POINTER
11539   067660  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
11540   067662  104401  001205                  TYPE    ,$CRLF          ;TYPE CR-LF
11541   067666  104401                          TYPE                    ;TYPE DATA HEADER
11542   067670  000000          4$:     .WORD   0                       ;DH POINTER GOES HERE
11543   067672  012001          5$:     MOV     (R0)+,R1        ;GET DT POINTER
11544   067674  001455                  BEQ     20$             ;BRANCH IF THERE ARE NONE
11545   067676  005004                  CLR     R4              ;SET INDENT SWITCH
11546   067700  012000                  MOV     (R0)+,R0        ;GET DF POINTER
11547   067702  012002                  MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
11548   067704  001446                  BEQ     17$             ;DH NUM IS 0-BRANCH
11549   067706  005104                  COM     R4              ;NO INDENT
```

```
11550  067710  104401  001205                TYPE    ,$CRLF
11551  067714  112003                10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
11552  067716  105720                        TSTB    (R0)+           ;BUMP PAST FORMAT WORD
11553  067720  005703                        TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
11554  067722  001407                        BEQ     14$             ;NO - SKIP DATA PRINT
11555  067724  013146                11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
11556  067726  104402                        TYPOC                   ;TYPE IT
11557  067730  005303                        DEC     R3              ;MORE DATA WORDS
11558  067732  001403                        BEQ     14$             ;NO-BRANCH
11559  067734  104401  070064                TYPE    ,SPACE2         ;TYPE SEPARATORS
11560  067740  000771                        BR      11$             ;LOOP
11561  067742  005302                14$:    DEC     R2              ;MORE DH'S?
11562  067744  003431                        BLE     20$             ;NO-BRANCH
11563  067746  104401  001205                TYPE    ,$CRLF
11564  067752  005760  000002                TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
11565  067756  001404                        BEQ     15$             ;YES-BRANCH BYPASS INDENT
11566  067760  005104                        COM     R4              ;INDENT?
11567  067762  001002                        BNE     15$             ;NO-BRANCH
11568  067764  104401  070064                TYPE    ,SPACE2         ;YES-TYPE SPACES
11569  067770  012037  067776        15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
11570  067774  104401                        TYPE                    ;TYPE DH
11571  067776  000000                16$:    .WORD   0               ;DH POINTER GOES HERE
11572  070000  105710                        TSTB    (R0)            ;TYPE A DT?
11573  070002  001003                        BNE     21$             ;YES-BRANCH
11574  070004  062700  000002                ADD     #2,R0           ;INCREMENT DF POINTER
11575  070010  000754                        BR      14$             ;SEE IF END OF DF BLOCK
11576  070012  104401  001205        21$:    TYPE    ,$CRLF
11577  070016  005704                        TST     R4              ;INDENT?
11578  070020  001335                        BNE     10$             ;NO-BRANCH
11579  070022  104401  070064        17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES
11580  070026  000732                        BR      10$             ;LOOP
11581  070030  104414                20$:    RESREG
11582
11583  070032  032777  010000  111100        BIT     #SW12,@SWR      ;ABORT DRV AFTER 20 ERRS?
11584  070040  001410                        BEQ     25$             ;BR IF NO
11585  070042  023727  001103  000024        CMP     $ERFLG,#20.     ;ELSE SEE IF 20 ERRS
11586  070050  001004                        BNE     25$             ;BR IF NO
11587  070052  012706  001100                MOV     #STACK,SP       ;ELSE RESTORE STK
11588  070056  000137  043470                JMP     $EOP            ;AND DROP DRIVE
11589  070062  000207                25$:    RTS     PC
11590  070064  020040     000        SPACE2: .ASCIZ/  /              ;2 SPACES
```

```
 11591          000001                          .END
```

```
ABASE = 177440          1719    1760    1774#
ACDW1 = 000000          1719    1762
ACDW2 = 000000          1719    1763
ACLO  = 000010          1380#
ACPUOP= 000000          1719    1734
ACT11   003450          1945#   3237*
ADDW0 = 000000          1719    1764
ADDW1 = 000000          1719    1765
ADDW10= 000000          1719
ADDW11= 000000          1719
ADDW12= 000000          1719
ADDW13= 000000          1719
ADDW14= 000000          1719
ADDW15= 000000          1719
ADDW2 = 000000          1719    1766
ADDW3 = 000000          1719    1767
ADDW4 = 000000          1719    1768
ADDW5 = 000000          1719    1769
ADDW6 = 000000          1719    1770
ADDW7 = 000000          1719    1771
ADDW8 = 000000          1719
ADDW9 = 000000          1719
ADEVCT= 000000          1719    1725
ADEVM = 000000          1719    1761
AENV  = 000000          1719    1730
AENVM = 000000          1719    1731
AFATAL= 000000          1719    1722
AMADR1= 000000          1719    1747
AMADR2= 000000          1719    1751
AMADR3= 000000          1719    1754
AMADR4= 000000          1719    1757
AMAMS1= 000000          1719    1741
AMAMS2= 000000          1719    1749
AMAMS3= 000000          1719    1752
AMAMS4= 000000          1719    1755
AMSGAD= 000000          1719    1727
AMSGLG= 000000          1719    1728
AMSGTY= 000000          1719    1721
AMTYP1= 000000          1719    1742
AMTYP2= 000000          1719    1750
AMTYP3= 000000          1719    1753
AMTYP4= 000000          1719    1756
APASS = 000000          1719    1724
APRIOR= 000000          1719
APTCSU= 000040          9615    9787#
APTENV= 000001          9562    9608    9743    9785#
APTSIZ= 000200          3166    9784#
APTSPO= 000100          9610    9745    9786#
ASWREG= 000000          1719    1732
ATESTN= 000000          1719    1723
ATTN    003324          1877#   4689    8608    8627    8653
AUNIT = 000000          1719    1726
AUSWR = 000000          1719    1733
AVECT1= 000000          1719    1758
AVECT2= 000000          1719    1759
BADHDR  003320          1868#   3245*   5166*   6852*   9335
```

```
BADTMO  050426          3317    9390#
BAI     = 000020        1343#   5175    5324
BA16    = 000400        1329#
BA17    = 001000        1330#
BIT0    = 000001        1272#   1326    1358    1377    1935    3354    9259
BIT00   = 000001        1262#   1272
BIT01   = 000002        1261#   1271
BIT02   = 000004        1260#   1270
BIT03   = 000010        1259#   1269
BIT04   = 000020        1258#   1268
BIT05   = 000040        1257#   1267
BIT06   = 000100        1256#   1266
BIT07   = 000200        1255#   1265
BIT08   = 000400        1254#   1264    9497
BIT09   = 001000        1253#   1263    9505    9573
BIT1    = 000002        1271#   1359    1936
BIT10   = 002000        1252#   1331    1349    1368    1400    1414    1428    1441    1455    4665    9550
BIT11   = 004000        1251#   1332    1350    1369    1386    1401    1415    1429    1442    1456    4667    9512
BIT12   = 010000        1250#   1333    1351    1370    1402    1416    1430    1443    1457
BIT13   = 020000        1249#   1334    1352    1371    1387    1403    1417    1431    1444    1458    9557
BIT14   = 040000        1248#   1335    1353    1372    1388    1404    1418    1432    1445    1459    1472    9096    9128
                        9483
BIT15   = 100000        1247#   1336    1337    1354    1373    1389    1405    1475    4184    9096    9124
BIT2    = 000004        1270#   1360    1379    1937
BIT3    = 000010        1269#   1342    1361    1380
BIT4    = 000020        1268#   1343    1362    1381    1394    1422    1449    6055
BIT5    = 000040        1267#   1344    1363    1382    1395    1409    1423    1436    1450
BIT6    = 000100        1266#   1327    1345    1364    1383    1396    1410    1424    1437    1451
BIT7    = 000200        1265#   1328    1346    1365    1384    1397    1411    1425    1438    1452    3344
BIT8    = 000400        1264#   1329    1347    1366    1385    1398    1412    1426    1439    1453
BIT9    = 001000        1263#   1330    1348    1367    1399    1413    1427    1440    1454    4661
BPTVEC= 000014          1279#
BSE     = 000200        1365#
BSERR   001456          1842#   6336*   6342*   6630
BSE22H  002304          1856#   6287    6339    6358    9122
BSE22S  054752          6349    9126    10411#
BYP     044704          3424    3447    3451    3466    3518    3526    3544    3548    3552    3556    8680#
BYPCER  001462          1845#   3342*   3568*   4268*   4300*   4881*   4979*   6384*   6482*   8328*   8796
BYPFMT  001464          1846#   5456*   5548*   5783*   5875*   9094
BYPT16  001340          1790#   3079*   3084*   3089*   3094*   3099*   3103*   4321
BYT16   007026          1489    3083#
BYT16A  007046          1493    3088#
CALADD  001366          1803#   5462*   5789*   6642*   6673*   6695    6781    6799    6837    6844*   6855*   9090    9100
                        9102    9150
CALDIF  001360          1800#   4423*   4439    4471*   4502    5580*   5900*   6098*   6677*   7241*   7584*   7942*   8306*
                        11334   11346
CCLR    = 100000        1337#   3641    3948    3977    4005    4018    4107    4135    4194    4235    4340    4381    4431
                        4445    4479    4508    4560    4601    4704    4711    4728    4793    4904    4937    5016    5636
                        5906    5960    6160    6407    6440    6519    6755    7070    7291    7419    7634    7762    7992
                        8120    8349    8356    8373    8837    8847    8857    8867    9339    9357
CCYL    001354          1798#   4330*   4417*   4418    4424    4465*   4466    4472    7370*   7713*   8071*
CDT     = 002000        1331#   3839
CERR    = 100000        1336#   3404    3504    3802    3852    3899    4768    5103    5191    5244    5334    5380    5474
                        5511    5611    5683    5726    5801    5838    5932    5996    6039    6135    6220    6299    6567
                        6659    6730    6807    6871    6933    6947    7045    7117    7168    7266    7337    7394    7465
                        7514    7609    7680    7737    7808    7857    7920    7967    8038    8095    8166    8215    8288
```

```
                    8798    8893
CFMT   = 010000     1333#   5329    5341    5376    5399
CHKFLG  001466      1850#   8690*   8772    8776    8780
CHKMSG  044720      3686    3939    3962    3996    4029    4056    4083    4220    4254    4285    4366    4400    4586
                    4620    4670    4781    4812    4897    4923    4970    5001    5035    5130    5206    5271    5359
                    5417    5489    5593    5624    5655    5742    5816    5918    5945    5979    6057    6119    6148
                    6179    6311    6400    6426    6473    6504    6538    6709    6743    6774    6822    7029    7058
                    7089    7254    7279    7310    7381    7407    7438    7597    7622    7653    7724    7750    7781
                    7955    7980    8011    8082    8108    8139    8334    8690#
CKCERR  045452      8558    8562    8574    8579    8796#
CKSWR  = 104407     9482    9546    9572    10400#
CLEAR  = 000005     1314#   3979    4237    4383    4447    4510    4603    4713    4795    4906    5018    5638    5962
                    6162    6409    6521    6757    7072    7293    7421    7636    7764    7994    8122    8358    9359
CLKOF   047710      4017    4047    4074    4100    4156    4161    9230#
CLKON   047614      4008    4038    4065    4091    9208#
CLOCK   047654      3266    9219#
CLRFLG  043640      3187    8435#
COE    = 001000     1367#
COUNT   001372      1807#   4006*   4036*   4063*   4089*   9220*   9222*
CR     = 000015     1187#   9654    9664    10414   10419   10421   10426   10430   10440   10449   10456   10464   10469
                    10474   10479   10485   10492   10500   10505   10510   10514   10518   10521   10526   10531   10535
                    10539   10545   10550   10553   10558   10562   10564   10567   10570   10575   10579   10589   10597
                    10606   10612   10616   10619   10622   10629   10647   10851   10921   11053   11057   11063   11070
CRLF   = 000200     1188#   9625    9664
CTO    = 004000     1332#   8802
CYLADD  001364      1802#   3696    3972    4230    4292    4376    4434    4482    4596    4680    4750    4819    5011
                    5141    5217    5282    5427    5662    5663    5748    5955    6063    6186    6514    6781    7096
                    7317    7445    7660    7788    8018    8146    8343    8972*   8973*   8974*   8975*   8976*   8977*
                    8995*   8996*   8997*   8998*   8999*   9000*
CYLDIF  001362      1801#   3693    3969    4227    4296    4373    4439    4491    4497    4502    4593    4677    4787
                    5008    5138    5214    5279    5424    5600    5630    5924    5952    6126    6154    6511    6716
                    6749    7036    7064    7285    7413    7628    7756    7986    8114    8340    8957*   8958*   8959*
                    8960*   8961*   8962*   8963    8965*   8982*   8983*   8984*   8985*   8986*   8987*   8988    8990*
CYLINV  042744      7567    8257#
DATAO   001444      1835#   5184    5295
DATAOI  001446      1836#
DATAI   001450      1837#   5325    5434
DCK    = 100000     1373#
DCLO   = 000020     1381#
DCPAR  = 020000     1334#   3877    3896
DDISP  = 177570     1194#   1694    3154
DDPCH   003446      1944#   3233*   3412    3597
DDT    = 000400     1385#   3825    3846
DDUMP   003444      1943#   3200*   8435
DF1     067240      1985    1991    1997    2003    2008    2014    2020    2026    2094    2100    2111    2116    2121
                    2126    2131    2136    2141    2146    2151    2156    2161    2271    2296    2321    2346    2406
                    2421    2476    2481    2486    2551    11385#
DF10    067334      2032    2038    2043    2048    2053    2089    2106    2216    2266    2381    2386    2391    2396
                    2401    2411    2416    2426    2431    2436    2441    2491    2511    2516    2531    2536    2541
                    2546    2576    2581    2586    2596    2671    2676    2711    2726    2731    2736    2741    2776
                    2786    2791    2812    2817    2882    2978    2983    2988    3003    3008    3018    3023    3028
                    3033    3038    11422#
DF12    067350      2496    11429#
DF13    067364      2631    2636    11435#
DF14    067400      2641    11442#
DF15    067410      2606    2611    2746    11448#
```

```
DF16    067424          2591    2626    11455#
DF17    067434          2751    2766    11460#
DF2     067250          3063    3068    3074    11390#
DF20    067454          2058    2063    2068    2073    2196    2201    2226    2231    2236    2241    2246    2251    2256
                        2261    2276    2281    2286    2291    2301    2306    2311    2316    2326    2331    2336    2341
                        2361    2366    2371    2376    2446    2451    2456    2461    2556    2561    2566    2571    2646
                        2651    2656    2661    2686    2691    2696    2701    2716    2721    2842    2847    2852    2857
                        2897    2902    2907    2912    2917    2922    2927    2932    2937    2942    2947    2952    2957
                        2962    3043    3048    11468#
DF21    067510          2993    2998    11482#
DF22    067520          2078    2083    2166    2171    2186    2191    2351    2356    2466    2471    2501    2506    2601
                        2616    2621    2892    2967    2972    3053    3058    11486#
DF23    067554          2206    2211    2221    11501#
DF3     067270          2781    11399#
DF4     067274          3013    11401#
DF5     067304          2761    11406#
DF6     067320          2176    2181    2666    2681    2706    2796    2801    2807    2822    2827    2832    2837    2862
                        2867    2887    11413#
DH1     064066          1983    1989    1995    2001    2006    2012    2018    2024    2092    2098    2109    2114    2119
                        2124    2129    2134    2139    2144    2149    2154    2159    2269    2294    2319    2344    2404
                        2419    2474    2479    2484    2549    2639    2991    2996    3011    11078#  11394   11408   11415
                        11424   11431   11437   11450   11457   11464   11470   11488   11503
DH10    064444          2244    2249    2254    2259    2264    2614    2619    11121#
DH11    064473          2274    2279    2284    2289    3026    11125#
DH12    064532          2299    2304    2309    2314    3031    11131#
DH13    064573          2324    2329    2334    2339    3036    11137#
DH14    064642          2164    2169    2174    2179    2349    2354    2399    2414    2499    2504    2794    2799    2885
                        11144#
DH16    064672          2209    2369    2374    3041    3046    3051    3056    11148#
DH17    064734          2184    2189    2204    2214    2409    2534    2709    2714    2719    2724    2729    2734    2789
                        2935    2940    11154#
DH18    064754          2030    2036    2041    2046    2051    3016    11157#
DH19    064775          2104    2379    11160#
DH2     064103          11081#  11387   11396   11426   11466
DH20    065014          2384    11163#
DH21    065043          2087    2389    2424    2429    11167#
DH22    065071          2514    2529    2784    2880    2895    2900    2925    2930    11171#
DH23    065117          2779    11175#
DH25    065140          2434    2439    2444    2449    2454    2459    2464    2469    2664    2669    2674    2759    2805
                        11178#
DH26    065157          2194    2199    2359    2364    2739    2744    11181#
DH28    065203          11185#  11472   11490   11505
DH29    065264          11194#  11476   11494   11509
DH30    065345          2594    2599    2604    2609    2890    2915    2920    2955    2960    3006    11203#
DH31    065373          11207#  11403
DH32    065435          11213#  11474   11478   11492   11496   11507   11511
DH33    065462          2644    2649    2654    2659    2679    11217#
DH34    065502          2489    11220#
DH35    065526          2494    11224#
DH36    065554          11228#  11433
DH37    065636          2509    11237#
DH38    065702          2554    2559    2564    2569    2574    2579    2584    2976    11243#
DH39    065720          2629    2634    2905    2910    2945    2950    2965    2970    3001    11246#
DH40    065747          11250#  11444
DH41    066024          2684    2689    2694    2699    2704    2810    2986    11258#
DH42    066045          2764    11261#
```

```
DH44     066105            2774   11267*  11462
DH45     066155            2589   11274*
DH46     066221            2624   11281*
DH47     066266           11288*  11480   11498   11513
DH48     066330            2815    2820    2825    2830    2835    2840    2845    2850    2855    2860    2865   11294*
DH49     066362           11299*  11392
DH5      064154           11088*  11410
DH51     066436            2539    2544   11307*
DH52     066465           11311*  11484
DH53     066536            2749   11318*
DH6      064223           11095*  11417
DH7      064305           11104*  11439   11452
DH8      064354            2056    2061    2066    2071    2076    2081    2219   11111*
DH9      064400            2224    2229    2234    2239    2394    2981    3021   11115*
DI     = 040000            1335*
DISPLA   001142            1694*   3154*   3162*   9527*   9549*
DISPRE   000174            1484*   3162
DLT    = 100000            1354*   5114    5255    5391    5521    5694    5848    6007    6958    7128    7348    7476    7691
                           7819    8049    8177
DLY      044666            4020    8671*   8674
DMD    = 000040            1395*
DOSEEK   037016            7566    7568*
DOTIM    003502            1963*   3265*   3274*   3945
DRA    = 000001            1377*   3441    3534
DRDY   = 000200            1384*
DRIVS    003454            1947*   3349*   3356*   3373    3388*   3414*   3431    3522*   3586    8398    8490*
DRIVO    003456            1952*   3247    3351    3390    3490    8473
DRIV1    003460            1953*
DRIV2    003462            1954*
DRIV3    003464            1955*
DRIV4    003466            1956*
DRIV5    003470            1957*
DRIV6    003472            1958*
DRIV7    003474            1959*
DROT   = 000040            1382*
DRPAR  = 000010            1361*
DRVMSK= 000007             1341*   3407    3507
DRVPTR   001346            1794*   3247*   3591    3601*
DSC    = 040000            1388*
DSWR   = 177570            1193*   1693    3153
DTE    = 010000            1370*
DTYE   = 000040            1363*   3828    3849
DT1      066606            1984    1990    1996    2002    2007    2013    2019    2025    2031    2037    2042    2047    2052
                           2088    2093    2099    2105    2110    2115    2120    2125    2130    2135    2140    2145    2150
                           2155    2160    2215    2265    2270    2295    2320    2345    2380    2385    2390    2395    2400
                           2405    2410    2415    2420    2425    2430    2435    2440    2475    2480    2485    2490    2510
                           2515    2530    2535    2540    2545    2550    2575    2580    2585    2590    2595    2625    2670
                           2675    2710    2725    2730    2735    2740    2750    2765    2775    2785    2790    2811    2816
                           2881    2977    2982    2987    3002    3007    3017    3022    3027    3032    3037    3062    3067
                           3073   11328*
DT10     067012            2997   11352*
DT11     067030            2780   11355*
DT12     067034            3012   11356*
DT13     067054            2057    2062    2067    2072    2195    2200    2225    2230    2235    2240    2245    2250    2255
                           2260    2275    2280    2285    2290    2300    2305    2310    2315    2325    2330    2335    2340
                           2360    2365    2370    2375    2445    2450    2455    2460    2555    2560    2565    2570    2645
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    2650    2655    2660    2685    2690    2695    2700    2715    2720    2841    2846    2851    2856
                    2896    2901    2906    2911    2916    2921    2926    2931    2936    2941    2946    2951    2956
                    2961    3042    3047   11359#
DT14     067114     2077    2082    2165    2170    2185    2190    2350    2355    2465    2470    2500    2505    2600
                    2615    2620    2891    2966    2971    3052    3057   11366#
DT15     067164     2205    2210    2220   11374#
DT3      066630     2760   11331#
DT4      066652     2175    2180    2665    2680    2705    2821    2826    2831    2836    2861    2866   11334#
DT5      066672     2605    2610    2630    2635    2745   11337#
DT6      066714     2495   11340#
DT7      066734     2640   11343#
DT8      066754     2795    2800    2806    2886   11346#
DT9      066774     2992   11349#
D.ACLO=  000100     1437#
D.BRHM=  000100     1424#   3680    3936    3956    3990    4026    4053    4080    4214    4248    4279    4360    4394
                    4580    4614    4655    4775    4806    4894    4917    4967    4995    5029    5124    5200    5265
                    5353    5411    5483    5590    5618    5649    5736    5810    5915    5939    5973    6050    6116
                    6142    6173    6305    6397    6420    6470    6498    6532    6706    6737    6768    6816    7026
                    7052    7083    7251    7273    7304    7378    7401    7432    7594    7616    7647    7721    7744
                    7775    7952    7974    8005    8079    8102    8133    8331
D.CART=  000400     1426#   3680    3936    3956    3990    4026    4053    4080    4214    4248    4279    4360    4394
                    4580    4614    4655    4775    4806    4894    4917    4967    4995    5029    5124    5200    5265
                    5353    5411    5483    5590    5618    5649    5736    5810    5915    5939    5973    6050    6116
                    6142    6173    6305    6397    6420    6470    6498    6532    6706    6737    6768    6816    7026
                    7052    7083    7251    7273    7304    7378    7401    7432    7594    7616    7647    7721    7744
                    7775    7952    7974    8005    8079    8102    8133    8331
D.DDT =  000400     1412#   3822    3843
D.DOOR=  000200     1425#   3680    3936    3956    3990    4026    4053    4080    4214    4248    4279    4360    4394
                    4580    4614    4655    4775    4806    4894    4917    4967    4995    5029    5124    5200    5265
                    5353    5411    5483    5590    5618    5649    5736    5810    5915    5939    5973    6050    6116
                    6142    6173    6305    6397    6420    6470    6498    6532    6706    6737    6768    6816    7026
                    7052    7083    7251    7273    7304    7378    7401    7432    7594    7616    7647    7721    7744
                    7775    7952    7974    8005    8079    8102    8133    8331
D.DRA =  000040     1409#   3675    3678    3934    3954    3988    4024    4051    4078    4212    4246    4277    4358
                    4392    4578    4612    4649    4773    4804    4892    4915    4965    4993    5027    5122    5198
                    5263    5351    5409    5481    5588    5616    5647    5734    5808    5913    5937    5971    6048
                    6114    6140    6171    6303    6395    6418    6468    6496    6530    6704    6735    6766    6814
                    7024    7050    7081    7249    7271    7302    7376    7399    7430    7592    7614    7645    7719
                    7742    7773    7950    7972    8003    8077    8100    8131    8329
D.DRDY=  000200     1411#   3954    3988    4094    4212    4246    4277    4358    4392    4578    4612    4649    4773
                    4804    4892    4915    4993    5027    5122    5198    5263    5351    5409    5481    5616    5647
                    5734    5808    5937    5971    6048    6140    6171    6303    6395    6418    6496    6530    6735
                    6766    6814    7050    7081    7271    7302    7399    7430    7614    7645    7742    7773    7972
                    8003    8100    8131    8329
D.DROT=  020000     1444#
D.DSC =  040000     1418#   3954    4212    4277    4358    4578    4649    4773    4892    4965    4993    5616    5734
                    5937    6140    6395    6468    6496    6735    7050    7271    7399    7614    7742    7972    8100
                    8324    8329
D.FLT =  000200     1438#   3890    4278    4654    4893    4966    6396    6469    8318    8330
D.FORM=  001000     1413#   5345    5403
D.FWD =  002000     1428#   4011    4026    4041    4068    4080    5590    6116    6706    7251    7594    8079
D.HDFL=  000200     1452#
D.HDHM=  000040     1423#   3680    3936    9046
D.IDAE=  000040     1436#   4654    4724    8315    8330    8369
D.ILF =  000400     1439#
D.LIMD=  020000     1458#   4968    6471    9029
```

```
D.LOAD= 010000  1430* 4026  9063
D.MHD = 000400  1453*
D.NMOV= 010000  1457* 4968  6471
D.OFF = 002000  1414*
D.PAR = 001000  1440* 3874  3893  4278  4893  6396
D.PIP = 020000  1417* 4024  4051  4078  4965  5588  5913  6114  6468  6704  7024  7249  7376
                7592  7719  7950  8077  8321
D.PLO = 004000  1456*
D.REV = 004000  1429* 4053  4967  5915  6470  7026  7378  7721  7952
D.RTZ = 020000  1431* 4053  4080  4205  4351  4571  5909  5915  9350
D.SECT= 000020  1449*
D.SKI = 002000  1441* 4654  4966  6469  8330
D.SPIN= 010000  1416* 3926  3934  3954  3988  4024  4051  4078  4110  4212  4246  4277  4358
                4392  4578  4612  4649  4773  4804  4892  4915  4965  4993  5027  5122  5198
                5263  5351  5409  5481  5588  5616  5647  5734  5808  5913  5937  5971  6048
                6114  6140  6171  6303  6395  6418  6468  6496  6530  6704  6735  6766  6814
                7024  7050  7081  7249  7271  7302  7376  7399  7430  7592  7614  7645  7719
                7742  7773  7950  7972  8003  8077  8100  8131  8329
D.SPLS= 010000  1443*
D.SPOK= 001000  1427* 3644  3956  3990  4026  4053  4080  4138  4214  4248  4279  4360  4394
                4580  4614  4655  4775  4806  4894  4917  4967  4995  5029  5124  5200  5265
                5353  5411  5483  5590  5618  5649  5736  5810  5915  5939  5973  6050  6116
                6142  6173  6305  6397  6420  6470  6498  6532  6706  6737  6768  6816  7026
                7052  7083  7251  7273  7304  7378  7401  7432  7594  7616  7647  7721  7744
                7775  7952  7974  8005  8079  8102  8133  8331
D.SSP = 000020  1422* 3956  3990  4053  4080  4214  4248  4279  4360  4394  4580  4614  4655
                4775  4806  4894  4917  4967  4995  5029  5124  5200  5265  5353  5411  5483
                5590  5618  5649  5736  5810  5915  5939  5973  6050  6116  6142  6173  6305
                6397  6420  6470  6498  6532  6706  6737  6768  6816  7026  7052  7083  7251
                7273  7304  7378  7401  7432  7594  7616  7647  7721  7744  7775  7952  7974
                8005  8079  8102  8133  8331
D.SUNS= 040000  1459*
D.TIB = 002000  1455*
D.UNLD= 040000  1432* 4943  4967  6446  6470
D.UNS = 040000  1445*
D.VV  = 000100  1410* 3673  3678  3758  3772  3934  3954  3988  4024  4051  4078  4212  4246
                4277  4358  4392  4578  4612  4649  4773  4804  4892  4915  4965  4993  5027
                5122  5198  5263  5351  5409  5481  5588  5616  5647  5734  5808  5913  5937
                5971  6048  6114  6140  6171  6303  6395  6418  6468  6496  6530  6704  6735
                6766  6814  7024  7050  7081  7249  7271  7302  7376  7399  7430  7592  7614
                7645  7719  7742  7773  7950  7972  8003  8077  8100  8131  8329
D.WCUR= 000040  1450*
D.WGAT= 000100  1451*
D.WLE = 004000  1442*
D.WRL = 004000  1415*
D.XERR= 001000  1454*
ECCW  = 020000  1403*
ECH   = 000100  1364*
EMTVEC= 000030  1282*  3137*  3138*
EM1     057255  8499  10629*
EM10    057735  2017  10684*
EM11    060000  2023  10690*
EM12    060044  2029  2378  2383  2388  2393  2398  2408  2433  2513  2538  2593  2628  2738
                10696*
EM13    060075  2035  2213  2263  2413  2438  2543  2573  2814  10701*
EM14    060117  2040  10705*
```

EM15	060144	2045	10709#									
EM16	060176	2050	2834	10714#								
EM17	060223	2055	2193	2223	2243	2273	2296	2323	2348	2368	2443	2553
		2643	2683	2713	2839	2924	2944	2954	10718#			
EM18	060236	2060	2198	2228	2248	2278	2303	2328	2353	2373	2448	2558
		2648	2688	2844	2894	2904	2914	2934	10720#			
EM19	060251	2065	2233	2253	2283	2308	2333	2358	2453	2498	2508	2563
		2653	2693	2718	2849	2929	2949	2959	3040	10722#		
EM2	057327	1982	10637#									
EM20	060264	2070	2238	2258	2288	2313	2338	2363	2458	2503	2568	2658
		2698	2854	2899	2909	2919	2939	3045	10724#			
EM21	060277	2086	2608	2633	2668	2708	2743	10726#				
EM22	060321	2091	10730#									
EM23	060360	2097	10736#									
EM24	060440	2103	10744#									
EM25	060464	2108	10748#									
EM26	060512	2113	10752#									
EM27	060532	2118	10755#									
EM28	060553	2123	10758#									
EM29	060600	2128	10762#									
EM3	057416	9394	10647#									
EM30	060647	2133	10769#									
EM31	060700	2138	10774#									
EM32	060726	2143	2824	10778#								
EM33	060752	2148	10782#									
EM34	061007	2153	10787#									
EM35	061040	2158	10792#									
EM36	061113	2163	2173	2663	2758	2793	2804	2864	10800#			
EM37	061153	2168	10806#									
EM38	061213	2178	2678	2703	2798	2859	10812#					
EM39	061264	2183	2423	2463	2598	2613	2723	2964	10819#			
EM40	061321	2188	2428	2468	2618	10824#						
EM41	061347	2889	2969	10828#								
EM43	061410	2203	10834#									
EM44	061437	2208	10838#									
EM45	061473	2343	10843#									
EM46	061614	2075	3050	10857#								
EM47	061627	2080	3055	10859#								
EM48	061642	2488	2493	10861#								
EM49	061655	2318	10863#									
EM5	057464	1988	3071	10654#								
EM50	061743	2268	10873#									
EM51	062014	2293	10880#									
EM52	062114	2473	10891#									
EM53	062153	2478	10897#									
EM54	062204	2483	10902#									
EM55	062237	2528	10907#									
EM56	062272	2778	10912#									
EM57	062315	3010	10916#									
EM58	062351	3015	10921#									
EM59	062417	2548	10928#									
EM6	057505	1994	10657#									
EM60	062450	2578	10933#									
EM61	062504	2583	10938#									
EM63	062540	2603	10943#									
EM64	062561	2218	10946#									

```
EM65    062610          2638   10950#
EM66    062630          2728   10953#
EM67    062664          2673   10958#
EM68    062723          2733    2788   10964#
EM69    062762          2773   10970#
EM7     057526          2000   10660#
EM70    063002          3020    3025    3030    3035   10973#
EM71    063015          2783   10975#
EM72    063036          2588    2623   10978#
EM73    063063          3060   10982#
EM74    063104          2809   10985#
EM75    063131          2819   10989#
EM76    063157          2829   10993#
EM77    063200          2879   10996#
EM78    063226          2884   11000#
EM79    063313          3065   11009#
EM8     057615          2005   10670#
EM80    063334          2975   11012#
EM81    063364          2980   11016#
EM82    063412          2985   11020#
EM83    063437          2990    2995   11024#
EM84    063477          3000    3005   11030#
EM85    063527          2533   11035#
EM86    063560          3061    3066    3072   11040#
EM87    063645          2748    2763   11049#
EM88    063672          2403   11053#
EM89    063763          2418   11063#
EM9     057665          2011   10677#
ERRVEC= 000004          1275#   3151    3152*   3163*   3170*   3171*   3184*   3185*   3250*   3254*   3261*   3299*   3317*
                        9488    9489*   9491*   9494*
ESEC    001404          1813#   4842*   4843*   4853   11340
E.A0    003424          1923#   3675*   3678*   3934*   3954*   3988*   4024*   4051*   4078*   4212*   4246*   4277*   4358*
                        4392*   4578*   4612*   4649*   4773*   4804*   4892*   4915*   4965*   4993*   5027*   5122*   5198*
                        5263*   5351*   5409*   5481*   5588*   5616*   5647*   5734*   5808*   5913*   5937*   5971*   6048*
                        6114*   6140*   6171*   6303*   6395*   6418*   6468*   6496*   6530*   6704*   6735*   6766*   6814*
                        7024*   7050*   7081*   7249*   7271*   7302*   7376*   7399*   7430*   7592*   7614*   7645*   7719*
                        7742*   7773*   7950*   7972*   8003*   8077*   8100*   8131*   8329*   8694*   8701    8703*   8732
                       11359   11366   11374
E.A1    003430          1925#   3680*   3936*   3956*   3990*   4025*   4053*   4080*   4214*   4248*   4279*   4360*   4394*
                        4580*   4614*   4655*   4775*   4806*   4894*   4917*   4967*   4995*   5029*   5124*   5200*   5265*
                        5353*   5411*   5483*   5590*   5618*   5649*   5736*   5810*   5915*   5939*   5973*   6050*   6116*
                        6142*   6173*   6305*   6397*   6420*   6470*   6498*   6532*   6706*   6737*   6768*   6816*   7026*
                        7052*   7083*   7251*   7273*   7304*   7378*   7401*   7432*   7594*   7616*   7647*   7721*   7744*
                        7775*   7952*   7974*   8005*   8079*   8102*   8133*   8331*   8695*   8705    8707*   8752   11359
                       11366   11374
E.A2    003434          1927#   3682*   3958*   3992*   4216*   4250*   4281*   4362*   4396*   4582*   4616*   4657*   4777*
                        4808*   4919*   4997*   5031*   5126*   5202*   5267*   5355*   5413*   5485*   5620*   5651*   5738*
                        5812*   5941*   5975*   6052*   6144*   6175*   6307*   6422*   6500*   6534*   6739*   6770*   6818*
                        7054*   7085*   7275*   7306*   7403*   7434*   7618*   7649*   7746*   7777*   7976*   8007*   8104*
                        8135*   8696*   8709    8711*  11366   11374
E.A3    003440          1929#   8697*
E.B0    003426          1924#   3679*   3935*   3955*   3989*   4025*   4052*   4079*   4213*   4247*   4278*   4359*   4393*
                        4579*   4613*   4652*   4654*   4774*   4805*   4893*   4916*   4966*   4994*   5028*   5123*   5199*
                        5264*   5352*   5410*   5482*   5589*   5617*   5648*   5735*   5809*   5914*   5938*   5972*   6049*
                        6115*   6141*   6172*   6304*   6396*   6419*   6469*   6497*   6531*   6705*   6736*   6767*   6815*
                        7025*   7051*   7082*   7250*   7272*   7303*   7377*   7400*   7431*   7593*   7615*   7646*   7720*
```

```
                    7743*  7774*  7951*  7973*  8004*  8078*  8101*  8132*  8330*  8713   8715*  8742   11359
                   11366  11374
E.B1     003432     1926#  3681*  3937*  3957*  3991*  4027*  4054*  4081*  4215*  4249*  4280*  4361*  4395*
                    4581*  4615*  4656*  4776*  4807*  4895*  4918*  4968*  4996*  5030*  5125*  5201*  5266*
                    5354*  5412*  5484*  5591*  5619*  5650*  5737*  5811*  5916*  5940*  5974*  6051*  6117*
                    6143*  6174*  6306*  6398*  6421*  6471*  6499*  6533*  6707*  6738*  6769*  6817*  7027*
                    7053*  7084*  7252*  7274*  7305*  7379*  7402*  7433*  7595*  7617*  7648*  7722*  7745*
                    7776*  7953*  7975*  8006*  8080*  8103*  8134*  8332*  8717   8719*  8762   11359  11366
                   11374
E.B2     003436     1928#  3683*  3959*  3993*  4217*  4251*  4282*  4283*  4363*  4397*  4583*  4617*  4658*
                    4778*  4809*  4920*  4998*  5032*  5127*  5203*  5268*  5356*  5414*  5486*  5621*  5652*
                    5739*  5813*  5942*  5976*  6053*  6055*  6145*  6176*  6308*  6423*  6501*  6535*  6740*
                    6771*  6819*  7055*  7086*  7276*  7307*  7404*  7435*  7619*  7650*  7747*  7778*  7977*
                    8008*  8105*  8136*  8721   8723*  11366  11374
E.B3     003442     1930#  3684*  3960*  3994*  4218*  4252*  4364*  4398*  4584*  4618*  4661*  4665*  4667*
                    4779*  4810*  4921*  4999*  5033*  5128*  5204*  5269*  5357*  5415*  5487*  5622*  5653*
                    5740*  5814*  5943*  5977*  6054*  6146*  6177*  6309*  6424*  6502*  6536*  6741*  6772*
                    6820*  7056*  7087*  7277*  7308*  7405*  7436*  7620*  7651*  7748*  7779*  7978*  8009*
                    8106*  8137*  8725   8727*  11374
FATT1    044512     3950   4209   4355   4575   4730   4989   5062   6492   6599   8375   8624#  9332   9354
FATT2    044606     4766   5609   5724   5930   6037   6133   6218   6565   6728   6869   6931   7043   7166
                    7264   7392   7512   7607   7735   7855   7918   7965   8093   8213   8286   8651#
FHDHM    047020     4984   6487   9043#  9051
FHDTAB   047142     5465   5792   6645   6670   6793   9076#
FLGTST   047406     9123   9127   9143#
FLIM     046742     4939   6442   9025#
FLOAD    047074     4986   6489   9060#
FMTE   = 000020     1362#
FMT1     001440     1832#  9086*  9087*  9088*  9093
FNS22    046322     4840   8937#
FORM     031646     6607#  9369
FORMAT   001436     1831#  5464*  5791*  6644*  6675*  9086   9120
FRCYL    001350     1796#  4328*  4464   4465   4480   4482   4486   4525*  4526   4528   5578*  5898*  6096*
                    6192*  6678*  6845*  6856*  7007*  7141*  7227*  7238   7369   7370   7445   7457   7484
                    7570*  7581   7712   7713   7788   7800   7827   7928*  7939   8070   8071   8146   9158
                    8185   8296*  11334  11346  11352
FRDY     044176     3383   3397   3402   3497   3502   3632   3769   3841   3922   3981   4126   4200   4239
                    4273   4346   4385   4427   4449   4475   4512   4566   4605   4645   4708   4715   4762
                    4797   4888   4908   4935   5020   5058   5101   5188   5242   5331   5343   5378   5401
                    5471   5509   5586   5640   5681   5720   5798   5836   5904   5964   5994   6033   6112
                    6164   6214   6296   6391   6411   6438   6523   6561   6595   6656   6702   6759   6804
                    6865   6927   6945   7022   7074   7115   7162   7247   7295   7335   7374   7423   7463
                    7508   7590   7638   7678   7717   7766   7806   7851   7914   7948   7996   8036   8075
                    8124   8164   8209   8282   8310   8353   8360   8553#  8556   8826   8888   9328   9345
                    9361
FRDY1    044244     8568#  8571   8842   8852   8862   8871
FSO22    046236     4836   8917#
FTITLE   001344     1792#  8446   8448*
GBA      044100     3219   8513#
GDRVS    043740     3217   8468#
GINT     044126     3221   8526#
GNS    = ****** U   1483   10392  10393  10394  10395  10396  10398  10400  10401  10402  10403  10404  10405
                   10406
GO     = 000001     1326#
GSTAT    045534     3643   3672   3724   3757   3800   3889   3925   4010   4040   4067   4093   4109   4137
                    4204   4350   4570   4723   4938   5116   5190   5257   5333   5393   5473   5523   5696
```

```
                    5800   5850   5908   6009   6298   6441   6658   6806   6960   7130   7350   7478   7693
                    7821   8051   8179   8368   8634   8638   8658   8662   8822#  8892   8903   8956   8971
                    9028   9045   9062   9349
GSTAT1  045576      8692   8835#
GTSWR = 104406      8457  10398#
HASOF   003352      1895#  8592*  8608   11328  11363  11371  11380
HBA     003342      1891#  8588*
HCS1    003334      1888#  3404   3504   3802   3852   3877   3896   3899   4768   5103   5191   5244   5334
                    5380   5474   5511   5611   5683   5726   5801   5838   5932   5996   6039   6135   6220
                    6299   6567   6659   6730   6807   6871   6933   6947   7045   7117   7168   7266   7337
                    7394   7465   7514   7609   7680   7737   7808   7857   7920   7967   8038   8095   8166
                    8215   8288   8585*  8798   8802   8893   11328  11331  11337  11340  11356  11363  11371
                   11380
HCS2    003336      1889#  3437   3439   3443   3455   3530   3532   3536   3539   3788   3805   8586*  8807
                    8811  11328  11331  11337  11340  11356  11363  11371  11380
HDA     003344      1892#  8589*  11337  11343  11349  11352  11356
HDB     003356      1897#
HDC     003354      1896#  8593*  11331  11334  11346  11356  11363  11371  11380
HDS     003346      1893#  3441   3534   3825   3846   8590*  11328  11331  11337  11363  11371  11380
HDTAB   001470      1852#  5176   5184*  5458   5535   5785   5862   6648   6797   9078
HDWD    001454      1840#  5296*  5297   5435*  5436   5537*  5539   5864*  5866   11343
HD1     001434      1830#  9080*  9081*  9082*  9083*  9084*  9085*  9092
HEAD    001430      1828#  5085*  5094*  5095   5096*  5145*  5146   5167*  5178*  5179   5180*  5182   5222*
                    5223   5226*  5235*  5236   5237*  5286*  5287   5288*  5293   5304*  5305   5463*  5790*
                    6643*  6650*  6651   6652*  6666*  6667   6674*  6696*  6697   6698*  6794*  6795   6796*
                    6834*  6835   6841   6843*  6917*  6918   9080   9155
HEADA   001432      1829#  3700   4628   4690   9015*  9016*  9017*  9018*
HER     003350      1894#  3828   3849   8591*  11328  11331  11337  11363  11371  11380
HMR1    003360      1898#  8594*
HMR2    003362      1899#  3406   3506   3644   3673   3725   3758   3772   3822   3843   3926   4011   4041
                    4068   4094   4110   4138   4205   4351   4571   4943   5345   5403   5909   6446   8321
                    8324   8572*  8577*  8595*  8844   8854   8864   8873   8957   9046   9063   9350  11328
                   11331  11334  11337  11340  11343  11346  11349  11352  11356
HMR3    003364      1900#  3874   3890   3893   4724   8315   8318   8369   8573*  8578*  8596*  8845   8855
                    8865   8874   8904   8972   9029  11328  11331  11334  11337  11340  11343  11346  11349
                   11352  11356
HOLD    044332      8557   8561   8585#  8836
HPAT    003370      1902#  8598*
HPEND   003322      1870#  3246*  6847   6851*  9337*
HPOS    003366      1901#  8597*
HT    = 000011      1185#  9623   9664
HVRC  = 000400      1366#
HWC     003340      1890#  8587*
HZ      001370      1805#  4006   4036   4063   4089   9222
H.A0    003404      1912#  8732   8873*  11359  11369  11377
H.A1    003410      1914#  8752   8864*  11359  11369  11377
H.A2    003414      1916#  8854*  8982   11369  11377
H.A3    003420      1918#  8844*
H.B0    003406      1913#  8742   8874*  11359  11369  11377
H.B1    003412      1915#  8762   8865*  11359  11369  11377
H.B2    003416      1917#  8855*  8995   11369  11377
H.B3    003422      1919#  8845*  9005   9015   11377
IDAE  = 002000      1368#
IE    = 000100      1327#
ILF   = 000001      1358#
INTER   050556      8541   9434#
```

```
INVCYL  003304          1860#   8305
IOTVEC= 000020          1280#   3135*   3136*
IR    = 000100          1345#
LCLKF   003476          1961#   3252*
LCVEC   001330          1782#   3253
LF    = 000012          1186#   9658    9664    10414   10419   10421   10426   10430   10440   10449   10456   10464   10469
                       10474   10479   10485   10492   10500   10505   10510   10514   10518   10521   10526   10531   10535
                       10539   10545   10550   10553   10558   10562   10564   10567   10570   10575   10579   10589   10597
                       10606   10612   10616   10619   10622   10629   10647   10851   10921   11053   11057   11063   11070
LIMERR  001460          1843#   6626    9025*   9033*
LKS     001326          1780#   3251    9211*   9232*
LPFLG   001410          1816#   4149    4157*   4162*   4880*   5064*   5754    5760*   5765*   6068    6073*   6078*   6236
                       6244*   6250*   6383*   6583    6601*   6887    6899    6905*   6911*   6976*   6981*   7184    7192*
                       7198*   7530    7536*   7541*   7546*   7551*   7873    7879*   7884*   7889*   7894*   8231    8237*
                       8242*   8247*   8252*   9291*
MCLK  = 000400          1398#
MDS   = 001000          1348#   3437    3530    8811
MDTST   007070          1495    3093#
MDTSTA  007110          1497    3098#
MEMBAS= 172100          1786#   3173    3182
MEMERR  050514          3176    9420#
MEMPAR  007442          3170#
MEMVEC= 000114          1785#   3176*   3177*
MERD  = 001000          1399#
MESMSK= 000017          1393#
MEWD  = 002000          1400#
MIND  = 000200          1397#
MODTST  001342          1791#   3080*   3085*   3090*   3095*   3100*   3104*   6622    6690    6720    6830    7003    7565
MSG1    055116          8449   10414#
MSG10   056322          3348    3387   10535#
MSG11   056347          9461   10539#
MSG12   056407          6638   10545#
MSG13   056440          3275   10550#
MSG14   056516          8680   10558#
MSG15   056541          3457    3602   10562#
MSG16   056553          3722   10564#
MSG17   056571          6357   10567#
MSG18   056606          5106    5118    5194    5247    5259    5337    5383    5395    5477    5514    5525    5686    5698
                       5804    5841    5852    5999    6011    6662    6810    6950    6962    7120    7132    7340    7352
                       7468    7480    7683    7695    7811    7823    8041    8053    8169    8181   10570#
MSG19   056642          3588   10575#
MSG2    056020          3201   10500#
MSG20   056672          6624   10579#
MSG21   057041          7005   10597#
MSG22   057121          7002   10606#
MSG3    056056          3216   10505#
MSG4    056110          3218   10510#
MSG5    056136          3220   10514#
MSG6    056163          9436   10518#
MSG7    056201          3235   10521#
MSG74   057151          9318   10612#
MSG75   057200          9320   10616#
MSG76   057222          9373   10619#
MSG77   057240          9377   10622#
MSG8    056236          4326   10526#
MSG9    056274          4323   10531#
```

```
MSP    = 000100          1396#
MULT6    047572          9173    9199#
M.ALGN= 040000           1472#
M.CADD= 017760           1471#   8973    8996
M.CDIF= 017760           1464#   8958    8983
M.DRV  = 000007          1463#
M.HEAD= 007000           1474#   9016
M.ID   = 000003          1470#
M.OFST= 017760           1465#
M.PAR  = 100000          1475#   9261
M.SECT= 000760           1473#   8905    9006
M.SER  = 077770          1466#
NED    = 010000          1351#   3443    3455    3536    3539    8807
NEM    = 004000          1350#
NUDRV    011356          3375    3469    3568#   8400
NXF    = 000004          1360#
OFFSET= 000015           1318#
OFST   = 000004          1379#
OPI    = 020000          1371#
OR     = 000200          1346#
PACK   = 000003          1313#   3767
PARAM    001336          1789#   3078*   3083*   3088*   3093*   3098*   3102*   3213
PARSRT   007006          1491    3078#
PAT    = 000020          1394#   3888    4269    4415    4463    4884    6387
PCA    = 004000          1401#
PCD    = 010000          1402#
PCLKF    003500          1962#   3256*   3263*   9209    9230
PCVEC    001332          1783#   3257    3264
PCYL     001356          1799#   4331*   4419    4424*   4467    4472*
PFSRT    011646          3655#   9463
PGE    = 002000          1349#
PIP    = 020000          1387#
PIRQ   = 177772          1192#
PIRQVE= 000240           1286#
PKRB     001324          1779#
PKS      001320          1777#   3255    3262    9214*   9234*
PKSB     001322          1778#   9213*
PPTP     003452          1946#   3204*
PRGSRT   007146          3081    3086    3091    3096    3101    3105#   8500
PR0    = 000000          1209#   3107    3294
PR1    = 000040          1210#
PR2    = 000100          1211#
PR3    = 000140          1212#
PR4    = 000200          1213#
PR5    = 000240          1214#   1776
PR6    = 000300          1215#   3267
PR7    = 000340          1216#   3171    3177    9458    9459
PS     = 177776          1189#   1190
PSEC     001402          1812#   4839*   4842    4844*   4846    4849    8940    8943    11340
PSW    = 177776          1190#
PWRVEC= 000024           1281#   3141*   3142*   9448*   9457*   9458*
RCYLA    046620          8778    8995#
RCYLD    046546          8774    8982#
RDCHR  = 104410          10066   10401#
RDCYLA   046476          4433    4481    4749    8970#
RDCYLD   046412          4438    4490    4496    4501    8955#
```

```
RDDATA= 000021          1320#   6294
RDGATE= 100000          1405#
RDHEAD= 000025          1322#   5099    5240    5376    5507    5679    5834    5992    6943    7113    7333    7461    7676
                        7804    8034    8162
RDLIN = 104411          8468   10140   10402#
RDOCT = 104412          8513    8526   10403#
RDSEC   046166          4845    4848    8902#   8919    8922    8939    8942    9171
RDY   = 000200          1328#   8553    8568
RECAL = 000013          1317#   4196    4342    4562    4706    5902    8351    9341
RESREG= 104414         10200   10270   10405#  11581
RESVEC= 000010          1276#
RHEAD   046714          8783    9015#
RHTAB   001674          1853#   5098    5239    5290    5375    5431    5505    5528    5678    5703    5707    5832    5855
                        5991    6015    6019    6942    6966    7112    7136    7332    7356    7460    7484    7675    7699
                        7803    7827    8033    8057    8161    8185    9178    9180    9184    9187   11349   11352
RKASOF= 000016          1299#   3308    8592    8627    8653
RKBA  = 000004          1294#   3304    5176*   5325*   5458*   5785*   6288*   6331*   6350*   6648*   6797*   8588
RKCS1 = 000000          1292#   3301    3400*   3500*   3630*   3641*   3767*   3839*   3920*   3948*   3977*   3979*   4005*
                        4018*   4107*   4124*   4135*   4194*   4196*   4235*   4237*   4271*   4340*   4342*   4381*   4383*
                        4425*   4431*   4445*   4447*   4473*   4479*   4508*   4510*   4560*   4562*   4601*   4603*   4643*
                        4704*   4706*   4711*   4713*   4728*   4760*   4793*   4795*   4886*   4904*   4906*   4933*   4937*
                        5016*   5018*   5056*   5099*   5186*   5240*   5329*   5341*   5376*   5399*   5469*   5507*   5584*
                        5636*   5638*   5679*   5718*   5796*   5834*   5902*   5906*   5960*   5962*   5992*   6031*   6110*
                        6160*   6162*   6212*   6294*   6389*   6407*   6409*   6436*   6440*   6519*   6521*   6559*   6593*
                        6654*   6700*   6755*   6757*   6802*   6863*   6925*   6943*   7020*   7070*   7072*   7113*   7160*
                        7245*   7291*   7293*   7333*   7372*   7419*   7421*   7461*   7506*   7588*   7634*   7636*   7676*
                        7715*   7762*   7764*   7804*   7849*   7912*   7946*   7992*   7994*   8034*   8073*   8120*   8122*
                        8162*   8207*   8280*   8308*   8349*   8351*   8356*   8358*   8373*   8553    8568    8585    8824*
                        8837*   8840*   8847*   8850*   8857*   8860*   8867*   8869*   9326*   9339*   9341*   9357*   9359*
RKCS2 = 000010          1296#   3302    3381*   3395*   3399*   3495*   3499*   3799*   3978*   4195*   4236*   4341*   4382*
                        4446*   4509*   4561*   4602*   4705*   4712*   4794*   4905*   5017*   5114    5175*   5255    5324*
                        5391    5521    5637*   5694    5848    5961*   6007    6161*   6408*   6520*   6756*   6958    7071*
                        7128    7292*   7348    7420*   7476    7635*   7691    7763*   7819    7993*   8049    8121*   8177
                        8350*   8357*   8586    8823*   8838*   8848*   8858*   8868*   8886*   8890*   9340*   9358*
RKDA  = 000006          1295#   3305    4640*   5095*   5179*   5236*   5287*   6290*   6333*   6352*   6651*   6697*   6795*
                        6922*   8589
RKDB  = 000024          1301#   3310    5109    5110    5111    5250    5251    5252    5386    5387    5388    5517    5518
                        5519    5689    5690    5691    5844    5845    5846    6002    6003    6004    6953    6954    6955
                        7123    7124    7125    7343    7344    7345    7471    7472    7473    7686    7687    7688    7814
                        7815    7816    8044    8045    8046    8172    8173    8174
RKDC  = 000020          1300#   3309    4270*   4416*   4432*   4464*   4480*   4885*   4932*   5467*   5503*   5581*   5676*
                        5794*   5830*   6029*   6108*   6210*   6292*   6388*   6435*   6557*   6695*   6799*   6861*   7018*
                        7109*   7158*   7243*   7329*   7369*   7457*   7504*   7586*   7672*   7712*   7800*   7847*   7910*
                        7944*   8030*   8070*   8158*   8205*   8307*   8593
RKDS  = 000012          1297#   3306    8590
RKECPS= 000030          1305#   3314    8597
RKECPT= 000032          1306#   3315    8598
RKER  = 000014          1298#   3307    8591
RKMR1 = 000026          1302#   3311    3642*   3723*   3756*   3888*   4009*   4039*   4066*   4092*   4108*   4136*   4203*
                        4269*   4349*   4415*   4463*   4569*   4884*   5907*   6387*   8594    8633*   8637*   8657*   8661*
                        8839*   8849*   8859*   8891*   8902*   8955*   8970*   9027*   9044*   9061*   9348*
RKMR2 = 000034          1303#   3312    8572    8577    8595
RKMR3 = 000036          1304#   3313    8573    8578    8596
RKPRI   001316          1776#   8542
RKVEC   001314          1775#   3240*   8530*   8533*   8540
RKWC  = 000002          1293#   3303    5177*   5326*   5459*   5786*   6293*   6649*   6798*   8587
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
RLS    = 000010          1342#   3799
RSEC     046656          8782    9005#
SAVREG= 104413          10185   10244   10404#  11525
SBPAR    047732          8702    8706    8710    8714    8718    8722    8726    9247#
SCLR   = 000040          1344#   3381    3395    3495    8836
SCOP1  = 104415          3392    3492    3761    3793    3833    3882    4187    4262    4333    4408    4457    4552    4633
                         4754    5088    5169    5227    5368    5497    5669    5824    6101    6684    6786    7011    7101
                         7231    7322    7362    7450    7574    7665    7705    7793    7932    8023    8063    8151    8299
                        10406#
SCOP1$   050004          9271#  10406
SDC    = 000000          1173    1494   10413   11591
SEC      001374          1809#   4007*   4037*   4064*   4090*   9223*
SECFLG   047336          9098    9119#
SECNT    001400          1811#   4834*   4856*   5366*   5496*   5823*
SECTOR   001406          1814#   4844    4846    4849    4853    8904*   8905*   8906*   8907*   8908*   8909*   8920    8923
                         8940    8943    9005*   9006*   9007*   9008*   9009*   9010*   9172*   9176    9177*   9178*   9189
                         9199*   9200    9201*   9202*
SEEK   = 000017          1319#   4271    4425    4473    4643    4760    4886    4933    5584    5718    6031    6110    6212
                         6389    6436    6559    6700    6863    6925    7020    7160    7245    7372    7506    7588    7715
                         7849    7912    7946    8073    8207    8280    8308
SELDRV= 000001           1312#   3400    3500    3839    5341    5399    8824    8840    8850    8860    8869
SETINT   044160          3241    8531    8540#
SIZFLG   003504          1964#   3242*   3385    8486*   8506*
SKI    = 000002          1359#
SORT     047464          5531    5858    9169#
SPACE2   070064         11559   11568   11579   11590#
SRTSPL= 000011           1316#   3920    5056    6593    9326
SRTTAB   002100          1854#   5534    5861    9181
STACK  = 001100          1180#   3106    3133    3292    3341    3393    3488    3493    3582    3623    3667    3715    3754
                         3762    3784    3794    3818    3834    3870    3883    3915    4181    4188    4263    4319    4334
                         4409    4458    4548    4553    4634    4745    4755    4828    4875    5083    5089    5164    5170
                         5228    5320    5369    5451    5498    5575    5670    5778    5825    5894    6092    6102    6279
                         6378    6621    6685    6787    7000    7012    7102    7225    7232    7323    7363    7451    7563
                         7575    7666    7706    7794    7904    7933    8024    8064    8152    8275    8300    8396    8788
                         9429    9460   11587
START    007132          1487    3102#
START1   007550          3195#
STKLMT= 177774           1191#
STOP     050106          6853    9309#   9907    9990
ST2      007616          3202    3213#
ST3      007662          3203    3231#
ST4      007714          3215    3236    3239#
ST5      007740          3222    3244#   3379    3435    8425    9379
SUBCLR   046106          3627    3637    3669    3719    3764    3786    3796    3820    3836    3872    3885    3917    4121
                         4131    4145    4190    4265    4336    4411    4460    4555    4636    4747    4757    4830    4877
                         4981    5053    5091    5172    5230    5322    5371    5453    5500    5576    5672    5715    5730
                         5780    5827    5896    5986    6027    6043    6094    6104    6207    6224    6232    6281    6318
                         6380    6484    6554    6571    6579    6590    6635    6687    6789    6858    6875    6883    6937
                         7014    7104    7155    7172    7180    7234    7325    7365    7453    7501    7518    7526    7577
                         7668    7708    7796    7844    7861    7869    7907    7924    7935    8026    8066    8154    8202
                         8219    8227    8277    8292    8302    8886#   9311    9323
SVAL   = 100000          1389#
SWR      001140          1693#   3131    3153*   3155    3161*   3168*   4158    4163    5065    5761    5766    6074    6079
                         6245    6251    6602    6906    6912    6977    6982    7193    7199    7537    7542    7547    7552
                         7880    7885    7890    7895    8238    8243    8248    8253    8455    8738    8748    8758    8768
                         9271    9292    9297    9299    9392    9400    9405    9425    9483    9497    9499    9505    9512
```

```
                        9550    9557    9569    9573    9910    9949   10004*  11583
SWREG   000176          1485#   3161    8455    9910    9949    9972
SWTST   050036          4115    5043    5711    6023    6198    6546    6894    6971    7147    7495    7838    8196    9290#
SW0    =  000001        1244#
SW00   =  000001        1234#   1244
SW01   =  000002        1233#   1243
SW02   =  000004        1232#   1242
SW03   =  000010        1231#   1241
SW04   =  000020        1230#   1240
SW05   =  000040        1229#   1239
SW06   =  000100        1228#   1238
SW07   =  000200        1227#   1237
SW08   =  000400        1226#   1236
SW09   =  001000        1225#   1235
SW1    =  000002        1243#
SW10   =  002000        1224#
SW11   =  004000        1223#
SW12   =  010000        1222#  11583
SW13   =  020000        1221#   9392
SW14   =  040000        1220#   9292    9405
SW15   =  100000        1219#
SW2    =  000004        1242#
SW3    =  000010        1241#
SW4    =  000020        1240#
SW5    =  000040        1239#
SW6    =  000100        1238#
SW7    =  000200        1237#
SW8    =  000400        1236#   9297
SW9    =  001000        1235#   4158    4163    5065    5761    5766    6074    6079    6245    6251    6602    6906    6912
                        6977    6982    7193    7199    7537    7542    7547    7552    7880    7885    7890    7895    8238
                        8243    8248    8253    8738    8748    8758    8768    9271    9400    9425

TBITVE=  000014         1277#
TEMP1    003372         1904#   3382*   3396*   3401*   3406*   3407*   3408    3496*   3501*   3506*   3507*   3508    3631*
                        3768*   3840*   3921*   3980*   4019*   4125*   4199*   4238*   4272*   4345*   4384*   4426*   4448*
                        4474*   4511*   4565*   4604*   4644*   4688*   4689*   4690    4707*   4714*   4761*   4764*   4796*
                        4887*   4907*   4934*   5019*   5057*   5100*   5187*   5241*   5295*   5297    5330*   5342*   5377*
                        5400*   5434*   5436    5470*   5508*   5538*   5539    5585*   5606*   5639*   5680*   5719*   5723*
                        5797*   5835*   5865*   5866    5903*   5929*   5963*   5993*   6032*   6036*   6111*   6131*   6163*
                        6213*   6217*   6295*   6390*   6410*   6437*   6522*   6560*   6564*   6594*   6655*   6701*   6726*
                        6758*   6803*   6864*   6868*   6926*   6930*   6944*   7021*   7041*   7073*   7114*   7161*   7165*
                        7246*   7262*   7294*   7334*   7373*   7390*   7422*   7462*   7507*   7511*   7589*   7605*   7637*
                        7677*   7716*   7733*   7765*   7805*   7850*   7854*   7913*   7917*   7947*   7963*   7995*   8035*
                        8074*   8091*   8123*   8163*   8208*   8212*   8281*   8285*   8309*   8352*   8359*   8555*   8570*
                        8625*   8629*   8655*   8671    8673*   8699    8701*   8703    8705*   8707    8709*   8711    8713*
                        8715    8717*   8719    8721*   8723    8725*   8727    8729*   8822    8825*   8828*   8835    8841*
                        8851*   8861*   8870*   8876*   8887*   8917    8918*   8925*   8927*   8929*   8937    8938*   8945*
                        8947*   8949*   9026*   9031*   9043*   9048*   9060*   9065*   9100*   9103*   9104    9253*   9261*
                        9327*   9344*   9360*  11343
TEMP2    003374         1905#   3949*   4208*   4354*   4574*   4729*   4983*   4988*   5061*   6283*   6321*   6322    6348*
                        6486*   6491*   6598*   8374*   8631*   9050*   9101*   9102*   9104*   9105    9331*   9353*
TEMP3    003376         1906#   4418*   4420    4466*   4468    6284*   6324    6345*   6346    7238*   7240    7581*   7583
                        7939*   7941*   7942
TEMP4    003400         1907#   4419*   4420*   4422*   4423    4467*   4468*   4470*   4471    6287*   6288    6331    6349*
                        6350    7239*   7240*   7241    7582*   7583*   7584    7940*   7941
TEMP5    003402         1908#   4184*   4301*   6289*   6290    6332*   6333    6351*   6352
TIMUP    001376         1810#   4013    4043    4070    4096    9208*   9219*   9225*
```

```
TITLE   043660          3197    3234    8446#
TKVEC = 000060          1284#   9887*   9888*
TOCYL   001352          1797#   4329*   4416    4417    4432    4434    4486    4488    4520*   4521    4528*   5460*   5462
                        5579*   5787*   5789    5899*   6097*   6108    6186    6190    6193*   6210    6227    6229*   6557
                        6574    6576*   6640*   6642    6679*   6846*   6861    6878    6880*   6923*   6966    7008*   7018
                        7096    7109    7136    7143*   7158    7175    7177*   7228*   7239    7243    7317    7329    7356
                        7490    7492*   7504    7521    7523*   7571*   7582    7586    7660    7672    7699    7833    7835*
                        7847    7864    7866*   7929*   7940    7944    8018    8030    8057    8191    8193*   8205    8222
                        8224*   8305*   8306    8307    8340    8343   11334   11346   11349
TPVEC = 000064          1285#
TRAPPC  001334          1787#   9421*  11355
TRAPVE= 000034          1283#   3139*   3140*   9459*
TRTVEC= 000014          1278#
TSTATN  044460          3634    3931    3983    4021    4048    4075    4101    4128    4241    4275    4387    4429    4451
                        4477    4514    4607    4647    4717    4799    4890    4910    4977    5022    5642    5966    6166
                        6393    6413    6480    6525    6761    7076    7297    7425    7640    7768    7998    8126    8312
                        8362    8606#   9363
TST1    010120          3268    3290#
TST10   012170          3718    3752#
TST11   012274          3759    3773    3782#
TST12   012402          3806    3816#
TST13   012570          3853    3868#
TST14   012734          3900    3913#
TST15   014430          4116    4179#
TST16   015240          4305    4317#
TST17   016530          4527    4546#
TST2    010274          3318    3339#
TST20   017572          4743#
TST21   020152          4820    4826#
TST22   020330          4873#
TST23   021422          5044    5081#
TST24   021726          5162#
TST25   022622          5306    5318#
TST26   023420          5449#
TST27   024110          5573#
TST3    011034          3386    3486#
TST30   025220          5712    5776#
TST31   025710          5892#
TST32   027036          6024    6090#
TST33   027760          6199    6277#
TST34   030424          6337    6356    6376#
TST35   031646          6547    6617#
TST36   033750          6895    6972    6998#
TST37   035044          7148    7223#
TST4    011362          3516    3580#
TST40   036770          7496    7561#
TST41   040726          7839    7902#
TST42   042744          8197    8273#
TST5    011514          362[illegible]
TST6    011646          3663#
TST7    012032          3701    3713#
TYPDS = 104405          8414   10396#
TYPE  = 104401          3201    3216    3218    3220    3235    3275    3348    3358    3387    3416    3457    3588    3602
                        3610    3722    3743    3744    4323    4326    5106    5118    5194    5247    5259    5337    5383
                        5395    5477    5514    5525    5686    5698    5804    5841    5852    5999    6011    6357    6361
                        6362    6624    6638    6662    6810    6950    6962    7002    7005    7120    7132    7340    7352
```

```
                    7468   7480   7683   7695   7811   7823   8041   8053   8169   8181   8412   8415   8449
                    8499   8680   9318   9320   9373   9377   9394   9436   9461   9552   9560   9628   9722
                    9848   9904   9916   9970   9971   9974   9985   9995  10006  10025  10073  10079  10084
                   10088  10093  10094  10096  10099  10103  10169  10171  10311  10392* 11535  11536  11540
                   11541  11550  11559  11563  11568  11570  11576  11579
TYPERR  067610      9559  11525*
TYPOC = 104402      9973  10393* 11556
TYPON = 104404     10395*
TYPOS = 104403      3361   3419   3460   3606   8683   9397   9439  10394*
T.A2  = 000001      1935*  3667   3963   3997   4221   4255   4286   4367   4401   4587   4621   4671   4782
                    4813   4924   5002   5036   5131   5207   5272   5418   5490   5594   5625   5656   5743
                    5817   5919   5946   5980   6058   6120   6149   6180   6427   6505   6539   6710   6744
                    6775   6823   7030   7059   7090   7255   7280   7311   7382   7408   7439   7598   7623
                    7654   7725   7751   7782   7956   7981   8012   8083   8109   8140   8335   8772
T.B2  = 000002      1936*  3687   3963   3997   4221   4255   4286   4367   4401   4587   4621   4671   4782
                    4813   4924   5002   5036   5131   5207   5272   5418   5490   5594   5625   5656   5743
                    5817   5946   5980   6058   6120   6149   6180   6427   6505   6539   6710   6744   6775
                    6823   7030   7059   7090   7255   7280   7311   7382   7408   7439   7598   7623   7654
                    7725   7751   7782   7956   7981   8012   8083   8109   8140   8335   8776
T.B3  = 000004      1937*  3687   4221   4367   4587   4671   8780
TI      001412      1820*  4208   4354   4574   4729   8374   9353
TIO     001414      1821*  3382   3396   3401   3496   3501   3631   3768   3840   3921   3980   4125   4199
                    4238   4272   4345   4384   4426   4448   4474   4511   4565   4604   4644   4707   4714
                    4796   4887   4907   4934   4983   4988   5019   5057   5342   5400   5585   5639   5719
                    5903   5963   6032   6111   6163   6213   6390   6410   6437   6486   6491   6522   6560
                    6594   6701   6758   6864   6926   7021   7073   7161   7246   7294   7373   7422   7507
                    7589   7637   7716   7765   7850   7913   7947   7995   8074   8123   8208   8281   8352
                    8359   8825   8841   8851   8861   8870   8887   9327   9344   9360
T100    001416      1822*  3949   5061   6598   9331
T2500   001422      1824*  5606   6131
T500    001420      1823*  5508   5835   8938
T5000   001424      1825*  5723   6036   6217   6564   6868   6930   7165   7262   7390   7511   7605   7733
                    7854   7917   7963   8091   8212   8285   8918
T50000  001426      1826*  5100   5187   5241   5330   5377   5470   5680   5797   5993   6295   6655   6803
                    6944   7114   7334   7462   7677   7805   8035   8163
UFE   = 000400      1347*  3439   3532   3788   3805
UNLD    003316      1866*  3244*  3625*  4114*  4119*  4882*  5042*  6385*  6545*  9314
UNLOAD= 000007      1315*  3630   4124
UNS   = 040000      1372*
UPE   = 020000      1352*
VV    = 000100      1383*
WCE   = 040000      1353*
WDCNT   001442      1833*  5292*  5300*  5301   5433*  5439*  5440   5533*  5543*  5544   5860*  5870*  5871
                   11343
WLE   = 004000      1369*
WORD    001452      1839*
WRDATA= 000023      1321*
WRHEAD= 000027      1323*  5186   5329   5469   5796   6654   6802
WRL   = 004000      1386*
WRTCHK= 000031      1324*
WRTGAT= 040000      1404*
$APTHD  001000      1519   1525*
$ASTAT= ****** U    9765   9780
$ATYC   052076      9736   9738*
$ATY1   052052      9734*
$ATY3   052060      9613   9735*
```

```
$ATY4   052070          9565    9737#
$AUTOB  001134          1690#   8459*   9963   10120
$BASE   001264          1760#   3239*   3300    8517*   8519*
$BDADR  001122          1685#
$BDDAT  001126          1687#
$BELL   001200          1711#   9552    9585    9916   10113
$CDW1   001270          1762#
$CDW2   001272          1763#
$CHARC  051622          9630*   9640*   9647    9656*   9661#
$CKSWR  053044          9949#  10400
$CMTAG  001100          1673#   3128    3129    3137    3143    3144    3145
$CM3    =  000000       1703#
$CM4    =  000006       1703#   1704#   1705#   1706#   1707#   1708#   1709#
$CNTLC  053754          9904    9985   10113#
$CNTLG  053766          9970   10115#
$CNTLU  053761          9995   10088   10114#
$CPUOP  001236          1734#
$CRLF   001205          1713#   3358    3416    3610    3743    3744    6361    6362    9560    9585    9629    9664   10006
                       10093   10113   10174   11535   11540   11550   11563   11576
$DBLK   052042          9688    9722    9730#
$DB20   054332         10226   10244#
$DB20   054156          6359   10185#
$DDW0   001274          1764#
$DDW1   001276          1765#
$DDW2   001300          1766#
$DDW3   001302          1767#
$DDW4   001304          1768#
$DDW5   001306          1769#
$DDW6   001310          1770#
$DDW7   001312          1771#
$DECVL  054512         10246   10292#
$DEVCT  001220          1725#   3188*   3248*   3592    8397*   8398
$DEVM   001266          1761#   3352
$DOAGN  043614          8408    8417    8423#
$DTBL   052032          9691    9726#
$ENDAD  043604          1505    8419#   9580
$ENDCT  043552          3143    8410#
$ENDMG  043623          8412    8427#
$ENULL  043620          8415    8426#
$ENV    001230          1730#   8453    9562    9608    9743    9767
$ENVM   001231          1731#   3166    3344    9610    9615    9745
$EOP    043470          4113    5107    5119    5195    5248    5260    5338    5384    5396    5478    5515    5526    5687
                        5699    5805    5842    5853    6000    6012    6663    6811    6951    6963    7121    7133    7341
                        7353    7469    7481    7684    7696    7812    7824    8042    8054    8170    8182    8381    8393#
                        9375   11588
$EOPCT  043544          3143*   8407#   8411    9374*
$EOPI   043516          3322    3589    8399    8401#   9416
$ERFLG  001103          1676#   9273    9472    9501    9503    9509*   9531    9547*   9585   11585
$ERMAX  001115          1682#   3146*   9503    9526*   9531
$ERROR  051144          3137    9545#
$ERRPC  001116          1683#   9554*   9555*   9556    9585   11328   11331   11334   11337   11340   11343   11346   11349
                       11352   11356   11359   11366   11374
$ERRTB  003506          1979#  11532
$ERTTL  001112          1680#   9553*   9585
$ESCAP  001176          1710#   3145*   3930*   3947*   4103*   4106*   4148*   4832*   4860*   4964*   4991*   5049*   5582*
                        5605*   5675*   5753*   5901*   5928*   5989*   6067*   6107*   6130*   6235*   6467*   6494*   6582*
```

```
                    6692*  6694*  6722*  6725*  6792*  6886*  6898*  7017*  7040*  7107*  7183*  7237*  7261*
                    7328*  7368*  7389*  7456*  7529*  7580*  7604*  7671*  7711*  7732*  7799*  7872*  7938*
                    7962*  8029*  8069*  8090*  8157*  8230*  8730   8734*  8744*  8754*  8764*  8785*  8789*
                    9290*  9420*  9424*  9525*  9576   9578   9585
$ETABL  001230      1729#
$ETEND  001314      1531   1772#
$FATAL  001212      1722#  9771*
$FFLG   052316      9734*  9737*  9765   9774*  9782#
$FILLC  001156      1701#  9633   9664
$FILLS  001155      1700#  9664
$GDADR  001120      1684#
$GDDAT  001124      1686#
$GET42  043574      8416#
$GTSWR  053134      9971#  10398
$HD   = 000000      1150
$HIBTS  001000      1526#
$HIOCT  054154      10162* 10173#
$ICNT   001104      1677#  9516*  9517   9519*  9530
$INTAG  001135      1691#  9968*  9987   10007  10120
$ITEMB  001114      1681#  9556*  9564   9585   11526
$LF     001206      1714#  9585   9664   10103  10113  10174
$LFLG   052315      9775*  9781#
$LPADR  001106      1678#  3147*  4152   5050   5757   6071   6239   6586   6890   6902   7187   7533   7876
                    8234   9403   9507*  9523*  9528   9530
$LPERR  001110      1679#  3148*  4151   5756   6070   6238   6585   6889   6901   7186   7532   7875   8233
                    8790   9275   9278*  9430   9507   9524*  9530   9575
$MADR1  001242      1747#
$MADR2  001246      1751#
$MADR3  001252      1754#
$MADR4  001256      1757#
$MAIL   001210      1527   1531   1720#  3165   8453   9522   9562   9608
$MAMS1  001240      1741#
$MAMS2  001244      1749#
$MAMS3  001250      1752#
$MAMS4  001254      1755#
$MBADR  001002      1527#
$MFLG   052314      9735*  9741   9776*  9780#
$MNEW   054004      9974   10118#
$MSGAD  001224      1727#  9751*  9754
$MSGLG  001226      1728#  9756*
$MSGTY  001210      1721#  9749   9757*  9769   9773*
$MSWR   053773      9971   10116#
$MTYP1  001241      1742#
$MTYP2  001245      1750#
$MTYP3  001251      1753#
$MTYP4  001255      1756#
$MXCNT  051142      9520   9530#
$NULL   001154      1699#  9635   9664
$NWTST= 000001      3280#  3282   3325#  3327   3471#  3473   3571#  3573   3612#  3614   3658#  3660   3706#
                    3708   3746#  3748   3776#  3778   3809#  3811   3856#  3858   3903#  3905   4168#  4170
                    4308#  4310   4533#  4535   4734#  4736   4823#  4862#  4864   5072#  5074   5150#  5152
                    5309#  5311   5446#  5553#  5555   5773#  5879#  5881   6082#  6084   6255#  6257   6364#
                    6366   6608#  6610   6991#  6993   7206#  7208   7558#  7899#  8260#  8262
$OCNT   052542      9820*  9849*  9862#
$OCTVL  054260      3726   10187  10212#
$OMODE  052544      9815*  9819*  9824   9827*  9838*  9864#
```

```
$OVER   051126          9484    9500    9508    9518    9527#
$PASS   001216          1724#   3165*   3717    6355    8404*   8405*   8413    8426    9514    9531
$PASTM  001006          1529#
$PWRCT  050662          9454*   9455*   9465#
$PWRDN  050576          3141    9448#   9457
$PWRUP  050610          9448    9454#
$QUES   001204          1712#   9585    9664   10025   10096   10113   10171   10174
$RDCHR  053406         10038#  10401
$RDDEC= ****** U        10404
$RDLIN  053476         10061#  10402
$RDOCT  054016         10135#  10403
$RDSZ = 000022         10054#
$RESRE  054624         10349#  10405
$RTNAD  043616          9425#
$R2A  = ****** U        10406
$SAVRE  054566         10333#  10404
$SB2D   054276         10224#
$SCOPE  050664          3135    9481#
$SETUP= 000137          3075#   3134    3135    3137    3139    3141    3143    3144    3145    3147    8402    8450    9482
                        9546    9572    9580    9908    9913    9914    9944   10120
$STUP = 177777          3075#
$SUPRS  054526          3742    6360   10303#
$SVLAD  051072          9492    9521#
$SVPC = 000274          1503#   1508
$SWR  = 167400          1135#   1150    1156    1157    1158    1159    1160    1161    1162    1163    1709    1710    1711
                        3144    3145    3147    3148    3291    3340    3487    3581    3622    3666    3714    3753    3783
                        3817    3869    3914    4180    4318    4547    4744    4827    4874    5082    5163    5319    5450
                        5574    5777    5893    6091    6278    6377    6618    6999    7224    7562    7903    8274    8390
                        8403    8418    8424    8426    9473    9474    9475    9476    9477    9483    9495    9497    9498
                        9501    9502    9503    9510    9511    9512    9524    9527    9530    9537    9538    9539    9540
                        9541    9550    9557    9569    9573    9585
$SWREG  001232          1732#   3168
$SWRMK= 000000          1163    1164    9477    9478    9499
$TESTN  001214          1723#   3583*   9522*  11328   11331   11334   11337   11340   11343   11346   11349   11352   11355
                       11356   11359   11366   11374
$TIMES  001174          1709#   3144*   3291*   3340*   3487*   3581*   3622*   3666*   3714*   3753*   3783*   3817*   3869*
                        3914*   4180*   4318*   4547*   4744*   4827*   4874*   5082*   5163*   5319*   5450*   5574*   5777*
                        5893*   6091*   6278*   6377*   6618*   6999*   7224*   7562*   7903*   8274*   8403*   9510*   9517
                        9520*   9530
$TKB    001146          1696#   9868    9889    9900    9925    9953    9980
$TKCNT  052546          9869#   9884*   9914    9931*  10045   10047*
$TKINT  052556          3111    9884#   9905    9966
$TKQEN= 052555          9873#   9939   10050
$TKQIN  052550          9870#   9885*   9886    9937*   9938*   9939    9941*
$TKQOU  052552          9871#   9886*  10048   10049*  10050   10052*
$TKQSR  052554          9872#   9885    9941   10052
$TKS    001144          1695#   9868    9890*   9921*   9923    9929*   9951    9967*   9977    9989*  10009*
$TKSRV  052626          9887    9900#
$TMP0   001160          1703#   3640*   3646*   4134*   4140*
$TMP1   001162          1704#
$TMP2   001164          1705#
$TMP3   001166          1706#
$TMP4   001170          1707#
$TMP5   001172          1708#   8730*   8785    8789
$TN   = 000043          1136#   1150    3268    3280    3291#   3318    3325    3340#   3386    3471    3487#   3516    3571
                        3581#   3583    3584    3612    3622#   3658    3666#   3701    3706    3714#   3718    3746    3753#
```

```
                    3759    3773    3776    3783#   3806    3809    3817#   3853    3856    3869#   3900    3903    3914#
                    4116    4168    4180#   4305    4308    4318#   4527    4533    4547#   4734    4744#   4820    4823
                    4827#   4862    4874#   5044    5072    5082#   5150    5163#   5306    5309    5319#   5446    5450#
                    5553    5574#   5712    5773    5777#   5879    5893#   6024    6082    6091#   6199    6255    6278#
                    6337    6356    6364    6377#   6547    6608    6618#   6895    6972    6991    6999#   7148    7206
                    7224#   7496    7558    7562#   7839    7899    7903#   8197    8260    8274#
$TNPWR  054442     10251   10252   10272#
$TPB    001152      1698#   9653*   9664
$TPFLG  001157      1702#   9602    9664
$TPS    001150      1697#   9651    9664
$TRAP   054662      3139   10369#
$TRAP2  054704     10380#  10391
$TRP    = 000016   10384#  10393#  10394#  10395#  10396#  10397#  10398   10399#  10400   10401#  10402#  10403#  10404#
                   10405#  10406#  10407#
$TRPAD  054716     10374   10391#
$TSTM   001004      1528#
$TSTNM  001102      1675#   3584*   8402*   9299    9472    9499    9521*   9522    9527    9531    9549    9585
$TTYIN  053732     10063   10064   10076   10094   10108   10112#
$TYPBN= ****** U   10397
$TYPDS  051626      9676#  10396
$TYPE   051344      9602#   9762   10384   10392
$TYPEC  051556      9632    9639    9646    9651#   9652   10011
$TYPEX  051624      9657    9659    9662#
$TYPOC  052344      9818#  10393
$TYPON  052360      9817    9820#  10395
$TYPOS  052320      9813#  10394
$UNIT   001222      1726#   3189*   3249*   3594*   3599    3603    3978    4195    4236    4341    4382    4446    4509
                    4561    4602    4705    4712    4794    4905    5017    5637    5961    6161    6408    6520    6756
                    7071    7292    7420    7635    7763    7993    8121    8350    8357    8607    8626    8652    8694
                    8695    8696    8697    8823    8838    8848    8858    8868    8890    9340    9358
$UNITM  001010      1530#
$USWR   001234      1733#
$VECT1  001260      1758#
$VECT2  001262      1759#
$XTSTR  050676      9486#
$$GET4= 000000      8418#
$OFILL  052543      9814*   9818*   9828    9863#
$40CAT= ****** U    9483    9559
.       = 070067    1479#   1483#   1488#   1490#   1492#   1494#   1496#   1503    1504#   1506#   1508#   1509#   1515
                    1516#   1518#   1520#   1672#   1715    1852#   1853#   1854#   1856#   3132    3147    3148    8426
                    8430    9450    9530    9531    9585    9664    9730#   9783#   9868    9872#   9873    9874#  10112#
                   10113   10120#  10174   10212#  10292#  10411#  11327#
.$ASTA= ****** U    9735    9738
.$X     = 001000    1515#   1520
```

```
CALIB   1579#  4193  4339  4559  9338
CHECK   1554#  3685  3938  3961  3995  4028  4055  4082  4219  4253  4284  4365  4399  4585  4619
        4668  4780  4811  4896  4922  4969  5000  5034  5129  5205  5270  5358  5416  5488  5592
        5623  5654  5741  5815  5917  5944  5978  6056  6118  6147  6178  6310  6399  6425  6472
        6503  6537  6708  6742  6773  6821  7028  7057  7088  7253  7278  7309  7380  7406  7437
        7596  7621  7652  7723  7749  7780  7954  7979  8010  8081  8107  8138  8333
CIDAE   1583#  4702  8347
COMMEN  1287#
CWD2    1559#  3692  3968  4226  4372  4592  4676  5007  5137  5213  5278  5423  5951  6510
DRCLR   1571#  3976  4234  4380  4443  4506  4600  4710  4791  4903  5015  5634  5959  6158  6406
        6518  6753  7068  7289  7417  7632  7760  7990  8118  8355  9356
ENDCOM  1287#
EOPGM   1664#  8394
ERROR   1181#  3321  3377  3384  3398  3403  3425  3433  3448  3452  3463  3467  3498  3503  3519
        3527  3545  3549  3553  3557  3628  3633  3635  3638  3648  3670  3688  3689  3690  3691
        3695  3698  3702  3720  3765  3770  3774  3787  3790  3797  3804  3807  3821  3824  3827
        3830  3837  3842  3845  3848  3851  3854  3873  3876  3879  3886  3892  3895  3898  3901
        3918  3923  3928  3933  3941  3942  3943  3944  3951  3964  3965  3966  3967  3971  3974
        3982  3985  3998  3999  4000  4001  4015  4023  4031  4032  4033  4034  4045  4050  4058
        4059  4060  4061  4072  4077  4085  4086  4087  4088  4098  4102  4112  4122  4127  4129
        4132  4142  4146  4191  4201  4207  4210  4222  4223  4224  4225  4229  4232  4240  4243
        4256  4257  4258  4259  4266  4274  4276  4287  4288  4289  4290  4294  4298  4337  4347
        4353  4356  4368  4369  4370  4371  4375  4378  4386  4389  4402  4403  4404  4405  4412
        4428  4430  4436  4441  4450  4453  4461  4476  4478  4484  4493  4499  4504  4513  4516
        4556  4567  4573  4576  4588  4589  4590  4591  4595  4598  4606  4609  4622  4623  4624
        4625  4630  4637  4646  4648  4672  4673  4674  4675  4679  4682  4694  4709  4716  4719
        4726  4731  4748  4752  4758  4763  4767  4770  4783  4784  4785  4786  4789  4798  4801
        4814  4815  4816  4817  4821  4831  4837  4841  4851  4855  4878  4889  4891  4899  4900
        4901  4902  4909  4912  4925  4926  4927  4928  4936  4940  4945  4972  4973  4974  4975
        4978  4982  4985  4987  4990  5003  5004  5005  5006  5010  5013  5021  5024  5037  5038
        5039  5040  5054  5059  5063  5092  5102  5105  5117  5132  5133  5134  5135  5140  5143
        5173  5189  5193  5208  5209  5210  5211  5216  5219  5231  5243  5246  5258  5273  5274
        5275  5276  5281  5284  5299  5323  5332  5336  5344  5347  5361  5362  5363  5364  5372
        5379  5382  5394  5402  5405  5419  5420  5421  5422  5426  5429  5438  5454  5472  5476
        5491  5492  5493  5494  5501  5510  5513  5524  5541  5577  5587  5595  5596  5597  5598
        5602  5610  5613  5626  5627  5628  5629  5632  5641  5644  5657  5658  5659  5660  5665
        5673  5682  5685  5697  5705  5709  5716  5721  5725  5728  5731  5744  5745  5746  5747
        5750  5781  5799  5803  5818  5819  5820  5821  5828  5837  5840  5851  5868  5897  5905
        5911  5920  5921  5922  5923  5926  5931  5934  5947  5948  5949  5950  5954  5957  5965
        5968  5981  5982  5983  5984  5987  5995  5998  6010  6017  6021  6028  6034  6038  6041
        6044  6059  6060  6061  6062  6065  6095  6105  6113  6121  6122  6123  6124  6128  6134
        6137  6150  6151  6152  6153  6156  6165  6168  6181  6182  6183  6184  6188  6208  6215
        6219  6222  6225  6233  6282  6297  6301  6313  6314  6315  6316  6319  6326  6328  6341
        6381  6392  6394  6402  6403  6404  6405  6412  6415  6428  6429  6430  6431  6439  6443
        6448  6475  6476  6477  6478  6481  6485  6488  6490  6493  6506  6507  6508  6509  6513
        6516  6524  6527  6540  6541  6542  6543  6555  6562  6566  6569  6572  6580  6591  6596
        6600  6628  6632  6636  6657  6661  6688  6703  6711  6712  6713  6714  6718  6729  6732
        6745  6746  6747  6748  6751  6760  6763  6776  6777  6778  6779  6783  6790  6805  6809
        6824  6825  6826  6827  6859  6866  6870  6873  6876  6884  6928  6932  6935  6938  6946
        6949  6961  6968  7015  7023  7031  7032  7033  7034  7038  7044  7047  7060  7061  7062
        7063  7066  7075  7078  7091  7092  7093  7094  7098  7105  7116  7119  7131  7138  7156
        7163  7167  7170  7173  7181  7235  7248  7256  7257  7258  7259  7265  7268  7281  7282
        7283  7284  7287  7296  7299  7312  7313  7314  7315  7319  7326  7336  7339  7351  7358
        7366  7375  7383  7384  7385  7386  7393  7396  7409  7410  7411  7412  7415  7424  7427
        7440  7441  7442  7443  7447  7454  7464  7467  7479  7486  7502  7509  7513  7516  7519
        7527  7578  7591  7599  7600  7601  7602  7608  7611  7624  7625  7626  7627  7630  7639
```

```
        7642    7655    7656    7657    7658    7662    7669    7679    7682    7694    7701    7709    7718    7726    7727
        7728    7729    7736    7739    7752    7753    7754    7755    7758    7767    7770    7783    7784    7785    7786
        7790    7797    7807    7810    7822    7829    7845    7852    7856    7859    7862    7870    7908    7915    7919
        7922    7925    7936    7949    7957    7958    7959    7960    7966    7969    7982    7983    7984    7985    7988
        7997    8000    8013    8014    8015    8016    8020    8027    8037    8040    8052    8059    8067    8076    8084
        8085    8086    8087    8094    8097    8110    8111    8112    8113    8116    8125    8128    8141    8142    8143
        8144    8148    8155    8165    8168    8180    8187    8203    8210    8214    8217    8220    8228    8278    8283
        8287    8290    8293    8303    8311    8313    8317    8320    8323    8326    8336    8337    8338    8339    8342
        8345    8354    8361    8364    8371    8376    8804    8809    8813    8827    8843    8853    8863    8872    8889
        9312    9324    9329    9333    9346    9352    9355    9362    9365    9422
ESCAPE  1287#
FSECA   1635#   8912
FSECB   1639#   8932
F.EAB   1545#   3952    3987    4211    4245    4357    4391    4577    4611    4771    4803    4914    4992    5026    5121
        5197    5262    5349    5407    5480    5614    5646    5733    5807    5935    5970    6047    6138    6170    6302
        6417    6495    6529    6733    6765    6813    7048    7080    7269    7301    7397    7429    7612    7644    7740
        7772    7970    8002    8098    8130
GETPRI  1287#
GETSWR  1287#   8450
HDCHK3  1624#   6940    7110    7330    7458    7673    7801    8031    8159
HDTBL   1631#   5461    5788    6641
HEADER  1655#   5445    5772
LIMIT   1650#   4876    6379
LOOP    1539#   3391    3491    3761    3792    3832    3881    4186    4262    4333    4408    4457    4551    4632    4753
        5087    5169    5227    5367    5497    5668    5824    6100    6683    6785    7010    7100    7230    7321    7362
        7449    7573    7664    7705    7792    7931    8022    8063    8150    8298
LPCHK   1561#   4148    5752    6066    6235    6582    6886    6898    7183    7529    7872    8230
MSG     3280#   3282    3325#   3327    3471#   3473    3571#   3573    3612#   3614    3658#   3660    3706#   3708    3746#
        3748    3776#   3778    3809#   3811    3856#   3858    3903#   3905    4168#   4170    4308#   4310    4533#   4535
        4734#   4736    4862#   4864    5072#   5074    5149#   5152    5309#   5311    5552#   5555    5879#   5881    6082#
        6084    6255#   6257    6364#   6366    6608#   6610    6991#   6993    7206#   7208    8260#   8262
MULT    1287#
NEWTST  1287#   3280    3325    3471    3571    3612    3658    3706    3746    3776    3809    3856    3903    4168    4308
        4533    4734    4823    4862    5072    5150    5309    5446    5553    5773    5879    6082    6255    6364    6608
        6991    7206    7558    7899    8260
OWNTAG  1533#   1774
POP     1287#   9717    9777    9778    10163   10354
PUSH    1287#   9676    9738    9740    9761    10137   10334
QKRPSK  1602#   6205    6552    6857    7153    7499    7842    8200
QKSEEK  1596#   5717    6030    6211    6558    6862    6924    7159    7505    7848    7911    8206    8279
QKSRT   1592#   5051    6588    9321
QKUNLD  1608#   3626    4120
RALLHD  1625#   5504    5831
RDHDR   1618#   5097    5238    5374    5677    5990    6941    7111    7331    7459    7674    7802    8032    8160
REPORT  1287#
SCOPE   1182#   3290    3339    3486    3580    3621    3665    3713    3752    3782    3816    3868    3913    4179    4317
        4546    4743    4826    4873    5081    5162    5318    5449    5573    5776    5892    6090    6277    6376    6617
        6998    7223    7561    7902    8273    8395    8401
SECTST  1644#   4835
SETPRI  1287#   10041
SETTRA  10384#  10393   10394   10395   10396   10398   10400   10401   10402   10403   10404   10405   10406
SETUP   1287#   3127
SKATN   1588#   4765    5608    6132    6727    7042    7263    7391    7606    7734    7964    8092
SKIP    1287#   3268    3318    3386    3516    3701    3718    3759    3773    3806    3853    3900    4116    4305    4527
        4820    5044    5306    5712    6024    6199    6337    6356    6547    6895    6972    7148    7496    7839    8197
SKOSC   1663#   7226    7569    7927
```

F04


```
SKRDY    1587#  5583   6109   6699   7019   7244   7371   7587   7714   7945   8072
SLASH    1287#
SPACE    1287#
STARS    1287#  1501   1512   1514   1521   1668   1715   1718   3280   3289   3325   3338   3471   3485   3571
         3579   3612   3620   3658   3664   3706   3712   3746   3751   3776   3781   3809   3815   3856   3867
         3903   3912   4168   4178   4308   4316   4533   4545   4734   4742   4823   4825   4862   4872   5072
         5080   5150   5161   5309   5317   5446   5448   5553   5572   5773   5775   5879   5891   6082   6089
         6255   6276   6364   6375   6608   6616   6991   6997   7206   7222   7558   7560   7899   7901   8260
         8272   8387   9469   9533   9587   9666   9733   9790   9867   9944   9959   10030  10054  10123  10176
        10215  10233  10295  10318  10363  11515  11524
SWRSU    1287#  3149#
SW814    1562#  4115   5043   5710   6022   6197   6546   6893   6971   7146   7494   7837   8195
TRMTRP  10384#
TSTSW9   1566#  4157   4162   5064   5759   5764   6072   6077   6243   6249   6601   6904   6910   6975   6980
         7191   7197   7535   7540   7545   7550   7878   7883   7888   7893   8236   8241   8246   8251
TYPBIN   1287#
TYPDEC   1287#  8413
TYPNAM   1287#
TYPNUM   1287#
TYPOCS   1287#  3359   3417   3458   3604   8681   9395   9437
TYPOCT   1287#  9972
TYPTXT   1287#
WRHDR    1613#  5185   5328   5468   5795   6653   6801
$$CMRE   1666#
$$CMTM   1666#  1703   1704   1705   1706   1707   1708
$$ESCA   1287#
$$NEWT   1287#  3280   3325   3471   3571   3612   3658   3706   3746   3776   3809   3856   3903   4168   4308
         4533   4734   4823   4862   5072   5150   5309   5446   5553   5773   5879   6082   6255   6364   6608
         6991   7206   7558   7899   8260
$$SET   10384#  10393  10394  10395  10396  10398  10400  10401  10402  10403  10404  10405  10406
$$SETM   3165#
$$SKIP   1287#  3268   3318   3386   3516   3701   3718   3759   3773   3806   3853   3900   4116   4305   4527
         4820   5044   5306   5712   6024   6199   6337   6356   6547   6895   6972   7148   7496   7839   8197
.EQUAT   1135#  1177
.HEADE   1135#  1140
.SETUP   1135#  3075
.SWRHI   1135#  1152
.SWRLO   1135#  1164#
.$ACTI   1135#  1499
.$APTB   1716#
.$APTH   1135#  1510
.$APTY   1135#  9731
.$CATC   1135#  1477
.$CMTA   1135#  1666
.$DB2D   1135#  10231
.$DB2O   1135#  10174
.$EOP    1135#  8385
.$ERRO   1135#  9531
.$RDOC   1135#  10121
.$READ   1135#  9865
.$SAVE   1135#  10316
.$SB2D   1135#  10213
.$SCOP   1135#  9467
.$SUPR   1135#  10293
.$TRAP   1135#  10361
.$TYPD   1135#  9664
```

```
.$TYPE   1135#   9585
.$TYPO   1135#   9788


. ABS.  070067     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZR6HC,DZR6HC.SEQ/SOL/NL:MD/EQ:SDC/CRF/NL:TOC/DOC=DZR6HC.P11
 RUN-TIME:  107  104  12 SECONDS
 RUN-TIME RATIO:  458/224=2.0
 CORE USED:  33K  (65 PAGES)

 DOCUMENT PAGES:  252
```

H04

Spooler runtime 39 Seconds, 161 KCS, 1007 disk reads, 0 disk writes, 252 pages

[redacted] Date 01-Dec-76 17:51:43 Monitor IPC-0 602 [600] **END**

```
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111111110
00000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111111222222222233312
```