RK611/RK06

DRIVE DIAGNOSTIC PART 2

MD-11-DZR6I-D

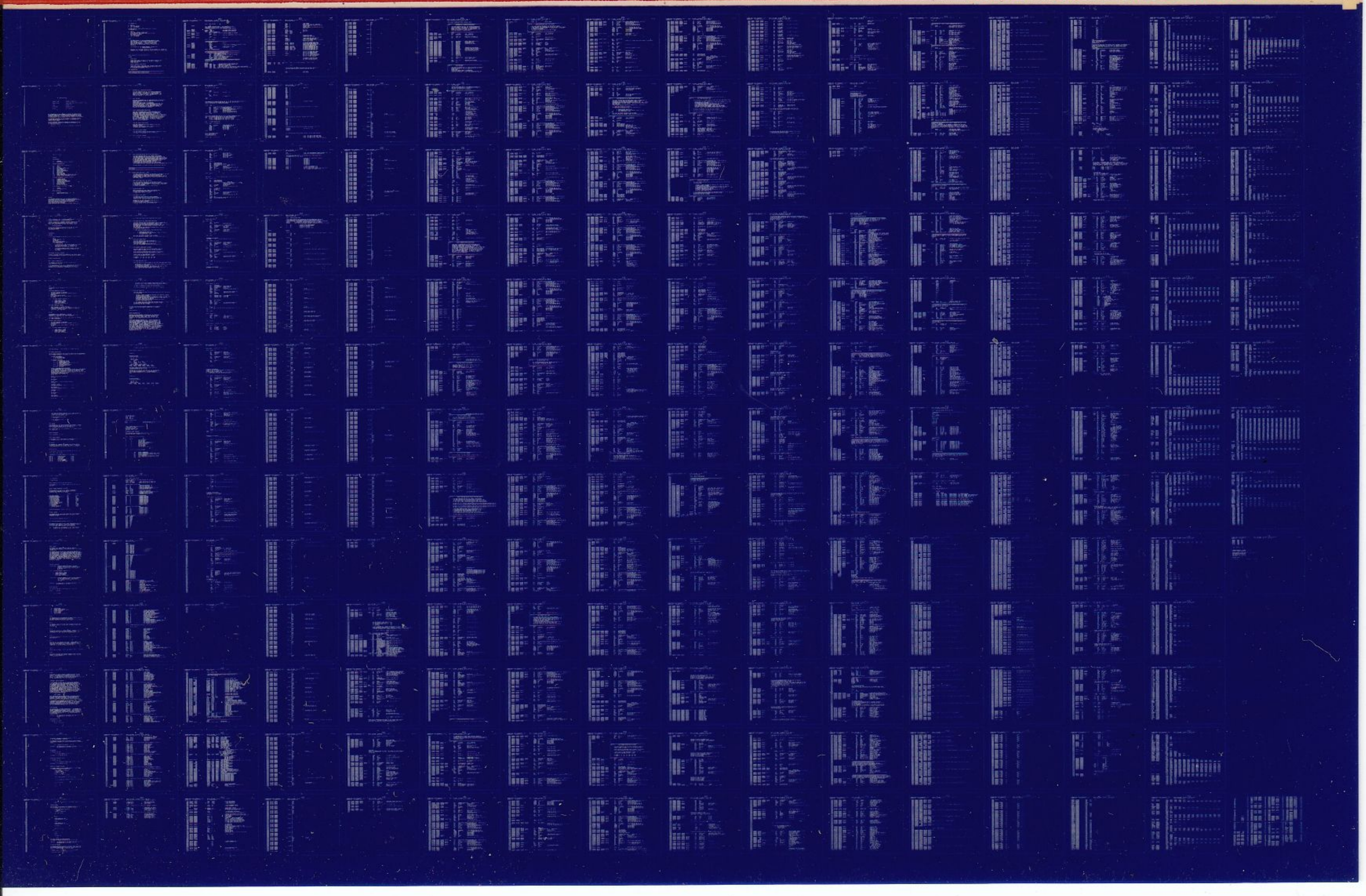
EP-DZR6I-D-DL-A

FICHE 1 OF 1

APR 1977

digital

MADE IN USA



B01

```
.REM %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6I-D-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 2
DATE:	JANUARY 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 2 OF THE DRIVE DIAGNOSTICS TO
INSURE THAT THE DISK IS CAPABLE OF PERFORMING READ AND WRITE
DATA OPERATIONS IN BOTH 20 AND 22 SECTOR FORMATS.
WORST CASE PATTERNS, SPIRAL WRITING AND READING, AND ALL
OFFSET OPERATIONS ARE PERFORMED.
ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF PART 2, THE DRIVE
IS READY FOR PART 3 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 MESSAGE AT THE BEGINNING OF THE TESTS WILL CONFIRM THIS.

2. A 22 SECTOR FORMATTED PACK IS REQ'D, BUT WILL BE A RESULT OF RUNNING DRIVE DIAGNOSTOC PART 1 (SEE BELOW).

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD
FIRST RUN SUCCESSFULLY FOLLOWED BY THE RK06 DRIVE DIAGNOSTIC- PART 1.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

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

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

3.2 XXDP

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

CHAIN MODE OPERATION (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. DRIVE 0 CAN BE TESTED, BUT THE OPERATOR IS FIRST GIVEN A MESSAGE TO REPLACE THE PACK IN DR0 WITH A SCRATCH PACK & TYPE <CR> WHEN DONE.

3.3 ACT/APT

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

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

AUTOMATIC MODE (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

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

3.3.1 APT ETABLE DEFINITIONS

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

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

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

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

4. SWITCH 2 (USER SWITCH REGISTER)
   NOT USED

5. CPU OPTIONS:
   NOT USED

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

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

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

9. INTERRUPT VECTOR 2:
   NOT USED

10. BUS PRIORITY 2:
    NOT USED

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

12. DEVICE MAP:

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

13. CONTROLLER DESCRIPTOR WORDS:
    NOT USED

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

3.4 DUAL ACCESS

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

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

3.5 MEMORY MANAGEMENT

MEMORY MANAGEMENT NOT USED

3.6 PARITY CHECK ENABLED

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

3.7 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR
INFORMATION CONTAINED ON CYLINDER 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: 2 MIN, 40 SEC

A BREAKDOWN OF THE MORE LENGTHY TESTS ARE SHOWN BELOW:

TEST 20	DRIVE OFF TRACK	45 SEC
TEST 21	UNLOAD HEADS	10 SEC
TEST 22	LOAD HEADS	20 SEC
TEST 23	ROTATIONAL TIMING	15 SEC
TEST 24	MAX SEEK TIMES	15 SEC
TEST 26	137 CYL SEEK TIMES	10 SEC
TEST 27	MAX VELOCITY TIMES	15 SEC

3.9 FAULT ISOLATION

TO BE DETERMINED.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS

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

3.11 DEFAULT UNIBUS ADDRESSES & VECTORS

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

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

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

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

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

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

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

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

POSITION.

4.1.4 PRESS 'START'

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

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

4.2 STARTING LOCATIONS

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

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

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 MESSAGE AND PERTINENT INFORMATION.
                        PRESSING "CONTINUE" CONTINUES 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 MESSAGES.  NORMALLY USED WHEN
                        LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).
                        WITH SWITCH <13> SET, SWITCH <15> SHOULD NOT BE SET.


                4.3.4   SW<12>

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


                4.3.5   SW<11>

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


                4.3.6   SW<10>

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


                4.3.7   SW<09>

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

4.3.8 SW<08>

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

4.4 'SOFTWARE' SWITCH REGISTER

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

SWR = NNNNNN NEW =

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

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

4.5 INPUT DIALOGUE

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

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.4.

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

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

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

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

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

4.5.2 BUS ADDRESS

THE REQUEST WILL BE:

TYPE IN BUSS ADDRESS IF NOT 177440

THE DEFAULT IS A CARRIAGE RETURN

4.5.3 CONTROLLER INTERRUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

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

```
UNIBUS RK06 DRIVE DIAGNOSTIC
   PART 2
MAINDEC-11-DZR6I-D-PB

DRIVES TO BE TESTED: 1,3<CR>

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>

WILL TEST DRIVES:
1
3

DRIVE 1
```

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

4.6 PROGRAM EXAMPLE

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

```
        UNIBUS RK06 DRIVE DIAGNOSTIC
           PART 2
        MAINDEC-11-DZR6I-D-PB

        WILL TEST DRIVES:
        0
        1

        DRIVE 0

        DRIVE SERIAL NO. AAA
        CARTRIDGE SERIAL NO. BBB

        DRIVE 1

        DRIVE SERIAL NO. CCC
        CARTRIDGE SERIAL NO. DDD

        END PASS #1


        WILL TEST DRIVES:
        0
        1

        DRIVE 0

        DRIVE 1

        END PASS # 2

           (ETC)
```

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

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

4.7 HALTING THE PROGRAM

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

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

IF HEADS ARE LOADED & FORMATTING IS VALID,

```
                        THE PROGRAM WILL:

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

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

                        1.  ECHO ↑C
                        2.  TYPE 'HALT PENDING, PLEASE WAIT'
                        3.  DO THE TEST(S) THAT LOADS HEADS AND/OR FORMATS
                            THE INVALID CYLINDERS
                        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.  WRITE TESTS

                            THESE TESTS CHECK THE ABILITY OF THE DRIVE TO WRITE & READ
                            WORSE CASE PATTERNS;  PERFORM ALL OFFSETS & PERFORM ALL
                            SPIRAL WRITING.

                        B.  SERVO & SPINDLE TIMING TESTS

                            THESE TESTS CHECK & TYPE HEAD LOAD, UNLOAD & INDEX TIMING,
                            ALSO MIN, MAX, AND AVERAGE SEEK TIMES, AND MAX VELOCITY
                            OF THE HEADS ARE MEASURED & TYPED.

                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 MESSAGE.  ANY
                                ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                                TESTS AND JUMPING TO  'END OF PASS'


                        TEST 2  SIZE THE BUSS

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


                        TEST 3  VERIFY OPERATOR DRIVE SELECTIONS

                                THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
                                DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
                                CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
                                PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
                                TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
                                BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
                                ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                                NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                                NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
                                VERIFY IT WAS NOT SPECIFIED.


                        TEST 4  FIND NEXT DRIVE TO BE TESTED

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


                        TEST 5  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 6  SET VV WITH PACK COMMAND

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

```
TEST 7  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 MESSAGE 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.


*****************************************************************
WRITE TESTS
*****************************************************************


TEST 10 BASIC WRITE DATA TEST; 1 WORD

        THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,
        ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
        A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
        PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
        FOR A WRITE ERROR.


TEST 11 BASIC WRITE DATA TEST; FULL SECTOR

        THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
        A FULL SECTOR.  ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
        & CHECKED BY A RD DATA COMMAND.  A FURTHER CHECK IS PERFORMED
        BY THE WRT CHK COMMAND.
        THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.


TEST 12 20 SECTOR FORMAT TEST

        DATA IS WRITTEN ON A FULL TRACK IN 20 SECTOR FORMAT.
        MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
        CYLINDER,TRACK, SECTOR 0 IS USED.


TEST 13 TEST OFFSET & RTC LOGIC

        THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
        THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
        'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG

        BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
        DIFFERENT CYLINDER IS ISSUED
        IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
        CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
```

```
                                        ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED


                                TEST 14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

                                        THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY
                                        WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN
                                        PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET
                                        POSITIONS UNTIL A FAILURE OCCURES. THE OFFSET POSITIONS
                                        ARE TYPED OUT.
                                        OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10.
                                        ALL HEADS ARE TESTED AT CYL 0.

                                        IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT

                                                A. HEADS DID NOT MOVE AT ALL
                                        OR      B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP
                                                   ARE EXCEPTIONALLY GOOD.

                                        NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR,
                                        BUT AN INDICATION OF SURFACE, HEAD, & R/W ELECTRONICS QUALITY ONLY


                                TEST 15 WRITE WITH HEADS OFFSET

                                        THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
                                        WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
                                        & THE DRIVE WILL WRITE
                                        SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
                                        THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY


                                TEST 16 TEST CURRENT CROSS-OVER CYLINDERS

                                        THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF
                                        CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:

                                        SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y
                                        WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.

                                        A WRITE CHECK IS THEN PREFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.
                                        THIS TEST IS PERFORMED FOR ALL 3 HEADS.

                                        CYLINDER X:  63  127  191  255  319  383
                                        CYLINDER Y:  64  128  192  256  320  384


                                TEST 17 TEST HEAD SWITCHING TIME

                                        TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.

                                        1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS
                                           TO SECTOR 25(8) IS ISSUED.
                                        2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL
                                           A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
                                        3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN
```

THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO
SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS

THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2

THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT

TEST 20 DRIVE OFF TRACK TEST

THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND
THE ALLOTED 3MS.

1. INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS
   (UNIQUE TO EACH CYLINDER)
2. A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO
   AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE
   HEADS WILL IMMEDIATELY BE WRITTEN.
3. IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS,
   DRIVE OFF TRACK ERROR WILL SET.

IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDE
100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
BEFORE DOING THE NEXT CYLINDER

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

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE 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 MESSAGE CONTAINS A PC. THIS PC INDICATES THE
POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE
ERROR MESSAGE, 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
MESSAGES ARE IN OCTAL.

NOTE

```
                                NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.


                        6.2     ERROR PRINTOUT EXAMPLES:


                                EXAMPLE #1


                                MESSAGE A0 ERROR
                                AFTER START SPINDLE CMD & FWD SET

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

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

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


                                EXAMPLE #2:


                                NO ATTN IN RKASOF
                                AFTER UNLOAD COMMAND

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


                                        [ END OF DOCUMENT ]

                        %
```

```
                                ;*** PGM REV 028 ***
                                .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 2
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY GARY PAPAZIAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*

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


                                .SBTTL  SUMMARY OF STARTING LOCATIONS
                                ;*
                                ;*
                                ;*              200     DEFAULT PARAMETERS
                                ;*              204     DEFAULT PARAMETERS & BYPASS WRITE TESTS
                                ;*              214     DEFAULT PARAMETERS & BYPASS TIMING TESTS
                                ;*              220     INPUT PARAMETERS
                                ;*              224     INPUT PARAMETERS & BYPASS WRITE TESTS
                                ;*              230     INPUT PARAMETERS & BYPASS TIMING TESTS
                                ;*              240     ODT11
                                ;*
```

```
                        .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
```

```
1149
1150                            ;       $BASE=177440
1151
1152            000000          RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1153            000002          RKWC=   2               ;WORD COUNT REGISTER
1154            000004          RKBA=   4               ;BUS ADDRESS REGISTER
1155            000006          RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1156            000010          RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1157            000012          RKDS=   12              ;DRIVE STATUS REGISTER
1158            000014          RKER=   14              ;ERROR REGISTER
1159            000016          RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1160            000020          RKDC=   20              ;DESIRED CYLINDER REGISTER
1161            000024          RKDB=   24              ;DATA BUFFER
1162            000026          RKMR1=  26              ;MAINTENANCE REGISTER 1
1163            000034          RKMR2=  34              ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1164            000036          RKMR3=  36              ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1165            000030          RKECPS= 30              ;ECC POSITION INFORMATION
1166            000032          RKECPT= 32              ;ECC PATTERN INFORMATION
1167
1168                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1169
1170                            ;       DRIVE COMMANDS
1171
1172            000001          SELDRV= 1               ;SELECT DRIVE (GET STATUS)
1173            000003          PACK=   3               ;PACK ACKNOWLEDGE
1174            000005          CLEAR=  5               ;DRIVE CLEAR
1175            000007          UNLOAD= 7               ;UNLOAD
1176            000011          SRTSPL= 11              ;START SPINDLE
1177            000013          RECAL=  13              ;RECALIBRATE
1178            000015          OFFSET= 15              ;OFFSET
1179            000017          SEEK=   17              ;SEEK
1180            000021          RDDATA= 21              ;READ DATA
1181            000023          WRDATA= 23              ;WRITE DATA
1182            000025          RDHEAD= 25              ;READ HEADER
1183            000027          WRHEAD= 27              ;WRITE HEADER AND DATA
1184            000031          WRTCHK= 31              ;WRITE CHECK
1185
1186            000001          GO=     BIT0            ;GO BIT
1187            000100          IE=     BIT6            ;INTERRUPT ENABLE
1188            000200          RDY=    BIT7            ;CONTROLLER READY
1189            000400          BA16=   BIT8            ;BUS ADDRESS BIT 16
1190            001000          BA17=   BIT9            ;BUS ADDRESS BIT 17
1191            002000          CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06)
1192            004000          CTO=    BIT11           ;CONTROLLER TIMEOUT
1193            010000          CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1194            020000          DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1195            040000          DI=     BIT14           ;DRIVE INTERRUPT
1196            100000          CERR=   BIT15           ;CONTROLLER ERROR
1197            100000          CCLR=   BIT15           ;CONTROLLER CLEAR
1198
1199                            .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1200
1201            000007          DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
1202            000010          RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1203            000020          BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
1204            000040          SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
1205            000100          IR=     BIT6                    ;INPUT READY
1206            000200          OR=     BIT7                    ;OUTPUT READY
1207            000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1208            001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1209            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1210            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1211            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1212            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1213            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1214            100000          DLT=    BIT15                   ;DATA LATE ERROR
1215
1216                            .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1217
1218            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1219            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
1220            000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
1221            000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
1222            000020          FMTE=   BIT4                    ;FORMAT ERROR
1223            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1224            000100          ECH=    BIT6                    ;ECC HARD
1225            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
1226            000400          HVRC=   BIT8                    ;HEADER VRC ERROR
1227            001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1228            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1229            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1230            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
1231            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1232            040000          UNS=    BIT14                   ;DRIVE UNSAFE
1233            100000          DCK=    BIT15                   ;DATA CHECK
1234
1235                            .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1236
1237            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1238                                                            ;    THIS BIT IS RESET)
1239            000004          OFST=   BIT2                    ;DRIVE OFFSET
1240            000010          ACLO=   BIT3                    ;AC LOW
1241            000020          DCLO=   BIT4                    ;DC LOW
1242            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
1243            000100          VV=     BIT6                    ;VOLUME VALID
1244            000200          DRDY=   BIT7                    ;DRIVE READY
1245            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
1246            004000          WRL=    BIT11                   ;WRITE LOCK
1247            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
1248            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
1249            100000          SVAL=   BIT15                   ;STATUS VALID
1250
1251                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1252
1253            000017          MESMSK= 17                      ;MESSAGE MASK
1254            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
1255            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
1256            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1257            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
1258            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
1259            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1260            002000          MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
```

```
1261    004000          PCA=     BIT11          ;PRECOMPENSATION ADVANCE
1262    010000          PCD=     BIT12          ;PRECOMPENSATION DELAY
1263    020000          ECCW=    BIT13          ;ECC WORD IS BEING READ OR WRITTEN
1264    040000          WRTGAT=  BIT14          ;WRITE GATE
1265    100000          RDGATE=  BIT15          ;READ GATE
1266
1267                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1268
1269    000040          D.DRA=   BIT5           ;DRIVE AVAILABLE
1270    000100          D.VV=    BIT6           ;VOLUME VALID
1271    000200          D.DRDY=  BIT7           ;DRIVE READY
1272    000400          D.DDT=   BIT8           ;DRIVE TYPE (0=RK06)
1273    001000          D.FORM=  BIT9           ;DRIVE FORMAT
1274    002000          D.OFF=   BIT10          ;OFFSET ON
1275    004000          D.WRL=   BIT11          ;WRITE LOCK
1276    010000          D.SPIN=  BIT12          ;SPINDLE ON
1277    020000          D.PIP=   BIT13          ;POSITIONING IN PROGRESS
1278    040000          D.DSC=   BIT14          ;DRIVE STATUS CHANGE
1279
1280                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1281
1282    000020          D.SSP=   BIT4           ;SERVO SIG PRES
1283    000040          D.HOHM=  BIT5           ;HEADS HOME
1284    000100          D.BRHM=  BIT6           ;BRUSHES HOME
1285    000200          D.DOOR=  BIT7           ;DOOR INTERLOCKED
1286    000400          D.CART=  BIT8           ;CARTRAGE INTERLOCK
1287    001000          D.SPOK=  BIT9           ;SPEED OK
1288    002000          D.FWD=   BIT10          ;FORWARD
1289    004000          D.REV=   BIT11          ;REVERSE
1290    010000          D.LOAD=  BIT12          ;HEADS LOADING
1291    020000          D.RTZ=   BIT13          ;RETURN TO ZERO
1292    040000          D.UNLD=  BIT14          ;HEADS UNLOADING
1293
1294                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1295
1296    000040          D.IDAE=  BIT5           ;INVALID DISK ADDRESS ERROR:HEAD/CYL
1297    000100          D.ACLO=  BIT6           ;AC LOW
1298    000200          D.FLT=   BIT7           ;DRIVE FAULT
1299    000400          D.ILF=   BIT8           ;ILLEGAL FUNCTION CODE
1300    001000          D.PAR=   BIT9           ;DRIVE DETECTED SERCON PARITY ERROR
1301    002000          D.SKI=   BIT10          ;SEEK INCOMPLETE
1302    004000          D.WLE=   BIT11          ;WRITE LOCK ERROR
1303    010000          D.SPLS=  BIT12          ;SPEED LOSS
1304    020000          D.DROT=  BIT13          ;DRIVE OFF TRACK
1305    040000          D.UNS=   BIT14          ;R/W UNSAFE
1306
1307                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1308
1309    000020          D.SECT=  BIT4           ;SECTOR ERROR
1310    000040          D.WCUR=  BIT5           ;WRITE CURRENT AND NO WRITE GATE
1311    000100          D.WGAT=  BIT6           ;WRITE GATE AND NO TRANSISTIONS
1312    000200          D.HDFL=  BIT7           ;HEAD FAULT
1313    000400          D.MHD=   BIT8           ;MULTIPLE HEAD SELECT
1314    001000          D.XERR=  BIT9           ;INDEX ERROR
1315    002000          D.TIB=   BIT10          ;TRIBIT ERROR
1316    004000          D.PLO=   BIT11          ;PLO ERROR
```

```
1317    010000          D.NMOV= BIT12                   ;SEEK AND NO MOTION
1318    020000          D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1319    040000          D.SUNS= BIT14                   ;SERVO UNSAFE
1320
1321                    .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)
1322
1323    000007          M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1324    017760          M.CDIF= 17760                   ;CYLINDER DIFF, BYTE 10
1325    017760          M.OFST= 17760                   ;OFFSET VALUE, BYTE 10
1326    077770          M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1327
1328                    .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)
1329
1330    000003          M.ID=   3                       ;BYTE ID, ALL BYTES
1331    017760          M.CADD= 17760                   ;CYLINDER ADDRESS, BYTE 10
1332    040000          M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1333    000760          M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1334    007000          M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1335    100000          M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
1336
1337                                    .SBTTL  TRAP CATCHER
1338
1339            000000                          .=0
1340                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1341                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1342                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1343            000174                          .=174
1344  000174   000000                  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
1345  000176   000000                  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
1346                                    .SBTTL  STARTING ADDRESS(ES)
1347  000200   000137  012542                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1348            000220                          .=220
1349  000220   000137  012532                  JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING
1350
1351            000240                          .=240
1352  000240   000137  055422                  JMP     0.ODT           ;ENTER ODT11
1353
1354                                    .SBTTL  ACT11 HOOKS
1355
1356                                    ;;*****************************************************************
1357                                    ;HOOKS REQUIRED BY ACT11
1358            000244                          $SVPC=.                 ;SAVE PC
1359            000046                          .=46
1360  000046   031636                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1361            000052                          .=52
1362  000052   100000                          .WORD   100000          ;;2)SET LOC.52 TO 100000
1363            000244                          .=$SVPC                 ;; RESTORE PC
1364            001000                          .=1000
1365                                    .SBTTL  APT PARAMETER BLOCK
1366
1367                                    ;;*****************************************************************
1368                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1369                                    ;;*****************************************************************
1370            001000                          .$X=.   ;;SAVE CURRENT LOCATION
1371            000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1372  000024   000200                          200     ;;FOR APT START UP
1373            000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1374  000044   001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
1375            001000                          .=.$X   ;;RESET LOCATION COUNTER
1376                                    ;;*****************************************************************
1377                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1378                                    ;INTERFACE SPEC.
1379
1380  001000                            $APTHD:
1381  001000   000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1382  001002   001210                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1383  001004   000430                  $TSTM:  .WORD   280.    ;;RUN TIM OF LONGEST TEST
1384  001006   001130                  $PASTM: .WORD   600.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1385  001010   001130                  $UNITM: .WORD   600.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1386  001012   000042                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1387
1388
1389                                            .LIST   MD
1390
1391                                    ;
```

```
1392                            ;USE LOOP X TO OMIT SUBCLR
1393                            ;
1394
1395                            .MACRO  LOOP    A
1396                                    SCOP1
1397                                    MOV     #STACK,SP       ;RESTORE STK PTR
1398
1399                            .IF     B       A
1400                                    JSR     PC,SUBCLR
1401                                    ERROR   24              ;CERR AFTER SCLR
1402
1403                            .ENDC
1404                            .ENDM   LOOP
1405
1406                            ;
1407                            ;THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD BITS
1408                            ;A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1409                            ;NOTE: A CAN BE ANY BIT COMBINATION DESIRED.
1410                            ;
1411                            .MACRO  F.EAB   A
1412
1413                                    MOV     #<A!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
1414                                    CLR     E.B0                ;EXPECTED MSG B0
1415                                    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
1416                                    MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
1417                                    CLR     E.A2                ;EXPECTED MSG A2
1418                                    MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
1419                                    MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
1420                            .ENDM   F.EAB
1421
1422
1423                            ;THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
1424                            ;USE    A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
1425                            ;USE    G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1426                            ;       H=T.B2  TO READ MSG B2 & PUT INFO INTO 'CYLADD'
1427                            ;       I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
1428                            ;
1429                            ;USE    F=<ERROR DESCRIPTION>
1430                            ;
1431                            .MACRO  CHECK   A,C,D,E,F,G,H,I
1432
1433                                    JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
1434                                    .WORD   G!H!I               ;& MSGS SPECIFIED HERE
1435                                    ERROR   A                   ;MSG A0 ERROR F
1436                                    ERROR   C                   ;MSH B0 ERROR
1437                                    ERROR   D                   ;MSG A1 ERROR
1438                                    ERROR   E                   ;MSG B1 ERROR
1439                            .ENDM   CHECK
1440
1441                            ;
1442                            ;A=CYL DIFF/OFFSET ERROR #
1443                            ;B=CYL ADDR ERROR #
1444                            ;C=<ERROR DESCRIPTION>
1445                            ;
1446                            .MACRO  CWD2    A,B,C,?D,?E
1447
```

```
1448                                    TST     CYLDIF          ;SEE IF MSG A2=0
1449                                    BEQ     D               ;BR IF YES
1450                                    ERROR   A               ;MSG A2 NOT CLEARED C
1451                            D:      TST     CYLADD          ;SEE IF MSG B2=0
1452                                    BEQ     E               ;BR IF YES
1453                                    ERROR   B               ;MSG B2 NOT CLEARED C
1454                            E:
1455                            .ENDM   CWD2
1456
1457                            .MACRO  DRCLR   ?A
1458
1459                                    MOV     #CCLR,RKCS1(R5)
1460                                    MOV     $UNIT,RKCS2(R5) ;DRIVE#
1461                                    MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
1462                                    MOV     T10,TEMP1       ;SETUP TIMEOUT
1463                                    JSR     PC,FRDY         ;FIND RDY
1464                                    ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
1465                                    JSR     PC,TSTATN       ;TEST FOR ATTN
1466                                    BR      A
1467                                    ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1468                            A:
1469
1470                            .ENDM   DRCLR
1471
1472
1473                            ;
1474                            ;USE CALIB      X   TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
1475                            ;
1476                            .MACRO  CALIB   A,?C
1477
1478                                    MOV     #CCLR,RKCS1(R5)
1479                                    MOV     $UNIT,RKCS2(R5)
1480                                    MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
1481                                                            ;RESET CYL DIFF/OFFSET & CYL ADDR REG
1482                                                            ;IN RKMR2 & RKMR3 RESP.
1483                                    MOV     T10,TEMP1       ;SETUP TIMEOUT
1484                                    JSR     PC,FRDY         ;FIND RDY
1485                                    ERROR   124             ;RDY NOT SET AFTER RECAL CMD
1486
1487                                    MOV     #1,RKMR1(R5)    ;SELECT WORD 1
1488                                    JSR     PC,GSTAT
1489                                    BIT     #D.RTZ,HMR2
1490                                    BNE     C
1491                                    ERROR   244             ;RTZ NOT SET DURING RECAL CMD
1492                            C:      MOV     T10,TEMP2       ;SETUP TIMEOUT
1493                                    JSR     PC,FATT1        ;FIND ATTN
1494                                    ERROR   55              ;NO ATTN AFTER RECAL CMD
1495                            .IF B   A
1496                                    F.EAB   DSC
1497                                    CHECK   221,275,222,276,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
1498                                    CWD2    47,50,<AFTER RECAL CMD>
1499                            .ENDC
1500                                    DRCLR
1501
1502                            .ENDM   CALIB
1503
```

```
1504                            ;
1505                            ;QUICK START SPINDLE
1506                            ;
1507                            .MACRO  QKSRT
1508
1509                                    JSR     PC,SUBCLR
1510                                    ERROR   24              ;CERR AFTER SCLR
1511
1512                                    MOV     #SRTSPL,RKCS1(R5)  ;START SPINDLE CMD
1513                                    MOV     T10,TEMP1       ;SET TIMEOUT
1514                                    JSR     PC,FRDY         ;FIND RDY
1515                                    ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.
1516
1517                                    MOV     T100,TEMP2      ;SETUP TIMEOUT
1518                                    JSR     PC,FATT1        ;FIND ATTN
1519                                    ERROR   74              ;NO ATTN AFTER ST SPIN CMD.
1520
1521                                    CLR     UNLD
1522
1523                            .ENDM   QKSRT
1524
1525                            ;
1526                            ; A=WRHEAD/<CFMT!WRHEAD>
1527                            ; USE WRHDR     <A>,X   TO OMIT CHECKING A0, B0, A1 & B1
1528                            ;
1529                            .MACRO  WRHDR   A,C,K,?D
1530
1531                                    MOV     #<A>,RKCS1(R5)  ;WRITE HEADER CMD
1532                                    MOV     T50000,TEMP1    ;SETUP  TIMEOUT
1533                                    JSR     PC,FRDY         ;FIND RDY
1534                                    ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
1535                            .IF     B       K
1536                                    JSR     PC,GSTAT        ;GET FRESH STATUS
1537                            .ENDC
1538                            .IF     NB      K
1539                                    MOV     #<CFMT!SELDRV>,RKCS1(R5)
1540                                    MOV     T10,TEMP1
1541                                    JSR     PC,FRDY
1542                                    ERROR   117             ;NO RDY AFTER SELDRV CMD
1543                            .ENDC
1544                                    BIT     #CERR,HCS1
1545                                    BEQ     D
1546                                    ERROR   201             ;CERR AFTER WRITE HEADER CMD
1547                                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1548                                    JMP     $EOP
1549                            D:
1550                            .IF B   C
1551                                    F.EAB   0
1552                                    CHECK   277,267,300,270,<AFTER WRITE HEADER CMD>,0,0,0
1553                            .ENDC
1554
1555                            .ENDM   WRHDR
1556
1557                            ;
1558                            ; A=RDHEAD/<CFMT!RDHEAD>
1559                            ;USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
```

```
1560
1561                            ;
                                .MACRO  RDHDR   A,C,?D,?E
1562
1563                                    MOV     #RHTAB,R0
1564                                    MOV     #<A>,RKCS1(R5)  ;READ HEADER CMD
1565                                    MOV     T50000,TEMP1    ;SETUP  TIMEOUT
1566                                    JSR     PC,FRDY         ;FIND RDY
1567                                    ERROR   171             ;NO RDY AFTER READ HEADER CMD
1568                                    BIT     #CERR,HCS1
1569                                    BEQ     D
1570                                    ERROR   174             ;CERR AFTER READ HEADER CMD
1571                                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1572                                    JMP     $EOP
1573
1574                            D:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
1575                                    MOV     RKDB(R5),(R0)+  ;2'ND WORD
1576                                    MOV     RKDB(R5),(R0)+  ;3'RD WORD
1577
1578
1579                                    BIT     #DLT,RKCS2(R5)
1580                                    BEQ     E
1581                                    JSR     PC,GSTAT
1582                                    ERROR   173             ;DLT AFTER READ HEADER CMD
1583                                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1584                                    JMP     $EOP
1585                            E:
1586                            .IF     B       C
1587                                    F.EAB   0
1588                                    CHECK 301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
1589                            .ENDC
1590
1591                            .ENDM   RDHDR
1592
1593
1594                            .MACRO  HDCHK3  ?A
1595
1596                                    RDHDR   RDHEAD
1597                                    CMP     RHTAB,TOCYL     ;CHECK WORD 0 ONLY, CYL#
1598                                    BEQ     A               ;BR IF SAME
1599                                    ERROR   51              ;WRONG CYL# ON HEADER
1600                            A:
1601
1602                            .ENDM   HDCHK3
1603
1604                            ;
1605                            ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1606                            ;
1607                            .MACRO  HDTBL   A,B,C
1608
1609                                    MOV     A,CALADD        ;SETUP
1610                                    MOV     #B,HEAD         ;TO FILL
1611                                    MOV     #C,FORMAT       ;HEADER
1612                                    JSR     PC,FHDTAB       ;TABLE
1613
1614                            .ENDM   HDTBL
1615
```

```
1616
1617                    ;QUICK SEEK.    ENTER WITH CYL# IN RKDC
1618                    ;
1619                    .MACRO  QKSEEK  ?A
1620
1621                            MOV     #SEEK,RKCS1(R5) ;SEEK CMD
1622                            MOV     T50,TEMP1       ;SETUP TIMEOUT
1623                            JSR     PC,FRDY         ;FIND RDY
1624                            ERROR   131             ;NO RDY AFTER SEEK CMD
1625
1626                            MOV     T50000,TEMP1    ;SETUP TIMEOUT
1627                            JSR     PC,FATT2        ;FIND ATTN
1628                            ERROR   132             ;NO ATTN AFTER SEEK CMD
1629
1630                            BIT     #CERR,HCS1
1631                            BEQ     A
1632                            ERROR   210             ;CERR AFTER SEEK CMD
1633
1634                    A:
1635
1636                    .ENDM   QKSEEK
1637
1638
1639                    ;
1640                    ;A=WRDATA/<CFMT!WRDATA>
1641                    ;C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
1642                    ;D=ADDR TO JMP TO BYPASS TEST
1643                    ;E: IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
1644                    ;E: IF NON BLANK WILL OMIT CHECKING A0 THRU B1
1645                    ;
1646
1647                    .MACRO  WDATA   A,C,D,E,J,K,?F,?G,?H,?I
1648
1649                            MOV     #<A>,RKCS1(R5)  ;WRITE DATA CMD
1650                            MOV     T50000,TEMP1    ;SETUP TIMEOUT
1651                            JSR     PC,FRDY         ;FIND RDY
1652                            ERROR   11              ;NO RDY AFTER WRITE DATA CMD
1653                    .IF     B       K
1654                            JSR     PC,GSTAT        ;GET FRESH STATUS
1655                    .ENDC
1656                    .IF     NB      K
1657                            MOV     #<CFMT!SELDRV>,RKCS1(R5)
1658                            MOV     T10,TEMP1
1659                            JSR     PC,FRDY
1660                            ERROR   117             ;NO RDY AFTER SELDRV CMD
1661                    .ENDC
1662                            BIT     #CERR,HCS1
1663                            BEQ     I               ;BR IF NO ERRORS
1664
1665                            BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
1666                            BEQ     G               ;BR IF NO
1667                            JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
1668                            BR      H               ;RETURN HERE IF NO
1669                    .IF     B       J
1670
1671                            INC     SECTOR          ;RETURN HERE IF YES
```

```
1672                                    CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
1673                                    BNE     F               ;BR IF NO
1674                                    ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
1675                                    JMP     D               ;BYPASS TEST
1676                                F:  MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
1677                                    JMP     C
1678                                .ENDC
1679                                .IF     NB      J
1680                                    JMP     3$              ;RET HERE IF YES
1681                                .ENDC
1682                                G:  ERROR   12              ;CERR WITH WRITE DATA CMD
1683                                    F.EAB   0
1684                                    CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1685                                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1686                                    JMP     $EOP
1687                                H:  ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
1688                                I:
1689                                .IF     B       E
1690                                    F.EAB   0
1691                                    CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1692                                .ENDC
1693                                .ENDM   WDATA
1694
1695
1696                                ;
1697                                ;A=RDDATA/<CFMT!RDDATA>
1698                                ;USE RDATA      <A>,X TO OMIT CHECKING A0, B0, A1 & B1
1699                                ;
1700
1701                                .MACRO  RDATA   A,C,K,?D,?E,?F,?G,?H
1702
1703                                    MOV     #<A>,RKCS1(R5)  ;READ DATA CMD
1704                                    MOV     T50000,TEMP1    ;SETUP TIMEOUT
1705                                    JSR     PC,FRDY         ;FIND RDY
1706                                    ERROR   13              ;NO RDY AFTER READ DATA CMD
1707                                .IF     B       K
1708                                    JSR     PC,GSTAT        ;GET FRESH STATUS
1709                                .ENDC
1710                                .IF     NB      K
1711                                    MOV     #<CFMT!SELDRV>,RKCS1(R5)
1712                                    MOV     T10,TEMP1
1713                                    JSR     PC,FRDY
1714                                    ERROR   117             ;NO RDY AFTER SELDRV CMD
1715                                .ENDC
1716                                    BIT     #CERR,HCS1
1717                                    BEQ     G
1718                                    BIT     #BSE,HER        ;SEE IF BAD SECTOR
1719                                    BEQ     E
1720                                    ERROR   65              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1721                                    BR      H
1722                                D:  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1723                                    JMP     $EOP
1724
1725                                E:  BIT     #DCK,HER        ;SEE IF DATA CHECK ERROR
1726                                    BEQ     F
1727                                    ERROR   21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
```

```
1728                                            BR      H
1729
1730                                    F:      ERROR   14              ;CERR AFTER READ DATA CMD.
1731
1732                                    H:      F.EAB   0
1733                                            CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1734                                            BR      D
1735                                    G:
1736                                    .IF     B       C
1737                                            F.EAB   0
1738                                            CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1739                                    .ENDC
1740                                    .ENDM   RDATA
1741
1742
1743                                    ;
1744                                    ;A=WRTCHK/<CFMT!WRTCHK>
1745                                    ;C=EXPECTED DATA FOR TYPEOUT
1746                                    ;USE WRCHK      <A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
1747                                    ;
1748
1749                                    .MACRO  WRCHK   A,C,D,K,?E,?F
1750
1751                                            MOV     #<A>,RKCS1(R5)  ;WRITE CHECK CMD
1752                                            MOV     T50000,TEMP1    ;SETUP TIMEOUT
1753                                            JSR     PC,FRDY         ;FIND RDY
1754                                            ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
1755                                    .IF     B       K
1756                                            JSR     PC,GSTAT        ;GET FRESH STATUS
1757                                    .ENDC
1758                                    .IF     NB      K
1759                                            MOV     #<CFMT!SELDRV>,RKCS1(R5)
1760                                            MOV     T10,TEMP1
1761                                            JSR     PC,FRDY
1762                                            ERROR   117             ;NO RDY AFTER SELDRV CMD
1763                                    .ENDC
1764                                            BIT     #CERR,HCS1
1765                                            BEQ     F
1766                                            BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
1767                                            BEQ     E
1768                                            MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
1769                                            MOV     C,WD2           ;EXPECTED WORD FOR TYPEOUT
1770                                            ERROR   16              ;WCE AFTER WRITE CMD
1771                                            BR      F
1772
1773                                    E:      ERROR   22              ;CERR AFTER WRITE CHECK CMD
1774                                            F.EAB   0
1775                                            CHECK   57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1776                                            TYPE    .MSG26          ;ABORTING BAL OF TESTS
1777                                            JMP     $EOP
1778
1779                                    F:
1780                                    .IF     B       D
1781                                            F.EAB   0
1782                                            CHECK   57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1783                                    .ENDC
```

```
1784                            .ENDM   WRCHK
1785
1786
1787                            .MACRO  OFSET ?A
1788
1789                                    MOV     #A,$ESCAPE
1790                                    MOV     #OFFSET,RKCS1(R5)   ;OFFSET CMD
1791                                    MOV     T100,TEMP1          ;SETUP TIMEOUT
1792                                    JSR     PC,FRDY
1793                                    ERROR   33                  ;NO RDY AFTER OFFSET CMD
1794
1795                                    MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
1796                                    CLR     E.B0
1797                                    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
1798                                    MOV     #1,E.B1
1799                                    CHECK   35,61,36,62,<DURING OFFSET CMD>,0,0,0
1800
1801                            A:      CLR     $ESCAPE
1802                                    MOV     T5000,TEMP1         ;SETUP TIMEOUT
1803                                    JSR     PC,FATT2            ;FIND ATTN
1804                                    ERROR   34                  ;NO ATTN AFTER OFFSET CMD
1805
1806                                    F.EAB   <D.DSC!D.OFF>
1807                                    CHECK   260,261,37,40,<AFTER OFFSET CMD>,T.A2,T.B2,0
1808
1809                            .ENDM   OFSET
1810
1811
1812                            .MACRO  EOPGM
1813
1814                                    SCOPE
1815                                    MOV     #1,$TIMES
1816                                    MOV     #STACK,SP
1817                                    INC     $DEVCT              ;INCR COUNT FOR # DRIVES CHECKED
1818                                    CMP     DRIVS,$DEVCT        ;ARE ALL DRIVES PRESENT TESTED?
1819                                    BEQ     $EOP1+2             ;BR IF YES
1820                                    JMP     NUDRV               ;ELSE TEST NEXT DRIVE PRESENT
1821                            $EOP1:  SCOPE
1822                            .ENDM   EOPGM
1823                            ;
1824                            ;A= ERROR #
1825                            ;B = ERROR CONDITION
1826                            ;
1827                            .MACRO  OFFDIR  A,B,?C,?D
1828                                    MOV     R2,RKASOF(R5)       ;REFRESH RKASOF
1829
1830                                    BIT     #BIT7,R2
1831                                    BNE     C                   ;BR IF NEG OFFSET
1832
1833                                    CMP     R2,CYLDIF           ;CHECK POS OFFSET
1834                                    BEQ     D
1835                                    ERROR   A                   ;OFFSET IN A2 NOT = RKASOF
1836                                    BR      D                   ;B
1837
1838                            C:      CMP     R1,CYLDIF           ;CHECK NEG OFFSET
1839                                    BEQ     D
```

```
1840                                    ERROR   A               ;OFFSET IN A2 NOT = RKASOF
1841                                                            ;B
1842                                D:
1843                                .ENDM   OFFDIR
1844
1845                                    .NLIST  MD
1846
```

```
                                .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
```

```
 1903  001212  000000                $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
 1904  001214  000000                $TESTN: .WORD   ATESTN  ;;TEST NUMBER
 1905  001216  000000                $PASS:  .WORD   APASS   ;;PASS COUNT
 1906  001220  000000                $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
 1907  001222  000000                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
 1908  001224  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
 1909  001226  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
 1910  001230                        $ETABLE:                ;;APT ENVIRONMENT TABLE
 1911  001230     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
 1912  001231     000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
 1913  001232  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
 1914  001234  000000                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
 1915  001236  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
 1916                                ;*                      BITS 15-11=CPU TYPE
 1917                                ;*                           11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
 1918                                ;*                           11/70=06,PDQ=07,Q=10
 1919                                ;*                      BIT 10=REAL TIME CLOCK
 1920                                ;*                      BIT  9=FLOATING POINT PROCESSOR
 1921                                ;*                      BIT  8=MEMORY MANAGEMENT
 1922  001240     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
 1923  001241     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
 1924                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
 1925                                ;*                           900 NSEC CORE=001
 1926                                ;*                           300 NSEC BIPOLAR=002
 1927                                ;*                           500 NSEC MOS=003
 1928  001242  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
 1929                                ;*                      MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
 1930  001244     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
 1931  001245     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
 1932  001246  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
 1933  001250     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
 1934  001251     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
 1935  001252  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
 1936  001254     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
 1937  001255     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
 1938  001256  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
 1939  001260  000000                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
 1940  001262  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
 1941  001264  177440                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
 1942  001266  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
 1943  001270  000000                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
 1944  001272  000000                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
 1945  001274  000000                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
 1946  001276  000000                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
 1947  001300  000000                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
 1948  001302  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
 1949  001304  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
 1950  001306  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
 1951  001310  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
 1952  001312  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
 1953  001314                        $ETEND:
 1954                                .MEXIT
 1955          177440                ABASE=  177440                  ;DEFAULT BUSS ADDRESS
 1956  001314  000210                RKVEC:  210                     ;DEFAULT CONTROLLER INTERRUPT VECTOR
 1957  001316  000240                RKPRI:  PR5                     ;PRIORITY
 1958  001320  172540                PKS:    172540                  ;P-CLOCK STATUS REG
```

```
 1959  001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
 1960  001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
 1961  001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.
 1962
 1963  001330  000100          LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
 1964  001332  000104          PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.
 1965
 1966          000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
 1967          172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
 1968  001334  000000          TRAPPC: 0               ;PC FOR MEM CHECK ENABLE TRAP
 1969
 1970  001336  000000          PARAM:  0               ;1 FOR 220 START, NO DEFAULT
 1971  001340  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
 1972
 1973  001342  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
 1974                                                  ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
 1975  001344  000000          FRCYL:  0               ;FROM CYLINDER
 1976  001346  000000          TOCYL:  0               ;TO CYLINDER
 1977  001350  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
 1978  001352  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
 1979  001354  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
 1980  001356  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
 1981  001360  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
 1982  001362  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
 1983
 1984  001364  000074          HZ:     60.             ;60 FOR 60 CPS
 1985                                                  ;50 FOR 50 CPS
 1986  001366  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
 1987                                                  ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
 1988  001370  000000          SEC:    0               ;SECOND COUNTER
 1989  001372  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
 1990  001374  000000          SECNT:  0               ;SECTOR COUNT
 1991  001376  000000          PSEC:   0               ;PREVIOUS SECTOR
 1992  001400  000000          ESEC:   0               ;EXPECTED SECTOR
 1993  001402  000000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
 1994
 1995  001404  000001          T1:     1               ;TIMEOUT CONSTANTS
 1996  001406  000012          T10:    10.
 1997  001410  000062          T50:    50.
 1998  001412  000764          T500:   500.
 1999  001414  000144          T100:   100.
 2000  001416  011610          T5000:  5000.
 2001  001420  141520          T50000: 50000.
 2002
 2003  001422  000077          CYL:    63.             ;CYLINDER NUMBERS USED IN
 2004  001424  000177                  127.            ;CURRENT CROSSOVER TEST
 2005  001426  000277                  191.
 2006  001430  000377                  255.
 2007  001432  000477                  319.
 2008  001434  000577                  383.
 2009
 2010  001436  000000          WD1:    0               ;ACTUAL HEADER/DATA WORD
 2011  001440  000000          WD2:    0               ;EXPECTED DATA WORD
 2012
 2013  001442  000000          OFFERR: 0               ;SET WHEN WRITE CHECK ERROR ON OFFSET
 2014
```

```
                                HEAD:   0                       ;HEAD NUMBER
                                HEADP:  0                       ;HEAD # FROM H.B3, RT. JUSTIFIED
                                HD1:    0                       ;SHIFTED HEAD# FOR FORMATTER ROUTINE
                                FORMAT: 0                       ;FORMAT TYPE
                                FMT1:   0                       ;SHIFTED FORMAT FOR FORMATTER ROUTINE
                                WDCNT:  0                       ;WORD COUNT

                                DATA0:  0                       ;ALL 0'S
                                DATA01: 52525                   ;0101 PATT
                                DATA1:  177777                  ;ALL 1'S
                                DPAT1:  133467
                                DPAT2:  70627

                                WORD:   0                       ;HEADER/DATA WORD
                                HDWD:   0                       ;HEADER WORD FROM RKDB

                                BSERR:  0                       ;CANNOT READ BSE INFO WHEN SET
                                LIMERR: 0                       ;LIMIT DETECT ERROR FLAG
                                BYPCERR:0                       ;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
                                CHKFLG: 0                       ;WORDS TO BE TESTED

                                HDTAB:  .BLKW 66.               ;CALCULATED HEADER WORD TABLE
                                RHTAB:  .BLKW 66.               ;FILLED AFTER READ HEADER CMD
                                SRTTAB: .BLKW 66.               ;ABOVE RHTAB SORTED STARTING FORM
                                                                ;SECTOR 0 BY SORT ROUTINE
                                BSE20H: .BLKW 256.              ;20 SECTOR HARDWARE BSE INFO
                                BSE22H: .BLKW 256.              ;22 SECTOR HARDWARE BSE INFO.
                                BSE20S: .BLKW  256.             ;20 SECTOR SOFTWARE BSE INFO.
                                BSE22S: .BLKW  256.             ;22 SECTOR SOFTWARE BSE INFO.
                                RDTAB:  .BLKW 256.              ;FILLED AFTER READ DATA CMD

                                UNLD:   0                       ;SET TO 0 IF HEADS ARE LOADED
                                                                ;SET TO 1 IF HEADS UNLOADED
                                BADHDR: 0                       ;SET TO 0 IF FORMATTING OK
                                                                ;SET TO 1 IF FORMATTING ALTERED
                                HPEND:  0                       ;SET TO 0 IF HALT NOT PENDING
                                                                ;SET TO 1 IF HALT PENDING

                                                                ;THE ABOVE 3 FLAGS ARE USED
                                                                ;BY 'STOP' ROUTINE TO BRING
                                                                ;THE CPU TO A VALID HALT.


                                ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.

                                        .EVEN

                                ;
                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
                                ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
                                ;

                                HCS1:   0                       ;HOLD RKCS1
                                HCS2:   0                       ;HOLD RKCS2
```

```
2071  007344  000000          HWC:    0                       ;HOLD RKWC
2072  007346  000000          HBA:    0                       ;ETC.
2073  007350  000000          HDA:    0
2074  007352  000000          HDS:    0
2075  007354  000000          HER:    0
2076  007356  000000          HASOF:  0
2077  007360  000000          HDC:    0
2078  007362  000000          HDB:    0
2079  007364  000000          HMR1:   0
2080  007366  000000          HMR2:   0
2081  007370  000000          HMR3:   0
2082  007372  000000          HPOS:   0
2083  007374  000000          HPAT:   0
2084
2085
2086  007376  000000          TEMP1:  0                       ;TEMPORARY STORAGE.
2087  007400  000000          TEMP2:  0
2088  007402  000000          TEMP3:  0
2089  007404  000000          TEMP4:  0
2090  007406  000000          TEMP5:  0
2091                          ;
2092                          ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
2093                          ;
2094  007410  000000          H.A0:   0
2095  007412  000000          H.B0:   0
2096  007414  000000          H.A1:   0
2097  007416  000000          H.B1:   0
2098  007420  000000          H.A2:   0
2099  007422  000000          H.B2:   0
2100  007424  000000          H.A3:   0
2101  007426  000000          H.B3:   0
2102                          ;
2103                          ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
2104                          ;
2105  007430  000000          E.A0:   0
2106  007432  000000          E.B0:   0
2107  007434  000000          E.A1:   0
2108  007436  000000          E.B1:   0
2109  007440  000000          E.A2:   0
2110  007442  000000          E.B2:   0
2111  007444  000000          E.A3:   0
2112  007446  000000          E.B3:   0
2113                          ;
2114                          ;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.
2115                          ;
2116          000001          T.A2=BIT0                       ;TEST MSG A2 IF SET
2117          000002          T.B2=BIT1
2118          000004          T.B3=BIT2
2119
2120                          ;
2121                          ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
2122                          ;
2123
2124  007450  000000          DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
2125  007452  000000          DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
2126  007454  000000          ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
```

```
2127  007456  000000          PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
2128  007460  000000          DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
2129
2130                          ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
2131                          ;IS PRESENT AND IS TO BE TESTED.
2132
2133  007462  000000          DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
2134  007464  000000          DRIV1:  0                       ;FOR DRIVE 1
2135  007466  000000          DRIV2:  0                       ;FOR DRIVE 2
2136  007470  000000          DRIV3:  0                       ;FOR DRIVE 3
2137  007472  000000          DRIV4:  0                       ;FOR DRIVE 4
2138  007474  000000          DRIV5:  0                       ;FOR DRIVE 5
2139  007476  000000          DRIV6:  0                       ;FOR DRIVE 6
2140  007500  000000          DRIV7:  0                       ;FOR DRIVE 7
2141
2142  007502  000000          LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
2143  007504  000000          PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
2144  007506  000000          DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
2145  007510  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

007512                          $ERRTB:

                                ;ERROR 1
007512  046136                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
007514  051572                          DH1
007516  053654                          DT1
007520  054456                          DF1

                                ;ERROR 2
007522  046355                          EM5                     ;DETECTED MDS
007524  051572                          DH1
007526  053654                          DT1
007530  054456                          DF1

                                ;ERROR 3
007532  046376                          EM6                     ;DETECTED UFE
007534  051572                          DH1
007536  053654                          DT1
007540  054456                          DF1

                                ;ERROR 4
007542  046417                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
007544  051572                          DH1
007546  053654                          DT1
007550  054456                          DF1
                                ;ERROR 5
007552  046506                          EM8                     ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
007554  051572                          DH1
007556  053654                          DT1
007560  054456                          DF1

                                ;ERROR 6
007562  046562                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
007564  051572                          DH1
007566  053654                          DT1
007570  054456                          DF1

                                ;ERROR 7
007572  046636                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
007574  051572                          DH1
007576  053654                          DT1
007600  054456                          DF1
```

ERROR POINTER TABLE

```
2202
2203                                    ;ERROR 10
2204  007602  046721                          EM11            ;DRA & NED BOTH SET
2205  007604  051572                          DH1
2206  007606  053654                          DT1
2207  007610  054456                          DF1
2208                                    ;ERR 11
2209  007612  046765                          EM12            ;NO RDY
2210  007614  052555                          DH27            ;AFTER WRITE DATA CMD
2211  007616  053654                          DT1
2212  007620  054606                          DF10
2213                                    ;ERR 12
2214  007622  047361                          EM21            ;CERR SET
2215  007624  052555                          DH27
2216  007626  053654                          DT1
2217  007630  054606                          DF10
2218                                    ;ERR 13
2219  007632  046765                          EM12            ;NO RDY
2220  007634  052525                          DH26            ;AFTER READ DATA CMD
2221  007636  053654                          DT1
2222  007640  054606                          DF10
2223                                    ;ERR 14
2224  007642  047361                          EM21            ;CERR SET
2225  007644  052525                          DH26
2226  007646  053654                          DT1
2227  007650  054606                          DF10
2228                                    ;ERR 15
2229  007652  046765                          EM12            ;NO RDY
2230  007654  052705                          DH32            ;AFTER WRITE CHECK CMD
2231  007656  053654                          DT1
2232  007660  054606                          DF10
2233                                    ;ERR 16
2234  007662  050547                          EM80            ;WRITE CHECK ERROR SET
2235  007664  052705                          DH32            ;AFTER WRITE CHECK CMD
2236  007666  053766                          DT6
2237  007670  054476                          DF3
2238                                    ;ERR 17
2239  007672  050606                          EM81            ;WRITE CHECK CMD NOT FUNCTIONING
2240  007674  053434                          DH52            ;WITH INTENTIONAL MISCOMPARE
2241  007676  053654                          DT1
2242  007700  054606                          DF10
2243                                    ;ERR 20
2244  007702  050652                          EM82            ;READ DATA NOT COMPARE WITH WRITE DATA
2245  007704  052525                          DH26            ;AFTER READ DATA CMD
2246  007706  053766                          DT6
2247  007710  054476                          DF3
2248                                    ;ERR 21
2249  007712  050724                          EM83            ;DATA CHECK ERROR
2250  007714  052525                          DH26
2251  007716  053654                          DT1
2252  007720  054606                          DF10
2253                                    ;ERR 22
2254  007722  047361                          EM21            ;CERR SET
2255  007724  052705                          DH32            ;AFTER WRITE CHECK CMD
2256  007726  053654                          DT1
2257  007730  054606                          DF10
```

```
2258
2259  007732  047276          ;ERR 23
2260  007734  052555                EM18                    ;MSG B0 ERROR
2261  007736  054212                DH27                    ;AFTER WRITE DATA CMD
2262  007740  054736                DT13
2263                            DF21
2264  007742  047361          ;ERROR 24
2265  007744  052375                EM21                    ;CERR SET
2266  007746  053654                DH21                    ;AFTER SCLR
2267  007750  054606                DT1
2268                            DF10
2269  007752  047340          ;ERR 25
2270  007754  052555                EM20                    ;MSG B1 ERROR
2271  007756  054212                DH27
2272  007760  054736                DT13
2273                            DF21
2274  007762  047276          ;ERR 26
2275  007764  052525                EM18
2276  007766  054212                DH26                    ;AFTER READ DATA CMD
2277  007770  054736                DT13
2278                            DF21
2279
2280  007772  047610          ;ERROR 27
2281  007774  052317                EM24                    ;VOL VALID NOT SET
2282  007776  053654                DH19                    ;AFTER PACK CMD
2283  010000  054606                DT1
2284                            DF10
2285  010002  047340          ;ERR 30
2286  010004  052525                EM20                    ;MSG B1 ERROR
2287  010006  054212                DH26                    ;AFTER READ DATA CMD.
2288  010010  054736                DT13
2289                            DF21
2290  010012  047276          ;ERR 31
2291  010014  052705                EM18                    ;MSG B0 ERROR
2292  010016  054212                DH32                    ;AFTER WRITE CHECK CMD
2293  010020  054736                DT13
2294                            DF21
2295  010022  047340          ;ERR 32
2296  010024  052705                EM20                    ;MSG B1 ERROR
2297  010026  054212                DH32
2298  010030  054736                DT13
2299                            DF21
2300  010032  046765          ;ERR 33
2301  010034  052455                EM12                    ;CONTR NOT READY
2302  010036  053654                DH24                    ;AFTER OFFSET CMD
2303  010040  054606                DT1
2304                            DF10
2305  010042  047023          ;ERR 34
2306  010044  052455                EM13                    ;NO ATTN
2307  010046  053654                DH24
2308  010050  054606                DT1
2309                            DF10
2310  010052  047255          ;ERR 35
2311  010054  053470                EM17                    ;MSG A0 ERROR
2312  010056  054212                DH53                    ;DURING OFFSET COMMAND
2313  010060  054736                DT13
                                DF21
```

```
2314
2315  010062  047317          ;ERR 36
2316  010064  053470                EM19                    ;MSG A1 ERROR
2317  010066  054212                DH53
2318  010070  054736                DT13
2319                           DF21
2320  010072  047317          ;ERR 37
2321  010074  052455                EM19                    ;MSG A1 ERROR
2322  010076  054212                DH24                    ;AFTER OFFSET CMD
2323  010100  054736                DT13
2324                           DF21
2325  010102  047340          ;ERR 40
2326  010104  052455                EM20                    ;MSG B1 ERROR
2327  010106  054212                DH24
2328  010110  054736                DT13
2329                           DF21
2330  010112  047045          ;ERR 41
2331  010114  051763                EM14                    ;UNEXP MEM PCRITY TRAP
2332  010116  053714                DH8                     ;TEST #, TRAP PC
2333  010120  054472                DT3
2334                           DF2
2335  010122  050263          ;ERR 42
2336  010124  052455                EM41                    ;CYL ADDR IN B2 DID NOT REMAIN CLEARED
2337  010126  054272                DH24
2338  010130  054772                DT14
2339                           DF22
2340  010132  051052          ;ERR 43
2341  010134  052423                EM85
2342  010136  053654                DH22
2343  010140  054606                DT1
2344                           DF10
2345  010142  047103          ;ERR 44
2346  010144  051572                EM15                    ;WCE AT CYL 411,TRK 2, SEC 21
2347  010146  053654                DH1
2348  010150  054522                DT1
2349                           DF4
2350  010152  051052          ;ERR 45
2351  010154  053401                EM85                    ;OFFSET BIT IN RKMR2 CLEARED
2352  010156  053654                DH51                    ;AFTER SEEK TO SELF
2353  010160  054606                DT1
2354                           DF10
2355  010162  047643          ;ERR 46
2356  010164  052555                EM25                    ;DETECTED 10 BAD SECTORS
2357  010166  053654                DH27                    ;AFTER WRITE DATA CMD.
2358  010170  054606                DT1
2359                           DF10
2360  010172  050156          ;ERROR 47
2361  010174  052273                EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2362  010176  054272                DH17                    ;AFTER RECAL CMD
2363  010200  054772                DT14
2364                           DF22
2365  010202  050225          ;ERROR 50
2366  010204  052273                EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
2367  010206  054272                DH17                    ;AFTER RECAL COMD
2368  010210  054772                DT14
2369                           DF22
                               ;ERR 51
```

ERROR POINTER TABLE

```
2370  010212  051214                EM93            ;WRONG CYL# IN HEADER WORD (MISPOSITION)
2371  010214  052502                DH25            ;AFTER SEEK CMD
2372  010216  054146                DT9
2373  010220  054712                DF20
2374                          ;ERR 52
2375  010222  047255                EM17            ;MSG AD ERROR
2376  010224  052555                DH27            ;AFTER WRITE DATA CMD
2377  010226  054212                DT13
2378  010230  054736                DF21
2379                          ;ERR 53
2380  010232  047317                EM19            ;MSG A1 ERROR
2381  010234  052555                DH27
2382  010236  054212                DT13
2383  010240  054736                DF21
2384                          ;ERR 54
2385  010242  047255                EM17            ;MSG AD ERROR
2386  010244  052525                DH26            ;AFTER READ DATA CMD
2387  010246  054212                DT13
2388  010250  054736                DF21
2389                          ;ERROR 55
2390  010252  047023                EM13            ;NO ATTN
2391  010254  052273                DH17            ;AFTER RECAL CMD
2392  010256  053654                DT1
2393  010260  054606                DF10
2394                          ;ERR 56
2395  010262  047317                EM19            ;MSG A1 ERROR
2396  010264  052525                DH26
2397  010266  054212                DT13
2398  010270  054736                DF21
2399                          ;ERR 57
2400  010272  047255                EM17            ;MSG AD ERROR
2401  010274  052705                DH32            ;AFTER WRITE CHECK CMD
2402  010276  054212                DT13
2403  010300  054736                DF21
2404                          ;ERR 60
2405  010302  047317                EM19            ;MSG A1 ERROR
2406  010304  052705                DH32
2407  010306  054212                DT13
2408  010310  054736                DF21
2409                          ;ERR 61
2410  010312  047276                EM18            ;MSG BD ERROR
2411  010314  053470                DH53            ;DURING OFFSET CMD
2412  010316  054212                DT13
2413  010320  054736                DF21
2414                          ;ERR 62
2415  010322  047340                EM20            ;MSG B1 ERROR
2416  010324  053470                DH53
2417  010326  054212                DT13
2418  010330  054736                DF21
2419                          ;ERR 63
2420  010332  047715                EM26            ;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
2421  010334  052555                DH27            ;AFTER WRITE DATA CMD
2422  010336  053654                DT1
2423  010340  054606                DF10
2424                          ;ERR 64
2425  010342  051153                EM88            ;DID NOT FIND SECTOR 0 FROM INDEX
```

```
2426  010344  053516                    DH54                    ;AFTER FORMAT CHANGE AND READY REC'D
2427  010346  053654                    DT1
2428  010350  054606                    DF10
2429                          ;ERR 65
2430  010352  047774                    EM27                    ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
2431  010354  051572                    DH1
2432  010356  053654                    DT1
2433  010360  054456                    DF1
2434                          ;ERR 66
2435  010362  000000                    0
2436  010364  000000                    0
2437  010366  000000                    0
2438  010370  000000                    0
2439                          ;ERR 67
2440  010372  000000                    0
2441  010374  000000                    0
2442  010376  000000                    0
2443  010400  000000                    0
2444                          ;ERROR 70
2445  010402  000000                    0
2446  010404  000000                    0
2447  010406  000000                    0
2448  010410  000000                    0
2449                          ;ERR 71
2450  010412  000000                    0
2451  010414  000000                    0
2452  010416  000000                    0
2453  010420  000000                    0
2454                          ;ERROR 72
2455  010422  000000                    0
2456  010424  000000                    0
2457  010426  000000                    0
2458  010430  000000                    0
2459                          ;ERR 73
2460  010432  000000                    0
2461  010434  000000                    0
2462  010436  000000                    0
2463  010440  000000                    0
2464                          ;ERR 74
2465  010442  047023                    EM13                    ;NO ATTN
2466  010444  052057                    DH10                    ;AT END OF HEAD LOADING
2467  010446  053654                    DT1
2468  010450  054606                    DF10
2469                          ;ERR 75
2470  010452  047403                    EM22                    ;NO DRIVS IN $DEVM
2471  010454  051572                    DH1
2472  010456  053654                    DT1
2473  010460  054456                    DF1
2474                          ;ERR 76
2475  010462  047510                    EM23                    ;NO DRIVS ON BUSS
2476  010464  051572                    DH1
2477  010466  053654                    DT1
2478  010470  054456                    DF1
2479                          ;ERR 77
2480  010472  000000                    0
2481  010474  000000                    0
```

```
2482  010476  000000                    0
2483  010500  000000                    0
2484                            ;ERR 100
2485  010502  000000                    0
2486  010504  000000                    0
2487  010506  000000                    0
2488  010510  000000                    0
2489                            ;ERROR 101
2490  010512  051275                    EM94        ;OFFSET NOT CLEARED
2491  010514  053254                    DH47        ;AFTER READ HEADER WITH MOVEMENT
2492  010516  054272                    DT14
2493  010520  054772                    DF22
2494                            ;ERROR 102
2495  010522  051331                    EM95        ;FORMAT NOT SET
2496  010524  052555                    DH27        ;AFTER WRITE DATA CMD
2497  010526  053654                    DT1
2498  010530  054606                    DF10
2499                            ;ERR 103
2500  010532  051331                    EM95
2501  010534  052705                    DH32        ;AFTER WRITE CHECK CMD
2502  010536  053654                    DT1
2503  010540  054606                    DF10
2504                            ;ERR 104
2505  010542  050156                    EM39        ;OFFSET NOT RESET
2506  010544  053614                    DH57        ;AFTER WRITE CMD WITH OFFSET
2507  010546  054272                    DT14
2508  010550  054772                    DF22
2509                            ;ERR 105
2510  010552  050225                    EM40        ;CYL ADDR NOT 0
2511  010554  053614                    DH57
2512  010556  054272                    DT14
2513  010560  054772                    DF22
2514                            ;ERR 106
2515  010562  051365                    EM96        ;CANNOT FIND SECTOR 23(8)
2516  010564  051572                    DH1
2517  010566  053654                    DT1
2518  010570  054456                    DF1
2519                            ;ERR 107
2520  010572  051416                    EM97        ;HEAD SWITCHING TOO LONG
2521  010574  052555                    DH27        ;AFTER WRITE DTA CMD
2522  010576  053654                    DT1
2523  010600  054646                    DF15
2524                            ;ERR 110
2525  010602  000000                    0
2526  010604  000000                    0
2527  010606  000000                    0
2528  010610  000000                    0
2529                            ;ERR 111
2530  010612  000000                    0
2531  010614  000000                    0
2532  010616  000000                    0
2533  010620  000000                    0
2534                            ;ERR 112
2535  010622  051503                    EM100       ;DRIVE OFF TRACK SET
2536  010624  052555                    DH27        ;AFTER WRITE DATA CMD
2537  010626  053720                    DT4
```

```
2538  010630  054562                    DF6
2539                              ;ERR 113
2540  010632  050113                    EM36          ;CYL ADDR IN RKMR3 INCORRECT
2541  010634  052555                    DH27
2542  010636  053720                    DT4
2543  010640  054562                    DF6
2544                              ;ERROR 114
2545  010642  051115                    EM86          ;OFFSET IN A2 NOT = RKASOF
2546  010644  052455                    DH24          ;AFTER OFFSET CMD
2547  010646  054272                    DT14
2548  010650  054772                    DF22
2549                              ;ERR 115
2550  010652  051115                    EM86
2551  010654  052423                    DH22          ;AFTER DRIVE CLEAR CMD
2552  010656  054272                    DT14
2553  010660  054772                    DF22
2554                              ;ERROR 116
2555  010662  046765                    EM12          ;CONT NOT RDY
2556  010664  052317                    DH19          ;AFTER PACK CMD
2557  010666  053654                    DT1
2558  010670  054606                    DF10
2559                              ;ERROR 117
2560  010672  046765                    EM12          ;CONT NOT RDY
2561  010674  052342                    DH20          ;AFTER SEL DR CMD
2562  010676  053654                    DT1
2563  010700  054606                    DF10
2564                              ;ERROR 120
2565  010702  046765                    EM12
2566  010704  052375                    DH21          ;AFTER SUBSYS CLEAR
2567  010706  053654                    DT1
2568  010710  054606                    DF10
2569                              ;ERROR 121
2570  010712  046765                    EM12
2571  010714  052004                    DH9           ;AFTER START SPINDLE CMD
2572  010716  053654                    DT1
2573  010720  054606                    DF10
2574                              ;ERROR 122
2575  010722  051540                    EM101         ;DID NOT GO TO CYL 10
2576  010724  052626                    DH30          ;AFTER READ HEADER CMD
2577  010726  054272                    DT14
2578  010730  054772                    DF22
2579                              ;ERROR 123
2580  010732  051115                    EM86          ;A2 OFFSET NOT = RKASOF
2581  010734  053401                    DH51          ;AFTER SEEK TO SELF
2582  010736  054272                    DT14
2583  010740  054772                    DF22
2584                              ;ERROR 124
2585  010742  046765                    EM12
2586  010744  052273                    DH17          ;AFTER RECAL CMD
2587  010746  053654                    DT1
2588  010750  054606                    DF10
2589                              ;ERR 125
2590  010752  050460                    EM73          ;CTO SET
2591  010754  050761                    EM84          ;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
2592  010756  053654                    DT1
2593  010760  054536                    DF5
```

```
2594
2595  010762  050526          ;ERR 126
2596  010764  050761                EM79                    ;NED SET
2597  010766  053654                EM84
2598  010770  054536                DT1
2599                           DF5
2600  010772  046355          ;ERR 127
2601  010774  050761                EM5                     ;MDS SET
2602  010776  053654                EM84
2603  011000  054536                DT1
2604                           DF5
2605  011002  000000          ;ERROR 130
2606  011004  000000                0
2607  011006  000000                0
2608  011010  000000                0
2609                           0
2610  011012  046765          ;ERROR 131
2611  011014  052502                EM12                    ;NO RDY
2612  011016  053654                DH25                    ;AFTER SEEK CMD
2613  011020  054606                DT1
2614                           DF10
2615  011022  047023          ;ERROR 132
2616  011024  052502                EM13                    ;NO ATTN
2617  011026  053654                DH25
2618  011030  054606                DT1
2619                           DF10
2620  011032  000000          ;ERROR 133
2621  011034  000000                0
2622  011036  000000                0
2623  011040  000000                0
2624                           0
2625  011042  000000          ;ERROR 134
2626  011044  000000                0
2627  011046  000000                0
2628  011050  000000                0
2629                           0
2630  011052  000000          ;ERROR 135
2631  011054  000000                0
2632  011056  000000                0
2633  011060  000000                0
2634                           0
2635  011062  000000          ;ERROR 136
2636  011064  000000                0
2637  011066  000000                0
2638  011070  000000                0
2639                           0
2640  011072  050156          ;ERROR 137
2641  011074  052502                EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2642  011076  053654                DH25
2643  011100  054606                DT1
2644                           DF10
2645  011102  047255          ;ERR 140
2646  011104  053401                EM17                    ;MSG AO ERROR
2647  011106  054212                DH51                    ;AFTER SEEK TO SELF
2648  011110  054736                DT13
2649                           DF21
                               ;ERR 141
```

```
2650  011112  047276                    EM18
2651  011114  053401                    DH51
2652  011116  054212                    DT13
2653  011120  054736                    DF21
2654                            ;ERR 142
2655  011122  047317                    EM19
2656  011124  053401                    DH51
2657  011126  054212                    DT13
2658  011130  054736                    DF21
2659                            ;ERR 143
2660  011132  047340                    EM20
2661  011134  053401                    DH51
2662  011136  054212                    DT13
2663  011140  054736                    DF21
2664                            ;ERROR 144
2665  011142  000000                    0
2666  011144  000000                    0
2667  011146  000000                    0
2668  011150  000000                    0
2669                            ;ERROR 145
2670  011152  000000                    0
2671  011154  000000                    0
2672  011156  000000                    0
2673  011160  000000                    0
2674                            ;ERROR 146
2675  011162  000000                    0
2676  011164  000000                    0
2677  011166  000000                    0
2678  011170  000000                    0
2679                            ;ERROR 147
2680  011172  000000                    0
2681  011174  000000                    0
2682  011176  000000                    0
2683  011200  000000                    0
2684                            ;ERROR 150
2685  011202  000000                    0
2686  011204  000000                    0
2687  011206  000000                    0
2688  011210  000000                    0
2689                            ;ERROR 151
2690  011212  046765                    EM12        ;NO RDY
2691  011214  052423                    DH22        ;AFTER CLEAR CMD
2692  011216  053654                    DT1
2693  011220  054606                    DF10
2694                            ;ERROR 152
2695  011222  000000                    0
2696  011224  000000                    0
2697  011226  000000                    0
2698  011230  000000                    0
2699                            ;ERROR 153
2700  011232  000000                    0
2701  011234  000000                    0
2702  011236  000000                    0
2703  011240  000000                    0
2704                            ;ERROR 154
2705  011242  050331                    EM55        ;ATTN NOT CLEARED
```

```
2706  011244  052423                    DH22
2707  011246  053654                    DT1
2708  011250  054606                    DF10
2709                              ;ERROR 155
2710  011252  000000                    0
2711  011254  000000                    0
2712  011256  000000                    0
2713  011260  000000                    0
2714                              ;ERROR 156
2715  011262  000000                    0
2716  011264  000000                    0
2717  011266  000000                    0
2718  011270  000000                    0
2719                              ;ERROR 157
2720  011272  000000                    0
2721  011274  000000                    0
2722  011276  000000                    0
2723  011300  000000                    0
2724                              ;ERROR 160
2725  011302  000000                    0
2726  011304  000000                    0
2727  011306  000000                    0
2728  011310  000000                    0
2729                              ;ERROR 161
2730  011312  000000                    0
2731  011314  000000                    0
2732  011316  000000                    0
2733  011320  000000                    0
2734                              ;ERROR 162
2735  011322  000000                    0
2736  011324  000000                    0
2737  011326  000000                    0
2738  011330  000000                    0
2739                              ;ERROR 163
2740  011332  000000                    0
2741  011334  000000                    0
2742  011336  000000                    0
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
```

```
2762
2763
2764  011340  000000                    0
2765                                ;ERROR 164
2766  011342  000000                    0
2767  011344  000000                    0
2768  011346  000000                    0
2769  011350  000000                    0
2770                                ;ERROR 165
2771  011352  000000                    0
2772  011354  000000                    0
2773  011356  000000                    0
2774  011360  000000                    0
2775                                ;ERROR 166
2776  011362  000000                    0
2777  011364  000000                    0
2778  011366  000000                    0
2779  011370  000000                    0
2780                                ;ERROR 167
2781  011372  000000                    0
2782  011374  000000                    0
2783  011376  000000                    0
2784  011400  000000                    0
2785                                ;ERROR 170
2786  011402  000000                    0
2787  011404  000000                    0
2788  011406  000000                    0
2789  011410  000000                    0
2790                                ;ERROR 171
2791  011412  046765                    EM12        ;NO RDY
2792  011414  052626                    DH30        ;AFTER READ HEADER CMD
2793  011416  053654                    DT1
2794  011420  054606                    DF10
2795                                ;ERROR 172
2796  011422  000000                    0
2797  011424  000000                    0
2798  011426  000000                    0
2799  011430  000000                    0
2800                                ;ERROR 173
2801  011432  050364                    EM63        ;DLT SET
2802  011434  052626                    DH30
2803  011436  053654                    DT1
2804  011440  054646                    DF15
2805                                ;ERROR 174
2806  011442  047361                    EM21        ;CERR SET
2807  011444  052626                    DH30
2808  011446  053654                    DT1
2809  011450  054646                    DF15
2810                                ;ERROR 175
2811  011452  050156                    EM39        ;CYL DIFF NOT CLEARED
2812  011454  052057                    DH10        ;AT END OF HEAD LOADING
2813  011456  053654                    DT1
2814  011460  054606                    DF10
2815                                ;ERROR 176
2816  011462  050225                    EM40        ;CYL ADDR NOT CLEARED.
2817  011464  052057                    DH10
```

```
                        ERROR POINTER TABLE

2818  011466  053654           DT1
2819  011470  054606           DF10
2820                    ;ERROR 177
2821  011472  000000           0
2822  011474  000000           0
2823  011476  000000           0
2824  011500  000000           0
2825                    ;ERROR 200
2826  011502  046765           EM12            ;NO RDY
2827  011504  052737           DH39            ;AFTER WRITE HEADER CMD
2828  011506  053654           DT1
2829  011510  054646           DF15
2830                    ;ERROR 201
2831  011512  047361           EM21            ;CERR SET
2832  011514  052737           DH39
2833  011516  053654           DT1
2834  011520  054646           DF15
2835                    ;ERROR 202
2836  011522  050405           EM65            ;READ HEADER ERROR
2837  011524  051572           DH1
2838  011526  054032           DT7
2839  011530  054626           DF14
2840                    ;ERROR 203
2841  011532  000000           0
2842  011534  000000           0
2843  011536  000000           0
2844  011540  000000           0
2845                    ;ERROR 204
2846  011542  000000           0
2847  011544  000000           0
2848  011546  000000           0
2849  011550  000000           0
2850                    ;ERROR 205
2851  011552  000000           0
2852  011554  000000           0
2853  011556  000000           0
2854  011560  000000           0
2855                    ;ERROR 206
2856  011562  000000           0
2857  011564  000000           0
2858  011566  000000           0
2859  011570  000000           0
2860                    ;ERROR 207
2861  011572  050113           EM36            ;CYL ADDR IN RKMR3 INCORRECT
2862  011574  052502           DH25            ;AFTER SEEK CMD
2863  011576  053720           DT4
2864  011600  054562           DF6
2865                    ;ERROR 210
2866  011602  047361           EM21            ;CERR SET
2867  011604  052502           DH25
2868  011606  053654           DT1
2869  011610  054606           DF10
2870                    ;ERROR 211
2871  011612  000000           0
2872  011614  000000           0
2873  011616  000000           0
```

```
2874  011620  000000                    0
2875                                ;ERROR 212
2876  011622  000000                    0
2877  011624  000000                    0
2878  011626  000000                    0
2879  011630  000000                    0
2880                                ;ERROR 213
2881  011632  000000                    0
2882  011634  000000                    0
2883  011636  000000                    0
2884  011640  000000                    0
2885                                ;ERROR 214
2886  011642  000000                    0
2887  011644  000000                    0
2888  011646  000000                    0
2889  011650  000000                    0
2890                                ;ERROR 215
2891  011652  000000                    0
2892  011654  000000                    0
2893  011656  000000                    0
2894  011660  000000                    0
2895                                ;ERROR 216
2896  011662  000000                    0
2897  011664  000000                    0
2898  011666  000000                    0
2899  011670  000000                    0
2900                                ;ERROR 217
2901  011672  000000                    0
2902  011674  000000                    0
2903  011676  000000                    0
2904  011700  000000                    0
2905                                ;ERROR 220
2906  011702  000000                    0
2907  011704  000000                    0
2908  011706  000000                    0
2909  011710  000000                    0
2910                                ;ERROR 221
2911  011712  047255                    EM17          ;MSG A0 ERROR
2912  011714  052273                    DH17
2913  011716  054212                    DT13
2914  011720  054736                    DF21
2915                                ;ERROR 222
2916  011722  047317                    EM19          ;MSG A1 ERROR
2917  011724  052273                    DH17
2918  011726  054212                    DT13
2919  011730  054736                    DF21
2920                                ;ERROR 223
2921  011732  000000                    0
2922  011734  000000                    0
2923  011736  000000                    0
2924  011740  000000                    0
2925                                ;ERROR 224
2926  011742  000000                    0
2927  011744  000000                    0
2928  011746  000000                    0
2929  011750  000000                    0
```

```
2930
2931  011752  000000                ;ERROR 225
2932  011754  000000                        0
2933  011756  000000                        0
2934  011760  000000                        0
2935                                        0
2936  011762  046765                ;ERROR 226
2937  011764  052525                        EM12                    ;NO RDY
2938  011766  053654                        DH26                    ;AFTER READ DATA CMD
2939  011770  054606                        DT1
2940                                        DF10
2941  011772  047361                ;ERROR 227
2942  011774  052525                        EM21                    ;CERR SET
2943  011776  053654                        DH26
2944  012000  054646                        DT1
2945                                        DF15
2946  012002  047212                ;ERROR 230
2947  012004  052123                        EM16                    ;CANNOT READ BSE INFO
2948  012006  053654                        DH13                    ;ON SEC 10, 12, 14, 16, 18, 20
2949  012010  054666                        DT1
2950                                        DF17
2951  012012  047212                ;ERROR 231
2952  012014  052207                        EM16
2953  012016  053654                        DH14                    ;ON SEC 11, 13, 15, 17, 19, 21
2954  012020  054666                        DT1
2955                                        DF17
2956  012022  000000                ;ERROR 232
2957  012024  000000                        0
2958  012026  000000                        0
2959  012030  000000                        0
2960                                        0
2961  012032  047212                ;ERROR 233
2962  012034  053053                        EM16                    ;CANNOT READ BSE INFO
2963  012036  053654                        DH42                    ;ON SECT 0,2,4,6,8
2964  012040  054666                        DT1
2965                                        DF17
2966  012042  047212                ;ERROR 234
2967  012044  053124                        EM16
2968  012046  053654                        DH43                    ;ON SECT 1,3,5,7,9
2969  012050  054666                        DT1
2970                                        DF17
2971  012052  050427                ;ERROR 235
2972  012054  053175                        EM69                    ;ALIGN CARTRIDGE USED
2973  012056  053654                        DH44                    ;WILL BYPASS FORMAT & ALL R/W TESTS
2974  012060  054606                        DT1
2975                                        DF10
2976  012062  000000                ;ERROR 236
2977  012064  000000                        0
2978  012066  000000                        0
2979  012070  000000                        0
2980                                        0
2981  012072  000000                ;ERROR 237
2982  012074  000000                        0
2983  012076  000000                        0
2984  012100  000000                        0
2985                                        0
                                    ;ERROR 240
```

ERROR POINTER TABLE

```
2986  012102  000000                    0
2987  012104  000000                    0
2988  012106  000000                    0
2989  012110  000000                    0
2990                              ;ERROR 241
2991  012112  000000                    0
2992  012114  000000                    0
2993  012116  000000                    0
2994  012120  000000                    0
2995                              ;ERROR 242
2996  012122  000000                    0
2997  012124  000000                    0
2998  012126  000000                    0
2999  012130  000000                    0
3000
3001                              ;ERROR 243
3002  012132  050113                    EM36          ;CYL ADDR IN RKMR3 INCORRECT
3003  012134  052502                    DH25          ;AFTER SEEK CMD
3004  012136  054100                    DT8
3005  012140  054562                    DF6
3006                              ;ERR 244
3007  012142  050501                    EM74          ;RTZ NOT SET
3008  012144  053026                    DH41          ;DURING RECAL CMD
3009  012146  053654                    DT1
3010  012150  054606                    DF10
3011                              ;ERR 245
3012  012152  000000                    0
3013  012154  000000                    0
3014  012156  000000                    0
3015  012160  000000                    0
3016                              ;ERR 246
3017  012162  000000                    0
3018  012164  000000                    0
3019  012166  000000                    0
3020  012170  000000                    0
3021                              ;ERR 247
3022  012172  000000                    0
3023  012174  000000                    0
3024  012176  000000                    0
3025  012200  000000                    0
3026                              ;ERR 250
3027  012202  000000                    0
3028  012204  000000                    0
3029  012206  000000                    0
3030  012210  000000                    0
3031                              ;ERR 251
3032  012212  000000                    0
3033  012214  000000                    0
3034  012216  000000                    0
3035  012220  000000                    0
3036                              ;ERR 252
3037  012222  000000                    0
3038  012224  000000                    0
3039  012226  000000                    0
3040  012230  000000                    0
3041                              ;ERR 253
```

```
3042  012232  000000                    0
3043  012234  000000                    0
3044  012236  000000                    0
3045  012240  000000                    0
3046                            ;ERR 254
3047  012242  000000                    0
3048  012244  000000                    0
3049  012246  000000                    0
3050  012250  000000                    0
3051                            ;ERR 255
3052  012252  000000                    0
3053  012254  000000                    0
3054  012256  000000                    0
3055  012260  000000                    0
3056                            ;ERR 256
3057  012262  000000                    0
3058  012264  000000                    0
3059  012266  000000                    0
3060  012270  000000                    0
3061                            ;ERR 257
3062  012272  000000                    0
3063  012274  000000                    0
3064  012276  000000                    0
3065  012300  000000                    0
3066                            ;ERR 260
3067  012302  047255                EM17          ;MSG AD ERROR
3068  012304  052455                DH24          ;AFTER OFFSET CMD
3069  012306  054212                DT13
3070  012310  054736                DF21
3071                            ;ERR 261
3072  012312  047276                EM18          ;MSG BD ERROR
3073  012314  052455                DH24
3074  012316  054212                DT13
3075  012320  054736                DF21
3076                            ;ERR 262
3077  012322  000000                    0
3078  012324  000000                    0
3079  012326  000000                    0
3080  012330  000000                    0
3081                            ;ERR 263
3082  012332  000000                    0
3083  012334  000000                    0
3084  012336  000000                    0
3085  012340  000000                    0
3086                            ;ERR 264
3087  012342  000000                    0
3088  012344  000000                    0
3089  012346  000000                    0
3090  012350  000000                    0
3091                            ;ERR 265
3092  012352  047276                EM18          ;MSG BD ERROR
3093  012354  052423                DH22          ;AFTER DRIVE CLEAR CMD
3094  012356  054212                DT13
3095  012360  054736                DF21
3096                            ;ERR 266
3097  012362  047340                EM20          ;MSG B1 ERROR
```

```
3098  012364  052423                    DH22
3099  012366  054212                    DT13
3100  012370  054736                    DF21
3101                              ;ERR 267
3102  012372  047276                    EM18          ;MSG B0 ERROR
3103  012374  052737                    DH39          ;AFTER WRITE HEADER CMD
3104  012376  054212                    DT13
3105  012400  054736                    DF21
3106                              ;ERR 270
3107  012402  047340                    EM20          ;MSG B1 ERROR
3108  012404  052737                    DH39
3109  012406  054212                    DT13
3110  012410  054736                    DF21
3111                              ;ERR 271
3112  012412  047276                    EM18
3113  012414  052626                    DH30          ;AFTER RD. HDR. CMD.
3114  012416  054212                    DT13
3115  012420  054736                    DF21
3116                              ;ERR 272
3117  012422  047340                    EM20
3118  012424  052626                    DH30
3119  012426  054212                    DT13
3120  012430  054736                    DF21
3121                              ;ERR 273
3122  012432  047255                    EM17          ;MSG A0 ERROR
3123  012434  052423                    DH22          ;AFTER DRV CLR CMD
3124  012436  054212                    DT13
3125  012440  054736                    DF21
3126                              ;ERR 274
3127  012442  047317                    EM19          ;MSG A1 ERROR
3128  012444  052423                    DH22
3129  012446  054212                    DT13
3130  012450  054736                    DF21
3131                              ;ERR 275
3132  012452  047276                    EM18          ;MSG B0 ERROR
3133  012454  052273                    DH17          ;AFTER RECAL CMD
3134  012456  054212                    DT13
3135  012460  054736                    DF21
3136                              ;ERR 276
3137  012462  047340                    EM20          ;MSG B1 ERROR
3138  012464  052273                    DH17
3139  012466  054212                    DT13
3140  012470  054736                    DF21
3141                              ;ERR 277
3142  012472  047255                    EM17          ;MSG A0 ERROR
3143  012474  052737                    DH39          ;AFTER WRITE HEADER CMD
3144  012476  054212                    DT13
3145  012500  054736                    DF21
3146                              ;ERR 300
3147  012502  047317                    EM19          ;MSG A1 ERROR
3148  012504  052737                    DH39
3149  012506  054212                    DT13
3150  012510  054736                    DF21
3151                              ;ERR 301
3152  012512  047255                    EM17
3153  012514  052626                    DH30          ;AFT RD HDR. CMD
```

```
  3154  012516  054212                    DT13
  3155  012520  054736                    DF21
  3156                            ;ERR 302
  3157  012522  047317                    EM19
  3158  012524  052626                    DH30
  3159  012526  054212                    DT13
  3160  012530  054736                    DF21
  3161
```

```
3162
3163                                    .SBTTL  PROGRAM SETUP
3164
3165  012532  012737  000001  001336  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
3166  012540  000402                          BR      PRGSRT          ;START PROGRAM
3167
3168  012542  005037  001336          START:  CLR     PARAM           ;CLEAR FOR 200 START
3169  012546  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
3170  012550  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3171  012554  012746  000000                  MOV     #PRO,-(SP)      ;PSW LOADED TO BE
3172  012560  012746  012566                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
3173  012564  000002                          RTI                     ;ENABLE ALL INTERRUPTS
3174
3175  012566  004737  041044          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
3176                                                                   ;& TURN ON KB INTERRUPT
3177
3178
3179                                  ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3180                                  ;*** ANY DEVICE WHICH SETS ITS   ***
3181                                  ;*** INTERRUPT ENABLE BIT WILL   ***
3182                                  ;*** SERVICED.                   ***
3183
3184                                  ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3185                                  ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3186                                  ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3187
3188                                  ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3189
3190                                                                   ;SYSMAC 'SETUP'
3191                                  .SBTTL  INITIALIZE THE COMMON TAGS
3192                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3193  012572  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3194  012576  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
3195  012600  022706  001140                  CMP     #SWR,R6 ;;DONE?
3196  012604  001374                          BNE     .-6             ;;LOOP BACK IF NO
3197  012606  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3198                                  ;;INITIALIZE A FEW VECTORS
3199  012612  012737  037152  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3200  012620  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
3201  012626  012737  037432  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3202  012634  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
3203  012642  012737  043262  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3204  012650  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
3205  012656  012737  036706  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
3206  012664  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
3207  012672  013737  031604  031576          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
3208  012700  005037  001174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
3209  012704  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3210  012710  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
3211  012716  012737  012716  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3212  012724  012737  012724  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
3213                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3214                                  ;;EQUAL TO A "-1"  SETUP FOR A SOFTWARE SWITCH REGISTER.
3215  012732  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3216  012736  012737  012772  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3217  012744  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
```

```
3218  012752  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3219  012760  022777  177777  166152          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3220  012766  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3221                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
3222  012770  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
3223  012772  012716  013000          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
3224  012776  000002                          RTI
3225  013000  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
3226  013006  012737  000174  001142          MOV     #DISPREG,DISPLAY
3227  013014  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
3228
3229  013020  005037  001216                  CLR     $PASS           ;;CLEAR PASS COUNT
3230  013024  132737  000200  001231          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
3231  013032  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
3232  013034  012737  001232  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
3233  013042                          67$:
3234  013042  012737  013106  000004  MEMPAR: MOV     #1$,ERRVEC      ;SETUP TIMEOUT VECTOR
3235  013050  012737  000340  000006          MOV     #PR7,ERRVEC+2
3236
3237  013056  012701  172100                  MOV     #MEMBAS,R1      ;ADDR OF MEM CSR
3238  013062  005011                  3$:     CLR     (R1)            ;SEE IF CAN REFERENCE
3239  013064  012711  000001                  MOV     #1,(R1)         ;SET ENABLE BIT IF YES
3240  013070  012737  036610  000114          MOV     #MEMERR,MEMVEC  ;LOAD VECTOR IF NO TIMEOUT
3241  013076  012737  000340  000116          MOV     #PR7,MEMVEC+2
3242  013104  000401                          BR      2$
3243
3244  013106  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
3245  013110  062701  000002          2$:     ADD     #2,R1           ;TRY NEXT CSR
3246  013114  020127  172140                  CMP     R1,#MEMBAS+40   ;SEE IF TRIED ALL
3247  013120  001360                          BNE     3$              ;BR IF NO
3248  013122  012737  000006  000004          MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
3249  013130  005037  000006                  CLR     ERRVEC+2
3250
3251  013134  004737  031672                  JSR     PC,CLRFLG       ;CLEAR DDUMP THRU SIZFLG
3252  013140  005037  001220                  CLR     $DEVCT
3253  013144  005037  001222                  CLR     $UNIT
3254
3255                                  ;
3256                                  ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
3257                                  ;
3258
3259  013150  005737  000042          START1: TST     42
3260  013154  001014                          BNE     1$              ;BR IF AUTO
3261  013156  004737  031712                  JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
3262  013162  123727  000041  000013          CMPB    41,#13          ;13=LOADED BY XXDP
3263  013170  001010                          BNE     2$
3264  013172  005237  007450                  INC     DDUMP           ;SET RK06 DUMP MODE FLAG
3265  013176  104401  044242                  TYPE    ,MSG2           ;REPLACE DRO PACK W/SCRATCH & DO<CR>
3266  013202  000137  013216                  JMP     ST2
3267  013206  000137  013262          1$:     JMP     ST3
3268  013212  005237  007456          2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG
3269
3270                                  ;
3271                                  ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
3272                                  ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
3273                                  ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
```

```
3274                                    ;       EX:     DRIVES TO BE TESTED:  1,2,4<CR>
3275                                    ;
3276
3277  013216  005737  001336            ST2:    TST     PARAM
3278  013222  001002                            BNE     1$              ;BR IF 220 START
3279  013224  000137  013314                    JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
3280  013230  104401  044313            1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
3281  013234  004737  031772                    JSR     PC,GDRVS        ;GET DR NOS.
3282  013240  104401  044345                    TYPE    ,MSG4           ;BUSS ADDR
3283  013244  004737  032132                    JSR     PC,GBA          ;GET BA
3284  013250  104401  044412                    TYPE    ,MSG5           ;CONT INT VECTOR
3285  013254  004737  032160                    JSR     PC,GINT         ;GET INT VECTOR
3286  013260  000427                            BR      ST5
3287
3288                                    ;
3289                                    ;AUTO MODE
3290                                    ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
3291                                    ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
3292                                    ;ON THE BUSS
3293                                    ;
3294
3295  013262  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
3296  013270  001007                            BNE     1$
3297  013272  005237  007452                    INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
3298  013276  004737  031712                    JSR     PC,TITLE
3299  013302  104401  044527                    TYPE    ,MSG7           ;DRO NOT TSTD
3300  013306  000402                            BR      ST4
3301  013310  005237  007454            1$:     INC     ACT11           ;SET ACT AUTO FLAG.
3302
3303  013314  012737  177440  001264  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
3304  013322  012737  000210  001314            MOV     #210,RKVEC      ;DEFAULT VALUE
3305  013330  004737  032212                    JSR     PC,SETINT
3306  013334  005237  007510                    INC     SIZFLG          ;DO "SIZE THE BUSS" TEST
3307
3308  013340  005037  007322  ST5:    CLR     UNLD            ;INITIALIZE FLAGS
3309  013344  005037  007324            CLR     BADHDR          ;USED IN 'STOP' ROUTINE
3310  013350  005037  007326            CLR     HPEND           ;FOR VALID PROGRAM HALTS
3311  013354  005037  001176            CLR     $ESCAPE
3312  013360  012737  007462  001342    MOV     #DRIVD,DRVPTR   ;SETUP
3313  013366  005037  001220            CLR     $DEVCT          ;NO. OF DRVS DONE
3314  013372  005037  001222            CLR     $UNIT           ;CURRENT DRV UNDER TEST
3315  013376  012737  013444  000004    MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
3316  013404  005777  165716            TST     @LKS            ;SEE IF L-CLOCK THERE
3317  013410  005237  007502            INC     LCLKF           ;PRESENT, SET FLAG.
3318  013414  013700  001330            MOV     LCVEC,R0        ;VECTOR ADDR
3319  013420  012737  013506  000004    MOV     #2$,ERRVEC
3320  013426  005777  165666            TST     @PKS            ;SEE IF P-CLOCK THERE
3321  013432  005237  007504            INC     PCLKF           ;PRESENT, SET FLAG
3322  013436  013700  001332            MOV     PCVEC,R0        ;VECTOR ADDR
3323  013442  000412                    BR      3$
3324
3325  013444  022626          1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
3326  013446  012737  013512  000004    MOV     #4$,ERRVEC
3327  013454  005777  165640            TST     @PKS            ;SEE IF P-CLOCK THERE
3328  013460  005237  007504            INC     PCLKF           ;PRESENT, SET FLAG
3329  013464  013700  001332            MOV     PCVEC,R0        ;VECTOR ADDR
```

```
  3330  013470  005237  007506  3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
  3331  013474  012720  036010          MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
  3332  013500  012710  000300          MOV     #PR6,(R0)
  3333  013504  000407                  BR      TST1            ;;GO TO NEXT TEST
  3334
  3335  013506  022626          2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
  3336  013510  000767                  BR      3$
  3337
  3338  013512  022626          4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
  3339  013514  005037  007506          CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
  3340  013520  104401  045070          TYPE    ,MSG13          ;HEAD SW. TEST BYPASSED
  3341
  3342
```

```
                                .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 MESSAGE.  ANY
                                ;*       ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                                ;*       TESTS AND JUMPING TO  'END OF PASS'
                                ;*
                                ;;*********************************************************************
013524  000004                  TST1:   SCOPE
013526  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
013534  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
013540  012746  000000                  MOV     #PRO,-(SP)      ;RESET PSW TO PRIORITY 0
013544  012746  013552                  MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
013550  000002                          RTI
013552                          5$:
013552  012737  013670  000004          MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
013560  013705  001264                  MOV     $BASE,R5        ;SETUP INDEX REG.
013564  005765  000000                  TST     RKCS1(R5)       ;REFERENCE ALL THE
013570  005765  000010                  TST     RKCS2(R5)       ;CONTROLLER REGISTERS
013574  005765  000002                  TST     RKWC(R5)
013600  005765  000004                  TST     RKBA(R5)
013604  005765  000006                  TST     RKDA(R5)
013610  005765  000012                  TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
013614  005765  000014                  TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
013620  005765  000016                  TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
013624  005765  000020                  TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
013630  005765  000024                  TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
013634  005765  000026                  TST     RKMR1(R5)
013640  005765  000034                  TST     RKMR2(R5)
013644  005765  000036                  TST     RKMR3(R5)
013650  005765  000030                  TST     RKECPS(R5)
013654  005765  000032                  TST     RKECPT(R5)

013660  012737  036522  000004          MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
013666  000404                          BR      TST2            ;;GO TO NEXT TEST

013670  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
013672  104007                          ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
013674  000137  031550                  JMP     $EOP1

                                ;;*********************************************************************
                                ;*TEST 2         SIZE THE BUSS
                                ;*
                                ;*       THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                                ;*       EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                                ;*       MANUAL MODE.
                                ;*       EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
                                ;*       CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
                                ;*       DRIVE WILL BE TESTED.  IF SET, THE PROGRAM WILL BYPASS
```

```
                                ;*      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.
                                ;*
                                ;*************************************************************************
013700  000004                TST2:   SCOPE
013702  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
013710  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

013714  005237  001502                INC     BYPCERR         ;DO NOT TEST CERR IN 'FRDY'

013720  132737  000200  001231        BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
013726  001002                        BNE     14$             ;BR IF YES
013730  000137  014044                JMP     12$             ;ELSE DO NORM SIZING OR VERIFY

013734  104401  044636        14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
013740  005037  007460                CLR     DRIVS           ;# OF DRIVES PRESENT
013744  005000                        CLR     R0              ;DRV ADDR
013746  012701  007462                MOV     #DRIV0,R1       ;DRV FLAG
013752  013702  001266                MOV     $DEVM,R2        ;APT DEVICE MAP

013756  032702  000001        15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
013762  001410                        BEQ     16$             ;BR IF NO
013764  005237  007460                INC     DRIVS           ;ELSE INCR DRIVE COUNT
013770  005211                        INC     (R1)            ;& SET DRIVE PRESENT FLAG
013772  104401  001205                TYPE    ,$CRLF
013776  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DRIVE #
014000  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
014002     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
014003     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS

014004  005721                16$:    TST     (R1)+           ;ADV POINTER TO NEXT FLAG
014006  005200                        INC     R0              ;INC DRIVE #
014010  022700  000010                CMP     #8.,R0          ;ALL 8 TESTED?
014014  001402                        BEQ     17$             ;BR IF YES

014016  006002                        ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
014020  000756                        BR      15$             ;& TRY AGAIN

014022  005737  007460        17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
014026  001402                        BEQ     18$             ;BR IF NO
014030  000137  014764                JMP     NUDRV           ;ELSE EXIT TEST

014034  104075                18$:    ERROR   75              ;NO DRIVES FOUND IN $DEVM
014036  000000                        HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
014040  000137  013340                JMP     ST5             ;TO TRY AGAIN

014044  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
014052  013737  001406  007376        MOV     T10,TEMP1       ;SET TIMEOUT
014060  004737  032230                JSR     PC,FRDY         ;FIND RDY
014064  104120                        ERROR   120             ;RDY NOT SET BY END OF SCLR
014066  005737  007510                TST     SIZFLG
014072  001562                        BEQ     TST3            ;;DO NOT SIZE, GOTO NEXT TEST
014074  104401  044636                TYPE    ,MSG10          ;WILL TEST DRIVES
```

```
3455  014100  005037  007460              CLR     DRIVS           ;# OF DRIVES PRESENT
3456  014104  005000                      CLR     R0              ;DRV ADDR
3457  014106  012701  007462              MOV     #DRIV0,R1       ;DRV FLAG
3458  014112                      1$:
3459  014112  104415                      SCOP1
3460  014114  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
3461
3462  014120  012765  000040  000010      MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3463  014126  013737  001406  007376      MOV     T10,TEMP1       ;SET TIMEOUT
3464  014134  004737  032230              JSR     PC,FRDY         ;FIND RDY
3465  014140  104120                      ERROR   120             ;RDY NOT SET BY END OF SCLR
3466  014142  010065  000010              MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
3467  014146  012765  000001  000000      MOV     #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
3468  014154  013737  001406  007376      MOV     T10,TEMP1
3469  014162  004737  032230              JSR     PC,FRDY         ;FIND RDY
3470  014166  104117                      ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
3471  014170  032737  100000  007340      BIT     #CERR,HCS1
3472  014176  001046                      BNE     2$
3473  014200  013737  007366  007376      MOV     HMR2,TEMP1
3474  014206  042737  177770  007376      BIC     #^C<DRVMSK>,TEMP1
3475  014214  020037  007376              CMP     R0,TEMP1        ;S/B SAME
3476  014220  001016                      BNE     3$
3477  014222  005700                      TST     R0
3478  014224  001003                      BNE     4$
3479  014226  005737  007452              TST     DDPCH           ;SEE IF XXDP CHAIN MODE
3480  014232  001014                      BNE     5$
3481  014234  005237  007460      4$:     INC     DRIVS           ;INC DRIVE COUNT.
3482  014240  005211                      INC     (R1)            ;SET DRIVE PRESENT FLAG
3483  014242  104401  001205              TYPE    ,$CRLF
3484  014246  010046                      MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3485                                                              ;;TYPE DR #
3486  014250  104403                      TYPOS                   ;;GO TYPE--OCTAL ASCII
3487  014252     001                      .BYTE   1               ;;TYPE 1 DIGIT(S)
3488  014253     000                      .BYTE   0               ;;SUPPRESS LEADING ZEROS
3489  014254  000403                      BR      5$
3490
3491  014256  004737  032736      3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
3492  014262  104001                      ERROR   1               ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
3493
3494  014264  005721              5$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR. FLAG
3495  014266  005200                      INC     R0              ;INC DR #
3496  014270  022700  000010              CMP     #8.,R0
3497  014274  001306                      BNE     1$              ;MORE LEFT.
3498  014276  005737  007460              TST     DRIVS
3499  014302  001054                      BNE     10$
3500  014304  104076                      ERROR   76              ;NO DRIVES FOUND ON BUSS
3501  014306  000000                      HALT                    ;SETUP CORRECTLY
3502  014310  000137  013340              JMP     ST5             ;AND PRESS 'CONTINUE'
3503
3504  014314  032737  001000  007342  2$: BIT     #MDS,HCS2
3505  014322  001015                      BNE     6$
3506  014324  032737  000400  007342      BIT     #UFE,HCS2
3507  014332  001015                      BNE     7$
3508  014334  032737  000001  007352      BIT     #DRA,HDS
3509  014342  001015                      BNE     8$
3510  014344  032737  010000  007342      BIT     #NED,HCS2
```

```
                014352  001424                  BEQ     9$
                014354  000743                  BR      5$
                
                014356  004737  032736  6$:     JSR     PC,BYP          ;TYPE BYP DR #
                014362  104002                  ERROR   2               ;MDS DETECTED
                014364  000737                  BR      5$
                
                014366  004737  032736  7$:     JSR     PC,BYP
                014372  104003                  ERROR   3               ;UFE DETECTED
                014374  000733                  BR      5$
                
                014376  032737  010000  007342  8$:     BIT     #NED,HCS2
                014404  001713                  BEQ     4$
                014406  104401  045211          TYPE    ,MSG15          ;DRV#
                014412  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                        ;;TYPE DR#
                014414  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
                014416     001                  .BYTE   1               ;;TYPE 1 DIGIT(S)
                014417     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
                014420  104010                  ERROR   10              ;DRA & NED BOTH SET
                014422  000720                  BR      5$
                
                014424  004737  032736  9$:     JSR     PC,BYP
                014430  104004                  ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
                014432  000714                  BR      5$
                014434  000137  014764  10$:    JMP     NUDRV
                
                ;;*****************************************************************
                ;*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.
                ;*
                ;*****************************************************************
                
                014440  000004          ST3:    SCOPE
                014442  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
                014450  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
                014454  005000                  CLR     R0              ;DRIVE ADDR
                014456  012701  007462          MOV     #DRIV0,R1       ;DRIVE FLAG
                014462                  1$:
                014462  104415                  SCOP1
                014464  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
                
                014470  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
                014476  013737  001406  007376          MOV     T10,TEMP1       ;SET TIME OUT
                014504  004737  032230                  JSR     PC,FRDY         ;FIND RDY
                014510  104120                  ERROR   120             ;NO RDY AFTER SCLR
                014512  010065  000010          MOV     R0,RKCS2(R5)    ;DRV ADDR
```

```
3567  014516  012765  000001  000000          MOV     #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
3568  014524  013737  001406  007376          MOV     T10,TEMP1
3569  014532  004737  032230                  JSR     PC,FRDY           ;FIND RDY
3570  014536  104117                          ERROR   117               ;NO RDY AFTER SELECT DRIVE CMD.
3571  014540  032737  100000  007340          BIT     #CERR,HCS1
3572  014546  001036                          BNE     2$
3573  014550  013737  007366  007376          MOV     HMR2,TEMP1
3574  014556  042737  177770  007376          BIC     #^C<DRVMSK>,TEMP1
3575  014564  020037  007376                  CMP     R0,TEMP1          ;S/B SAME
3576  014570  001010                          BNE     3$
3577  014572  005711                  11$:    TST     (R1)
3578  014574  001417                          BEQ     5$
3579  014576  005721                  4$:     TST     (R1)+             ;SHIFT PTR TO NEXT DR FLAG
3580  014600  005200                          INC     R0                ;INC DR#
3581  014602  022700  000010                  CMP     #8.,R0
3582  014606  001325                          BNE     1$                ;MORE LEFT
3583  014610  000467                          BR      TST4              ;;GO TO NEXT TEST
3584
3585  014612  004737  032736          3$:     JSR     PC,BYP            ;TRY BYPASS DRIVE#
3586  014616  104001                          ERROR   1                 ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
3587  014620  005711                          TST     (R1)
3588  014622  001765                          BEQ     4$                ;BRANCH IF NOT SPEC BY INPUT
3589  014624  005337  007460          12$:    DEC     DRIVS             ;DECREMENT TOTAL DRIVS
3590  014630  005011                          CLR     (R1)              ;CLEAR DRIVE FLAG
3591  014632  000761                          BR      4$
3592
3593  014634  004737  032736          5$:     JSR     PC,BYP
3594  014640  104005                          ERROR   5                 ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
3595  014642  000755                          BR      4$
3596
3597  014644  032737  001000  007342  2$:     BIT     #MDS,HCS2
3598  014652  001027                          BNE     6$
3599  014654  032737  000400  007342          BIT     #UFE,HCS2
3600  014662  001027                          BNE     7$
3601  014664  032737  000001  007352          BIT     #DRA,HDS
3602  014672  001005                          BNE     8$
3603  014674  032737  010000  007342          BIT     #NED,HCS2
3604  014702  001423                          BEQ     9$
3605  014704  000404                          BR      10$
3606  014706  032737  010000  007342  8$:     BIT     #NED,HCS2
3607  014714  001726                          BEQ     11$
3608  014716  005711                  10$:    TST     (R1)
3609  014720  001726                          BEQ     4$
3610
3611  014722  004737  032736                  JSR     PC,BYP            ;TYPE BYPASS DRIVE#
3612  014726  104006                          ERROR   6
3613  014730  000735                          BR      12$
3614
3615  014732  004737  032736          6$:     JSR     PC,BYP            ;TYPE BYPASS DRIVE#
3616  014736  104002                          ERROR   2                 ;MDS DETECTED
3617  014740  000762                          BR      8$
3618
3619  014742  004737  032736          7$:     JSR     PC,BYP
3620  014746  104003                          ERROR   3                 ;UFE DETECTED
3621  014750  000756                          BR      8$
3622
```

```
                                9$:     JSR     PC,BYP
                                        ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
                                        BR      8$
                                        BNE     1$              ;BRANCH IF MORE LEFT.
                                        ;
                                        ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
                                        ;DRIVE PRESENT
                                        ;
                                        ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
                                        ;UNDER TEST
                                NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
                                                                ;ALLOW CHECKING CERR IN 'FRDY'
                                ;;*****************************************************************
                                ;*TEST 4        FIND NEXT DRIVE TO BE TESTED
                                ;*
                                ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                ;*      ADDRESS IN '$UNIT'.
                                ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                                ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
                                ;*
                                ;;*****************************************************************
014770  000004                  TST4:   SCOPE
014772  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
015000  012706  001100                  MOV     #$STACK,SP      ;RESTORE STK PTR
015004  012737  000004  001214          MOV     #$TN-1,$TESTN
015012  012737  000004  001102          MOV     #$TN-1,$TSTNM
015020  005737  007460                  TST     DRIVS           ;ANY DRIVES PRESENT?
015024  001004                          BNE     4$              ;YES BRANCH
015026  104401  045355                  TYPE    ,MSG27          ;ALL DRIVES TESTED
015032  000137  031550                  JMP     $EOP1           ;NO, GO TO END
015036  013701  001342          4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
015042  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
015046  001402                          BEQ     2$              ;YES, BRANCH
015050  005237  001222          1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
015054  005721                  2$:     TST     (R1)+           ;IS DRIVE PRESENT?
015056  001774                          BEQ     1$              ;NO,FIND NEXT DRIVE PRESENT
015060  005737  007452                  TST     DDPCH           ;DDP CHAIN MODE?
015064  001403                          BEQ     3$              ;NO, BRANCH
015066  005737  001222                  TST     $UNIT           ;YES, IS IT DRIVE 0?
015072  001766                          BEQ     1$              ;IF YES, DON'T TEST DR 0
015074  010137  001342          3$:     MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT DR. FLAG
015100  104401  045211                  TYPE    ,MSG15          ;"DRIVE"
015104  013700  001222                  MOV     $UNIT,R0
015110  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;DRIVE #
015112  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
015114     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
015115     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
                                        TYPE    ,$CRLF

                                PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
                                ;;*****************************************************************
                                ;*TEST 5         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
                                ;*
                                ;;*****************************************************************
015122  000004                  TST5:   SCOPE
015124  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
015132  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

015136  005737  001216                  TST     $PASS
015142  001046                          BNE     TST6            ;;GO TO NEXT IF NOT FIRST PASS
015144  004737  034140                  JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
015150  104024                          ERROR   24              ;CERR AFTER SCLR

015152  104401  045223                  TYPE    ,MSG16          ;DRIVE SERIAL NO.
015156  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
015164  004737  033566                  JSR     PC,GSTAT        ;GET STATUS
015170  013701  007366                  MOV     HMR2,R1         ;GET SERIAL #
015174  012704  042546                  MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
015200  010446                          MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
015202  012703  000003                  MOV     #3,R3           ;SETUP CHAR COOUNT
015206  006101                          ROL     R1              ;INITIALIZE BIT POSITIONS
015210  006101                          ROL     R1
015212  006101                  1$:     ROL     R1              ;GET NEXT 4 BITS
015214  006101                          ROL     R1
015216  006101                          ROL     R1
015220  006101                          ROL     R1
015222  010100                          MOV     R1,R0           ;GET WORKING COPY
015224  042700  177760                  BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
015230  052700  000060                  BIS     #60,R0          ;CONVERT TO ASCII DIGIT
015234  110024                          MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
015236  005303                          DEC     R3
015240  001364                          BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
015242  105014                          CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR

015244  004737  043014                  JSR     PC,$SUPRS       ;TYPE
015250  104401  001205                  TYPE    ,$CRLF
015254  104401  001205                  TYPE    ,$CRLF

                                ;;*****************************************************************
                                ;*TEST 6         SET VV WITH PACK COMMAND
                                ;*
                                ;*      IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
                                ;*
                                ;;*****************************************************************
015260  000004                  TST6:   SCOPE
015262  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
015270  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

015274  004737  034140                  JSR     PC,SUBCLR
015300  104024                          ERROR   24              ;CERR AFTER SCLR
```

```
3735
3736  015302  032737  000100  007366          BIT     #D.VV,HMR2
3737  015310  001024                          BNE     TST7                    ;;GO TO NEXT TEST IF VV SET
3738
3739  015312  104415                          SCOP1
3740  015314  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR
3741
3742  015320  004737  034140                  JSR     PC,SUBCLR
3743  015324  104024                          ERROR   24                      ;CERR AFTER SCLR
3744
3745  015326  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;CMD TO SET VV
3746  015334  012737  000010  007376          MOV     #10,TEMP1
3747  015342  004737  032230                  JSR     PC,FRDY                 ;FIND RDY
3748  015346  104116                          ERROR   116                     ;RDY NOT SET AFTER PACK CMD
3749
3750  015350  032737  000100  007366          BIT     #D.VV,HMR2
3751  015356  001001                          BNE     TST7                    ;;GO TO NEXT TEST IF VV NOW SET
3752  015360  104027                          ERROR   27                      ;PACK DID NOT SET V.V.
3753
3754
3755
3756                                  ;;*****************************************************************
3757                                  ;*TEST 7        READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
3758                                  ;*
3759                                  ;*      THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
3760                                  ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
3761                                  ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
3762                                  ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
3763                                  ;*
3764                                  ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
3765                                  ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
3766                                  ;*
3767                                  ;*      SECTORS 1,3,5,7,9 CONTAIN IDENTICAL INFO FOR 20 SECTOR HARDWARE DETECTED BAD SEC
3768                                  ;*      SECTORS 11,13,15,17,19,21 CONTAIN IDENTICAL INFO FOR 20 SECTOR SOFTWARE DETECTED
3769                                  ;*
3770                                  ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
3771                                  ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
3772                                  ;*      A MESSAGE WILL BE TYPED INDICATING THAT ALL
3773                                  ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
3774                                  ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
3775                                  ;*
3776                                  ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
3777                                  ;*
3778                                  ;*****************************************************************
3779  015362  000004                  TST7:   SCOPE
3780  015364  012737  000001  001174          MOV     #1,$TIMES               ;;DO 1 ITERATION
3781  015372  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR
3782
3783  015376  004737  034140                  JSR     PC,SUBCLR
3784  015402  104024                          ERROR   24                      ;CERR AFTER SCLR
3785
3786
3787  015404  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3788  015412  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
3789  015420  012765  000013  000000          MOV     #RECAL,RKCS1(R5)                ;RECAL CMD
                                                                                ;RESET CYL DIFF/OFFSET & CYL ADDR REG
3790
```

```
3791                                                    ;IN RKMR2 & RKMR3 RESP.
3792  015426  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
3793  015434  004737  032230                  JSR     PC,FRDY         ;FIND RDY
3794  015440  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
3795
3796  015442  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3797  015450  004737  033566                  JSR     PC,GSTAT
3798  015454  032737  020000  007366          BIT     #D.RTZ,HMR2
3799  015462  001001                          BNE     64$
3800  015464  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
3801  015466  013737  001406  007400  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
3802  015474  004737  032544                  JSR     PC,FATT1        ;FIND ATTN
3803  015500  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
3804
3805  015502  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3806  015510  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
3807  015516  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3808  015524  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
3809  015532  004737  032230                  JSR     PC,FRDY         ;FIND RDY
3810  015536  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3811  015540  004737  032512                  JSR     PC,TSTATN       ;TEST FOR ATTN
3812  015544  000401                          BR      65$
3813  015546  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3814  015550                          65$:
3815
3816
3817
3818  015550  004737  034140                  JSR     PC,SUBCLR
3819  015554  104024                          ERROR   24              ;CERR AFTER SCLR
3820
3821  015556  005037  007400                  CLR     TEMP2           ;SECTOR CTR
3822  015562  005037  007402                  CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
3823                                                                  ;1=22 SECTOR SOFTWARE DETECTED TABLE
3824                                                                  ;2=20 SECTOR HARDWARE DETECTED TABLE
3825                                                                  ;3=20 SECTOR SOFTWARE DETECTED TABLE
3826  015566  012737  003322  007404          MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE ADDR.
3827  015574  013765  007404  000004          MOV     TEMP4,RKBA(R5)
3828  015602  012737  001000  007406          MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
3829  015610  013765  007406  000006          MOV     TEMP5,RKDA(R5)
3830
3831  015616  012765  000632  000020  1$:     MOV     #410.,RKDC(R5)  ;CYL 410
3832  015624  012765  177400  000002          MOV     #-256.,RKWC(R5) ;LOAD WORD CT
3833  015632  012765  000021  000000          MOV     #RDDATA,RKCS1(R5);READ DATA COMMAND
3834  015640  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
3835  015646  004737  032230                  JSR     PC,FRDY         ;FIND RDY
3836  015652  104226                          ERROR   226             ;NO RDY AFTER READ DATA CMD
3837  015654  004737  033566                  JSR     PC,GSTAT        ;GET FRESH DATA
3838  015660  032737  100000  007340          BIT     #CERR,HCS1
3839  015666  001504                          BEQ     8$
3840  015670  104227                          ERROR   227             ;CERR AFTER READ DATA CMD
3841
3842  015672  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
3843  015700  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
3844  015704  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
3845  015712  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
3846  015720  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
```

```
3847  015724  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
3848  015732  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
3849
3850  015740  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
3851  015744  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
3852  015746  104054                          ERROR   52              ;MSG A0 ERROR AFTER READ DATA CMD
3853  015750  104026                          ERROR   [illegible]     ;MSG B0 ERROR
3854  015752  104056                          ERROR   56              ;MSG A1 ERROR
3855  015754  104030                          ERROR   30              ;MSG B1 ERROR
3856
3857  015756  004737  034140                  JSR     PC,SUBCLR
3858  015762  104024                          ERROR   24              ;CERR AFTER SUBCLR
3859
3860  015764  005237  007400                  INC     TEMP2
3861  015770  023727  007400  000005          CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
3862  015776  001023                          BNE     5$
3863  016000  005737  007402                  TST     TEMP3
3864  016004  001002                          BNE     2$
3865  016006  104233                          ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
3866  016010  000430                          BR      3$
3867
3868  016012  023727  007402  000001  2$:     CMP     TEMP3,#1
3869  016020  001002                          BNE     4$
3870  016022  104230                          ERROR   230             ;CANT READ SECTORS 10,12...
3871  016024  000422                          BR      3$
3872
3873  016026  023727  007402  000002  4$:     CMP     TEMP3,#2
3874  016034  001002                          BNE     6$
3875  016036  104234                          ERROR   234             ;CANT READ SECTORS 1,3,5 ...
3876  016040  000414                          BR      3$
3877
3878  016042  104231                  6$:     ERROR   231             ;CANT READ SECTORS 11,13,15 ...
3879  016044  000412                          BR      3$
3880
3881  016046  013765  007404  000004  5$:     MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
3882  016054  062737  000002  007406          ADD     #2,TEMP5        ;READ 2 SECTORS FROM LAST
3883  016062  013765  007406  000006          MOV     TEMP5,RKDA(R5)
3884  016070  000652                          BR      1$
3885
3886  016072  005237  001476          3$:     INC     BSERR           ;SET BSE FLAG
3887  016076  000553                          BR      TST10           ;;GO TO NEXT TEST
3888  016100  005737  003330          8$:     TST     BSE22H+6        ;TEST CARTRIDGE TYPE
3889  016104  001404                          BEQ     9$              ;BRANCH IF DATA CARTRIDGE
3890  016106  104235                          ERROR   235             ;ALIGNMENT CARTRIDGE USED
3891  016110  005237  001476                  INC     BSERR           ;SET BSE ERROR FLAG
3892  016114  000476                          BR      10$
3893
3894  016116  005237  007402          9$:     INC     TEMP3
3895  016122  023727  007402  000001          CMP     TEMP3,#1
3896  016130  001020                          BNE     11$
3897  016132  005037  007400                  CLR     TEMP2           ;SECTOR CTR
3898  016136  012737  005322  007404          MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
3899  016144  013765  007404  000004          MOV     TEMP4,RKBA(R5)
3900  016152  012737  001012  007406          MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12(8)
3901  016160  013765  007406  000006          MOV     TEMP5,RKDA(R5)
3902  016166  000137  015616                  JMP     1$              ;REPEAT
```

```
3903
3904  016172  023727  007402  000002  11$:    CMP     TEMP3,#2
3905  016200  001020                          BNE     12$
3906  016202  005037  007400                  CLR     TEMP2           ;SECTOR CTR
3907  016206  012737  002322  007404          MOV     #BSE20H,TEMP4   ;STORE 20 SECTOR HARDWARE BSE ADDR.
3908  016214  013765  007404  000004          MOV     TEMP4,RKBA(R5)
3909  016222  012737  001001  007406          MOV     #1001,TEMP5     ;TRACK 2, SECTOR 1
3910  016230  013765  007406  000006          MOV     TEMP5,RKDA(R5)
3911  016236  000137  015616                  JMP     1$              ;REPEAT
3912
3913  016242  023727  007402  000003  12$:    CMP     TEMP3,#3
3914  016250  001020                          BNE     10$
3915  016252  005037  007400                  CLR     TEMP2           ;SECTOR CTR
3916  016256  012737  004322  007404          MOV     #BSE20S,TEMP4   ;STORE 20 SECTOR SOFTWARE BSE ADDR
3917  016264  013765  007404  000004          MOV     TEMP4,RKBA(R5)
3918  016272  012737  001013  007406          MOV     #1013,TEMP5     ;TRACK 2, SECTOR 13(8)
3919  016300  013765  007406  000006          MOV     TEMP5,RKDA(R5)
3920  016306  000137  015616                  JMP     1$              ;REPEAT
3921
3922  016312  005737  001216          10$:    TST     $PASS
3923  016316  001043                          BNE     TST10           ;;GO TO NEXT TST IF NOT 1'ST PASS
3924  016320  104401  045247                  TYPE    ,MSG17          ;CART SERIAL #
3925  016324  012746  003322                  MOV     #BSE22H,-(SP)
3926  016330  004737  042444                  JSR     PC,$DB20        ;CONVERT DBL BINARY WORD TO OCTAL
3927  016334  004737  043014                  JSR     PC,$SUPRS       ;TYPE SERIAL #
3928  016340  104401  001205                  TYPE    ,$CRLF
3929  016344  104401  001205                  TYPE    ,$CRLF
3930
3931  016350  004737  034140                  JSR     PC,SUBCLR
3932  016354  104024                          ERROR   24              ;CERR AFTER SCLR
3933                                                                  ;GO BACK TO CYL 0
3934
3935  016356  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
3936  016364  013737  001410  007376          MOV     T50,TEMP1       ;SETUP TIMEOUT
3937  016372  004737  032230                  JSR     PC,FRDY         ;FIND RDY
3938  016376  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
3939
3940  016400  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
3941  016406  004737  032640                  JSR     PC,FATT2        ;FIND ATTN
3942  016412  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
3943
3944  016414  032737  100000  007340          BIT     #CERR,HCS1
3945  016422  001401                          BEQ     66$
3946  016424  104210                          ERROR   210             ;CERR AFTER SEEK CMD
3947
3948  016426                          66$:
3949
3950
3951
3952                                  .SBTTL WRITE TESTS
3953
3954
3955                                  ;;*****************************************************************
3956                                  ;*TEST 10       BASIC WRITE DATA TEST; 1 WORD
3957                                  ;*
3958                                  ;*      THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,
```

```
                                        ;*      ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
                                        ;*      A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
                                        ;*      PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
                                        ;*      FOR A WRITE ERROR.
                                        ;*
                                        ;**********************************************************************
016426  000004                  TST10:  SCOPE
016430  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
016436  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

016442  005737  001476                  TST     BSERR           ;SEE IF ALIGN CART
016446  001406                          BEQ     2$              ;BR IF NO
016450  104401  045605                  TYPE    ,MSG40          ;BSE OR ALIGN CART USED
016454  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
016460  000137  031514                  JMP     $EOP

016464  004737  034140          2$:     JSR     PC,SUBCLR
016470  104024                          ERROR   24              ;CERR AFTER SCLR

016472  005237  007324                  INC     BADHDR          ;USED FOR VALID HALT

016476  012700  001506                  MOV     #HDTAB,R0       ;MAKE ALL CYL 0 HEADERS IDENTICAL

016502  005020                  1$:     CLR     (R0)+           ;HEADER WORD 0: CYL 0
016504  012720  140000                  MOV     #140000,(R0)+   ;HEADER WORD 1: SECTOR 0
016510  012720  140000                  MOV     #140000,(R0)+   ;HEADER WORD 2: XOR OF 1 & 2

016514  020027  001712                  CMP     R0,#HDTAB+132.  ;ALL HEADERS DONE? (22X6=132)
016520  001370                          BNE     1$              ;BR IF NO

016522  012765  001506  000004          MOV     #HDTAB,RKBA(R5) ;HEADER TABLE
016530  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT

016536  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
016544  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
016552  004737  032230                  JSR     PC,FRDY         ;FIND RDY
016556  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
016560  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
016564  032737  100000  007340          BIT     #CERR,HCS1
016572  001405                          BEQ     64$
016574  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
016576  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
016602  000137  031514                  JMP     $EOP
016606                          64$:

016606  104415                  SCOP1
016610  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

016614  004737  034140                  JSR     PC,SUBCLR
016620  104024                          ERROR   24              ;CERR AFTER SCLR

016622  005037  001402                  CLR     SECTOR
016626  013765  001402  000006  3$:     MOV     SECTOR,RKDA(R5) ;TRACK/SECTOR #
016634  012765  001464  000004          MOV     #DATA1,RKBA(R5) ;DATA TO BE ALL 1'S
016642  012765  177777  000002          MOV     #-1,RKWC(R5)    ;WORD COUNT=1
```

```
4015
4016
4017  016650  012765  000023  000000          MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
4018  016656  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4019  016664  004737  032230                  JSR     PC,FRDY         ;FIND RDY
4020  016670  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4021  016672  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
4022  016676  032737  100000  007340          BIT     #CERR,HCS1
4023  016704  001465                          BEQ     68$             ;BR IF NO ERRORS
4024
4025  016706  032737  000200  007354          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
4026  016714  001421                          BEQ     66$             ;BR IF NO
4027  016716  004737  035610                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4028  016722  000455                          BR      67$             ;RETURN HERE IF NO
4029
4030  016724  005237  001402                  INC     SECTOR          ;RETURN HERE IF YES
4031  016730  023727  001402  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4032  016736  001003                          BNE     65$             ;BR IF NO
4033  016740  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
4034  016742  000137  017144                  JMP     5$              ;BYPASS TEST
4035  016746  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4036  016754  000137  016626                  JMP     3$
4037  016760  104012                  66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
4038
4039  016762  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4040  016770  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4041  016774  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4042  017002  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4043  017010  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4044  017014  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4045  017022  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4046
4047  017030  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4048  017034  000003                          .WORD   T.A2!T.B2!0             ;& MSG$ SPECIFIED HERE
4049  017036  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4050  017040  104023                          ERROR   23              ;MSH B0 ERROR
4051  017042  104053                          ERROR   53              ;MSG A1 ERROR
4052  017044  104025                          ERROR   25              ;MSG B1 ERROR
4053  017046  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4054  017052  000137  031514                  JMP     $EOP
4055  017056  104063                  67$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
4056  017060                          68$:
4057
4058  017060  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4059  017066  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4060  017072  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4061  017100  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4062  017106  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4063  017112  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4064  017120  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4065
4066  017126  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4067  017132  000003                          .WORD   T.A2!T.B2!0             ;& MSG$ SPECIFIED HERE
4068  017134  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4069  017136  104023                          ERROR   23              ;MSH B0 ERROR
4070  017140  104053                          ERROR   53              ;MSG A1 ERROR
```

```
                        ERROR   25              ;MSG B1 ERROR
                    5$:
                    ;;*****************************************************************
                    ;*TEST 11        BASIC WRITE DATA TEST; FULL SECTOR
                    ;*
                    ;*      THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
                    ;*      A FULL SECTOR.  ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
                    ;*      & CHECKED BY A RD DATA COMMAND.  A FURTHER CHECK IS PERFORMED
                    ;*      BY THE WRT CHK COMMAND.
                    ;*      THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.
                    ;*
                    ;;*****************************************************************
017144  000004      TST11:  SCOPE
017146  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
017154  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR

017160  004737  034140              JSR     PC,SUBCLR
017164  104024                      ERROR   24              ;CERR AFTER SCLR

017166  012765  001506  000004      MOV     #HDTAB,RKBA(R5) ;RESTORE TO 22 SECTOR
017174  012765  177676  000002      MOV     #-66.,RKWC(R5)  ;STANDARD FORMAT
017202  005037  001346              CLR     TOCYL

017206  013737  001346  001362      MOV     TOCYL,CALADD    ;SETUP
017214  012737  000000  001444      MOV     #0,HEAD         ;TO FILL
017222  012737  000000  001452      MOV     #0,FORMAT       ;HEADER
017230  004737  035122              JSR     PC,FHDTAB       ;TABLE

017234  012765  000027  000000      MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
017242  013737  001420  007376      MOV     T50000,TEMP1    ;SETUP  TIMEOUT
017250  004737  032230              JSR     PC,FRDY         ;FIND RDY
017254  104200                      ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
017256  004737  033566              JSR     PC,GSTAT        ;GET FRESH STATUS
017262  032737  100000  007340      BIT     #CERR,HCS1
017270  001405                      BEQ     64$
017272  104201                      ERROR   201             ;CERR AFTER WRITE HEADER CMD
017274  104401  045277              TYPE    ,MSG26          ;ABORTING BAL OF TESTS
017300  000137  031514              JMP     $EOP
017304                      64$:

017304  005037  007324              CLR     BADHDR          ;USED FOR VALID HALT
017310  104415                      SCOP1
017312  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR

017316  004737  034140              JSR     PC,SUBCLR
017322  104024                      ERROR   24              ;CERR AFTER SCLR

017324  005037  001402              CLR     SECTOR
017330  013765  001402  000006  8$: MOV     SECTOR,RKDA(R5) ;SETUP SECTOR
017336  012765  001460  000004      MOV     #DATAO,RKBA(R5) ;WRITE ALL 0'S
017344  013700  001460              MOV     DATAO,R0
017350  052765  000020  000010  1$: BIS     #BAI,RKCS2(R5)
017356  012765  177400  000002      MOV     #-256.,RKWC(R5)

017364  012765  000023  000000      MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
```

```
4127  017372  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4128  017400  004737  032230                  JSR     PC,FRDY         ;FIND RDY
4129  017404  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4130  017406  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
4131  017412  032737  100000  007340          BIT     #CERR,HCS1
4132  017420  001465                          BEQ     68$             ;BR IF NO ERRORS
4133
4134  017422  032737  000200  007354          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
4135  017430  001421                          BEQ     66$             ;BR IF NO
4136  017432  004737  035610                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4137  017436  000455                          BR      67$             ;RETURN HERE IF NO
4138
4139  017440  005237  001402                  INC     SECTOR          ;RETURN HERE IF YES
4140  017444  023727  001402  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4141  017452  001003                          BNE     65$             ;BR IF NO
4142  017454  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
4143  017456  000137  020752                  JMP     7$              ;BYPASS TEST
4144  017462  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4145  017470  000137  017330                  JMP     8$
4146  017474  104012                  66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
4147
4148  017476  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4149  017504  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4150  017510  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4151  017516  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4152  017524  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4153  017530  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4154  017536  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4155
4156  017544  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4157  017550  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4158  017552  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4159  017554  104023                          ERROR   23              ;MSH B0 ERROR
4160  017556  104053                          ERROR   53              ;MSG A1 ERROR
4161  017560  104025                          ERROR   25              ;MSG B1 ERROR
4162  017562  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4163  017566  000137  031514                  JMP     $EOP
4164  017572  104063                  67$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
4165  017574                          68$:
4166
4167  017574  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4168  017602  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4169  017606  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4170  017614  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4171  017622  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4172  017626  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4173  017634  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4174
4175  017642  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4176  017646  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4177  017650  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4178  017652  104023                          ERROR   23              ;MSH B0 ERROR
4179  017654  104053                          ERROR   53              ;MSG A1 ERROR
4180  017656  104025                          ERROR   25              ;MSG B1 ERROR
4181  017660  104415                          SCOP1
4182  017662  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
```

```
4183
4184  017666  004737  034140          JSR     PC,SUBCLR
4185  017672  104024                  ERROR   24                      ;CERR AFTER SCLR
4186
4187  017674  013765  001402  000006  MOV     SECTOR,RKDA(R5) ;SETUP SECTOR
4188  017702  012765  006322  000004  MOV     #RDTAB,RKBA(R5)
4189  017710  012765  177400  000002  MOV     #-256.,RKWC(R5)
4190
4191
4192  017716  012765  000021  000000  MOV     #<RDDATA>,RKCS1(R5)     ;READ DATA CMD
4193  017724  013737  001420  007376  MOV     T50000,TEMP1    ;SETUP TIMEOUT
4194  017732  004737  032230          JSR     PC,FRDY         ;FIND RDY
4195  017736  104013                  ERROR   13              ;NO RDY AFTER READ DATA CMD
4196  017740  004737  033566          JSR     PC,GSTAT        ;GET FRESH STATUS
4197  017744  032737  100000  007340  BIT     #CERR,HCS1
4198  017752  001454                  BEQ     72$
4199  017754  032737  000200  007354  BIT     #BSE,HER        ;SEE IF BAD SECTOR
4200  017762  001406                  BEQ     70$
4201  017764  104065                  ERROR   65              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
4202  017766  000413                  BR      73$
4203  017770  104401  045277  69$:    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4204  017774  000137  031514          JMP     $EOP
4205
4206  020000  032737  100000  007354  70$:    BIT     #DCK,HER        ;SEE IF DATA CHECK ERROR
4207  020006  001402                  BEQ     71$
4208  020010  104021                  ERROR   21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
4209  020012  000401                  BR      73$
4210
4211  020014  104014          71$:    ERROR   14              ;CERR AFTER READ DATA CMD.
4212
4213  020016                  73$:
4214
4215  020016  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4216  020024  005037  007432          CLR     E.B0            ;EXPECTED MSG B0
4217  020030  012737  001720  007434  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4218  020036  012737  000001  007436  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4219  020044  005037  007440          CLR     E.A2            ;EXPECTED MSG A2
4220  020050  012737  000002  007442  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4221  020056  012737  000003  007446  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4222
4223  020064  004737  032752          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4224  020070  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4225  020072  104054                  ERROR   54              ;MSG A0 ERROR AFTER READ DATA CMD
4226  020074  104026                  ERROR   26              ;MSH B0 ERROR
4227  020076  104056                  ERROR   56              ;MSG A1 ERROR
4228  020100  104030                  ERROR   30              ;MSG B1 ERROR
4229  020102  000732                  BR      69$
4230  020104                  72$:
4231
4232  020104  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4233  020112  005037  007432          CLR     E.B0            ;EXPECTED MSG B0
4234  020116  012737  001720  007434  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4235  020124  012737  000001  007436  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4236  020132  005037  007440          CLR     E.A2            ;EXPECTED MSG A2
4237  020136  012737  000002  007442  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4238  020144  012737  000003  007446  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
```

```
4239
4240  020152  004737  032752          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4241  020156  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4242  020160  104054                  ERROR   54              ;MSG A0 ERROR AFTER READ DATA CMD
4243  020162  104026                  ERROR   26              ;MSH B0 ERROR
4244  020164  104056                  ERROR   56              ;MSG A1 ERROR
4245  020166  104030                  ERROR   30              ;MSG B1 ERROR
4246
4247  020170  012701  006322          MOV     #RDTAB,R1
4248  020174  011137  001436  2$:     MOV     (R1),WD1        ;ACTUAL WORD FOR TYPEOUT
4249  020200  010037  001440          MOV     R0,WD2          ;EXPECTED DATA FOR TYPEOUT
4250
4251  020204  020021                  CMP     R0,(R1)+        ;COMPARE READ DATA TABLE AGAINST
4252  020206  001401                  BEQ     3$              ;THE 'SHOULD BE' VALUE
4253  020210  104020                  ERROR   20              ;READ DATA DID NOT COMPARE WITH WRITE DATA
4254
4255  020212  020127  007322  3$:     CMP     R1,#RDTAB+512.  ;SEE IF REACHED END OF TABLE
4256  020216  001366                  BNE     2$              ;BR IF NO & DO NEXT WORD
4257
4258  020220  020037  001460          CMP     R0,DATA0        ;SEE IF DID ALL 0'S
4259  020224  001401                  BEQ     4$              ;BR IF YES
4260  020226  000412                  BR      5$
4261
4262  020230  012765  001464  000004  4$:     MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
4263  020236  013700  001464          MOV     DATA1,R0
4264  020242  013765  001402  000006  MOV     SECTOR,RKDA(R5)
4265  020250  000137  017350          JMP     1$
4266
4267  020254                  5$:
4268  020254  104415                  SCOP1
4269  020256  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
4270
4271  020262  004737  034140          JSR     PC,SUBCLR
4272  020266  104024                  ERROR   24              ;CERR AFTER SCLR
4273
4274  020270  052765  000020  000010  BIS     #BAI,RKCS2(R5)  ;THIS PORTION OF THE TEST CHECKS
4275  020276  012765  001464  000004  MOV     #DATA1,RKBA(R5) ;OUT THE WRITE CHECK CMD
4276  020304  012765  177400  000002  MOV     #-256.,RKWC(R5) ;ALL 1'S WERE PREVIOUSLY WRITTEN
4277  020312  013765  001402  000006  MOV     SECTOR,RKDA(R5)
4278
4279  020320  012765  000031  000000  MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4280  020326  013737  001420  007376  MOV     T50000,TEMP1    ;SETUP TIMEOUT
4281  020334  004737  032230          JSR     PC,FRDY         ;FIND RDY
4282  020340  104015                  ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4283  020342  004737  033566          JSR     PC,GSTAT        ;GET FRESH STATUS
4284  020346  032737  100000  007340  BIT     #CERR,HCS1
4285  020354  001453                  BEQ     75$
4286  020356  032737  040000  007342  BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
4287  020364  001410                  BEQ     74$
4288  020366  016537  000024  001436  MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
4289  020374  013737  001464  001440  MOV     DATA1,WD2               ;EXPECTED WORD FOR TYPEOUT
4290  020402  104016                  ERROR   16              ;WCE AFTER WRITE CMD
4291  020404  000437                  BR      75$
4292
4293  020406  104022          74$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4294
```

```
4295  020410  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4296  020416  005037  007432                  CLR     E.B0                ;EXPECTED MSG B0
4297  020422  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4298  020430  012737  000001  007436          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
4299  020436  005037  007440                  CLR     E.A2                ;EXPECTED MSG A2
4300  020442  012737  000002  007442          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
4301  020450  012737  000003  007446          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
4302
4303  020456  004737  032752                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
4304  020462  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
4305  020464  104057                          ERROR   57                  ;MSG A0 ERROR AFTER WRITE CHECK CMD
4306  020466  104031                          ERROR   31                  ;MSH B0 ERROR
4307  020470  104060                          ERROR   60                  ;MSG A1 ERROR
4308  020472  104032                          ERROR   32                  ;MSG B1 ERROR
4309  020474  104401  045277                  TYPE    ,MSG26              ;ABORTING BAL OF TESTS
4310  020500  000137  031514                  JMP     $EOP
4311
4312  020504                          75$:
4313
4314  020504  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4315  020512  005037  007432                  CLR     E.B0                ;EXPECTED MSG B0
4316  020516  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4317  020524  012737  000001  007436          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
4318  020532  005037  007440                  CLR     E.A2                ;EXPECTED MSG A2
4319  020536  012737  000002  007442          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
4320  020544  012737  000003  007446          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
4321
4322  020552  004737  032752                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
4323  020556  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
4324  020560  104057                          ERROR   57                  ;MSG A0 ERROR AFTER WRITE CHECK CMD
4325  020562  104031                          ERROR   31                  ;MSH B0 ERROR
4326  020564  104060                          ERROR   60                  ;MSG A1 ERROR
4327  020566  104032                          ERROR   32                  ;MSG B1 ERROR
4328
4329  020570  104415                          SCOP1
4330  020572  012706  001100                  MOV     #STACK,SP           ;RESTORE STK PTR
4331
4332  020576  004737  034140                  JSR     PC,SUBCLR
4333  020602  104024                          ERROR   24                  ;CERR AFTER SCLR
4334
4335  020604  012765  001460  000004          MOV     #DATA0,RKBA(R5)     ;SETUP TO CHECK AGAINST WRONG DATA
4336  020612  012765  177400  000002          MOV     #-256.,RKWC(R5)
4337  020620  013765  001402  000006          MOV     SECTOR,RKDA(R5)
4338  020626  012765  000031  000000          MOV     #WRTCHK,RKCS1(R5)
4339  020634  012737  141520  007376          MOV     #50000.,TEMP1
4340  020642  004737  032230                  JSR     PC,FRDY
4341  020646  104015                          ERROR   15                  ;NO RDY AFTER WRITE CHECK CMD
4342  020650  004737  033566                  JSR     PC,GSTAT            ;GET FRESH STATUS
4343  020654  032737  040000  007342          BIT     #WCE,HCS2           ;EXPECT MISCOMPARE
4344  020662  001001                          BNE     6$
4345  020664  104017                          ERROR   17                  ;WRITE CHECK CMD NOT FUNCTIONING
4346                                                                      ;WITH INTENTIONAL MISCOMPARE
4347  020666                          6$:
4348
4349  020666  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4350  020674  005037  007432                  CLR     E.B0                ;EXPECTED MSG B0
```

```
4351  020700  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4352  020706  012737  000001  007436          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
4353  020714  005037  007440                  CLR     E.A2                ;EXPECTED MSG A2
4354  020720  012737  000002  007442          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
4355  020726  012737  000003  007446          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
4356
4357  020734  004737  032752                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
4358  020740  000000                          .WORD   0!0!0               ;& MSGS SPECIFIED HERE
4359  020742  104057                          ERROR   57                  ;MSG A0 ERROR AFT WRT CHK CMD
4360  020744  104031                          ERROR   31                  ;MSH B0 ERROR
4361  020746  104060                          ERROR   60                  ;MSG A1 ERROR
4362  020750  104032                          ERROR   32                  ;MSG B1 ERROR
4363  020752                          7$:
4364
4365                                  ;;*****************************************************************
4366                                  ;*TEST 12       20 SECTOR FORMAT TEST
4367                                  ;*
4368                                  ;*      ALL 1'S ARE WRITTEN ON A FULL SECTOR IN 20 SECTOR FORMAT.
4369                                  ;*      MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
4370                                  ;*      CYL 0, TRACK 0, & SECTOR 0 IS USED.
4371                                  ;*
4372                                  ;;*****************************************************************
4373  020752  000004                  TST12:  SCOPE
4374  020754  012737  000001  001174          MOV     #1,$TIMES           ;;DO 1 ITERATION
4375  020762  012706  001100                  MOV     #STACK,SP           ;RESTORE STK PTR
4376
4377  020766  004737  034140                  JSR     PC,SUBCLR
4378  020772  104024                          ERROR   24                  ;CERR AFTER SCLR
4379
4380  020774  012765  001506  000004          MOV     #HDTAB,RKBA(R5)     ;HEADER WORD TABLE
4381  021002  012765  177704  000002          MOV     #-60.,RKWC(R5)      ;WORD COUNT FOR 20 SECTOR FMT
4382  021010  005037  001346                  CLR     TOCYL
4383  021014  005237  007324                  INC     BADHDR              ;USED FOR VALID HALT
4384
4385
4386  021020  013737  001346  001362          MOV     TOCYL,CALADD        ;SETUP
4387  021026  012737  000000  001444          MOV     #0,HEAD             ;TO FILL
4388  021034  012737  000001  001452          MOV     #1,FORMAT           ;HEADER
4389  021042  004737  035122                  JSR     PC,FHDTAB           ;TABLE
4390
4391
4392  021046  012765  010027  000000          MOV     #<CFMT!WRHEAD>,RKCS1(R5)        ;WRITE HEADER CMD
4393  021054  013737  001420  007376          MOV     T50000,TEMP1        ;SETUP  TIMEOUT
4394  021062  004737  032230                  JSR     PC,FRDY             ;FIND RDY
4395  021066  104200                          ERROR   200                 ;NO RDY AFTER WRITE HEADER CMD
4396  021070  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)
4397  021076  013737  001406  007376          MOV     T10,TEMP1
4398  021104  004737  032230                  JSR     PC,FRDY
4399  021110  104117                          ERROR   117                 ;NO RDY AFTER SELDRV CMD
4400  021112  032737  100000  007340          BIT     #CERR,HCS1
4401  021120  001405                          BEQ     64$
4402  021122  104201                          ERROR   201                 ;CERR AFTER WRITE HEADER CMD
4403  021124  104401  045277                  TYPE    ,MSG26              ;ABORTING BAL OF TESTS
4404  021130  000137  031514                  JMP     $EOP
4405  021134                          64$:
4406
```

```
4407  021134  104415                          SCOP1
4408  021136  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4409
4410  021142  004737  034140                  JSR     PC,SUBCLR
4411  021146  104024                          ERROR   24              ;CERR AFTER SCLR
4412
4413  021150  005037  001402                  CLR     SECTOR
4414  021154  013765  001402  000006  4$:     MOV     SECTOR,RKDA(R5)
4415  021162  012765  001464  000004          MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
4416  021170  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUSS ADDR INCR INHIBIT
4417  021176  012765  177400  000002          MOV     #-256.,RKWC(R5)  ;DO FULL SECTOR
4418
4419
4420  021204  012765  010023  000000          MOV     #<CFMT!WRDATA>,RKCS1(R5)        ;WRITE DATA CMD
4421  021212  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4422  021220  004737  032230                  JSR     PC,FRDY         ;FIND RDY
4423  021224  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4424  021226  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)
4425  021234  013737  001406  007376          MOV     T10,TEMP1
4426  021242  004737  032230                  JSR     PC,FRDY
4427  021246  104117                          ERROR   117             ;NO RDY AFTER SELDRV CMD
4428  021250  032737  100000  007340          BIT     #CERR,HCS1
4429  021256  001465                          BEQ     68$             ;BR IF NO ERRORS
4430
4431  021260  032737  000200  007354          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
4432  021266  001421                          BEQ     66$             ;BR IF NO
4433  021270  004737  035610                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4434  021274  000455                          BR      67$             ;RETURN HERE IF NO
4435
4436  021276  005237  001402                  INC     SECTOR          ;RETURN HERE IF YES
4437  021302  023727  001402  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4438  021310  001003                          BNE     65$             ;BR IF NO
4439  021312  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
4440  021314  000137  022256                  JMP     3$              ;BYPASS TEST
4441  021320  012765  100000  000000  65$:     MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4442  021326  000137  021154                  JMP     4$
4443  021332  104012                  66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
4444
4445  021334  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4446  021342  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4447  021346  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4448  021354  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4449  021362  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4450  021366  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4451  021374  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4452
4453  021402  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4454  021406  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4455  021410  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4456  021412  104023                          ERROR   23              ;MSH B0 ERROR
4457  021414  104053                          ERROR   53              ;MSG A1 ERROR
4458  021416  104025                          ERROR   25              ;MSG B1 ERROR
4459  021420  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4460  021424  000137  031514                  JMP     $EOP
4461  021430  104063                  67$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
4462  021432                          68$:
```

```
4463  021432  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)
4464  021440  013737  001406  007376          MOV     T10,TEMP1
4465  021446  004737  032230                  JSR     PC,FRDY         ;FIND RDY
4466  021452  104117                          ERROR   117             ;RDY NOT FOUND AFTER SELDRV CMD
4467  021454  032737  001000  007366          BIT     #D.FORM,HMR2
4468  021462  001001                          BNE     1$
4469  021464  104102                          ERROR   102             ;FORMAT NOT SET
4470
4471  021466                          1$:
4472
4473  021466  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4474  021474  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4475  021500  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4476  021506  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4477  021514  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4478  021520  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4479  021526  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4480
4481  021534  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4482  021540  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4483  021542  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4484  021544  104023                          ERROR   23              ;MSH B0 ERROR
4485  021546  104053                          ERROR   53              ;MSG A1 ERROR
4486  021550  104025                          ERROR   25              ;MSG B1 ERROR
4487  021552  012765  177400  000002          MOV     #-256.,RKWC(R5)
4488  021560  012765  001464  000004          MOV     #DATA1,RKBA(R5)
4489  021566  052765  000020  000010          BIS     #BAI,RKCS2(R5)
4490  021574  013765  001402  000006          MOV     SECTOR,RKDA(R5)
4491
4492  021602  012765  010031  000000          MOV     #<CFMT!WRTCHK>,RKCS1(R5)        ;WRITE CHECK CMD
4493  021610  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4494  021616  004737  032230                  JSR     PC,FRDY         ;FIND RDY
4495  021622  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4496  021624  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)
4497  021632  013737  001406  007376          MOV     T10,TEMP1
4498  021640  004737  032230                  JSR     PC,FRDY
4499  021644  104117                          ERROR   117             ;NO RDY AFTER SELDRV CMD
4500  021646  032737  100000  007340          BIT     #CERR,HCS1
4501  021654  001453                          BEQ     70$
4502  021656  032737  040000  007342          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
4503  021664  001410                          BEQ     69$
4504  021666  016537  000024  001436          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
4505  021674  013737  001464  001440          MOV     DATA1,WD2               ;EXPECTED WORD FOR TYPEOUT
4506  021702  104016                          ERROR   16              ;WCE AFTER WRITE CMD
4507  021704  000437                          BR      70$
4508
4509  021706  104022                  69$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4510
4511  021710  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4512  021716  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
4513  021722  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4514  021730  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4515  021736  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
4516  021742  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4517  021750  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4518
```

```
4519  021756  004737  032752          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4520  021762  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4521  021764  104057                  ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
4522  021766  104031                  ERROR   31              ;MSH B0 ERROR
4523  021770  104060                  ERROR   60              ;MSG A1 ERROR
4524  021772  104032                  ERROR   32              ;MSG B1 ERROR
4525  021774  104401  045277          TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4526  022000  000137  031514          JMP     $EOP
4527
4528  022004                  70$:
4529  022004  012765  010001  000000  MOV     #<CFMT!SELDRV>,RKCS1(R5)
4530  022012  013737  001406  007376  MOV     T10,TEMP1
4531  022020  004737  032230          JSR     PC,FRDY         ;FIND RDY
4532  022024  104117                  ERROR   117             ;NO RDY AFTER SELDRV CMD
4533  022026  032737  001000  007366  BIT     #D.FORM,HMR2
4534  022034  001001                  BNE     2$
4535  022036  104103                  ERROR   103             ;FORMAT NOT SET
4536
4537  022040                  2$:
4538
4539  022040  012737  010340  007430  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4540  022046  005037  007432          CLR     E.B0            ;EXPECTED MSG B0
4541  022052  012737  001720  007434  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4542  022060  012737  000001  007436  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4543  022066  005037  007440          CLR     E.A2            ;EXPECTED MSG A2
4544  022072  012737  000002  007442  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4545  022100  012737  000003  007446  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4546
4547  022106  004737  032752          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4548  022112  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4549  022114  104057                  ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
4550  022116  104031                  ERROR   31              ;MSH B0 ERROR
4551  022120  104060                  ERROR   60              ;MSG A1 ERROR
4552  022122  104032                  ERROR   32              ;MSG B1 ERROR
4553  022124  104415                  SCOP1
4554  022126  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
4555
4556  022132  004737  034140          JSR     PC,SUBCLR
4557  022136  104024                  ERROR   24              ;CERR AFTER SCLR
4558
4559  022140  012765  001506  000004  MOV     #HDTAB,RKBA(R5) ;RESTORE CYL 0 TO 22 SECTOR FMT
4560  022146  012765  177676  000002  MOV     #-66.,RKWC(R5)
4561  022154  005037  001346          CLR     TOCYL
4562
4563
4564  022160  013737  001346  001362  MOV     TOCYL,CALADD    ;SETUP
4565  022166  012737  000000  001444  MOV     #0,HEAD         ;TO FILL
4566  022174  012737  000000  001452  MOV     #0,FORMAT       ;HEADER
4567  022202  004737  035122          JSR     PC,FHDTAB       ;TABLE
4568
4569
4570  022206  012765  000027  000000  MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
4571  022214  013737  001420  007376  MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4572  022222  004737  032230          JSR     PC,FRDY         ;FIND RDY
4573  022226  104200                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4574  022230  004737  033566          JSR     PC,GSTAT        ;GET FRESH STATUS
```

```
4575  022234  032737  100000  007340          BIT     #CERR,HCS1
4576  022242  001405                          BEQ     71$
4577  022244  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
4578  022246  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4579  022252  000137  031514                  JMP     $EOP
4580  022256                          71$:
4581
4582  022256  005037  007324          3$:     CLR     BADHDR          ;USED FOR VALID HALT
4583
4584                                  ;;*****************************************************************
4585                                  ;*TEST 13       TEST OFFSET & RTC LOGIC
4586                                  ;*
4587                                  ;*      THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
4588                                  ;*      THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
4589                                  ;*      'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG
4590                                  ;*      BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
4591                                  ;*      DIFFERENT CYLINDER IS ISSUED
4592                                  ;*      IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
4593                                  ;*      CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
4594                                  ;*      ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED
4595                                  ;*
4596                                  ;*****************************************************************
4597  022262  000004                  TST13:  SCOPE
4598  022264  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4599  022272  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4600
4601  022276  012702  000001                  MOV     #1,R2           ;MIN POS OFFSET
4602
4603  022302  004737  034140          1$:     JSR     PC,SUBCLR
4604  022306  104024                          ERROR   24              ;CERR AFTER SCLR
4605
4606  022310  010265  000016                  MOV     R2,RKASOF(R5)   ;SET OFFSET
4607
4608  022314  012737  022410  001176          MOV     #64$,$ESCAPE
4609  022322  012765  000015  000000          MOV     #OFFSET,RKCS1(R5)  ;OFFSET CMD
4610  022330  013737  001414  007376          MOV     T100,TEMP1      ;SETUP TIMEOUT
4611  022336  004737  032230                  JSR     PC,FRDY
4612  022342  104033                          ERROR   33              ;NO RDY AFTER OFFSET CMD
4613
4614  022344  012737  032140  007430          MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4615  022352  005037  007432                  CLR     E.B0
4616  022356  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4617  022364  012737  000001  007436          MOV     #1,E.B1
4618
4619  022372  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4620  022376  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4621  022400  104035                          ERROR   35              ;MSG A0 ERROR DURING OFFSET CMD
4622  022402  104061                          ERROR   61              ;MSH B0 ERROR
4623  022404  104036                          ERROR   36              ;MSG A1 ERROR
4624  022406  104062                          ERROR   62              ;MSG B1 ERROR
4625
4626  022410  005037  001176          64$:    CLR     $ESCAPE
4627  022414  013737  001416  007376          MOV     T5000,TEMP1     ;SETUP TIMEOUT
4628  022422  004737  032640                  JSR     PC,FATT2        ;FIND ATTN
4629  022426  104034                          ERROR   34              ;NO ATTN AFTER OFFSET CMD
4630
```

```
4631
4632  022430  012737  052340  007430          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4633  022436  005037  007432                  CLR     E.B0                    ;EXPECTED MSG B0
4634  022442  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4635  022450  012737  000001  007436          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
4636  022456  005037  007440                  CLR     E.A2                    ;EXPECTED MSG A2
4637  022462  012737  000002  007442          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
4638  022470  012737  000003  007446          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
4639
4640  022476  004737  032752                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
4641  022502  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
4642  022504  104260                          ERROR   260                     ;MSG A0 ERROR AFTER OFFSET CMD
4643  022506  104261                          ERROR   261                     ;MSH B0 ERROR
4644  022510  104037                          ERROR   37                      ;MSG A1 ERROR
4645  022512  104040                          ERROR   40                      ;MSG B1 ERROR
4646
4647  022514  005737  001360                  TST     CYLADD
4648  022520  001401                          BEQ     17$
4649  022522  104042                          ERROR   42                      ;CYL ADDR IN B2 WAS NOT 0
4650                                                                          ;AFTER OFFSET CMD FROM CYL 0
4651  022524                          17$:
4652  022524  010265  000016                  MOV     R2,RKASOF(R5)           ;REFRESH RKASOF
4653
4654  022530  032702  000200                  BIT     #BIT7,R2
4655  022534  001005                          BNE     65$                     ;BR IF NEG OFFSET
4656
4657  022536  020237  001356                  CMP     R2,CYLDIF               ;CHECK POS OFFSET
4658  022542  001406                          BEQ     66$
4659  022544  104114                          ERROR   114                     ;OFFSET IN A2 NOT = RKASOF
4660  022546  000404                          BR      66$                     ;AFTER OFFSET CMD
4661
4662  022550  020137  001356          65$:    CMP     R1,CYLDIF               ;CHECK NEG OFFSET
4663  022554  001401                          BEQ     66$
4664  022556  104114                          ERROR   114                     ;OFFSET IN A2 NOT = RKASOF
4665                                                                          ;AFTER OFFSET CMD
4666  022560                          66$:
4667
4668  022560  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4669  022566  013765  001222  000010          MOV     $UNIT,RKCS2(R5)         ;DRIVE#
4670  022574  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)                ;DRIVE CLEAR CMD
4671  022602  013737  001406  007376          MOV     T10,TEMP1               ;SETUP TIMEOUT
4672  022610  004737  032230                  JSR     PC,FRDY                 ;FIND RDY
4673  022614  104151                          ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
4674  022616  004737  032512                  JSR     PC,TSTATN               ;TEST FOR ATTN
4675  022622  000401                          BR      67$
4676  022624  104154                          ERROR   154                     ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4677  022626                          67$:
4678
4679  022626  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4680  022634  004737  033566                  JSR     PC,GSTAT
4681  022640  032737  002000  007366          BIT     #D.OFF,HMR2
4682  022646  001001                          BNE     4$
4683  022650  104043                          ERROR   43                      ;OFFSET BIT IN RKMR2 CLEARED
4684                                                                          ;AFTER DRIVE CLEAR CMD & SELECT DRV CMD
4685  022652  012737  012340  007430  4$:     MOV     #<D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
4686  022660  005037  007432                  CLR     E.B0
```

```
 4687  022664  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 4688  022672  012737  000001  007436          MOV     #1,E.B1
 4689
 4690  022700  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4691  022704  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4692  022706  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 4693  022710  104265                          ERROR   265             ;MSH B0 ERROR
 4694  022712  104274                          ERROR   274             ;MSG A1 ERROR
 4695  022714  104266                          ERROR   266             ;MSG B1 ERROR
 4696  022716  010265  000016                  MOV     R2,RKASOF(R5)   ;REFRESH RKASOF
 4697
 4698  022722  032702  000200                  BIT     #BIT7,R2
 4699  022726  001005                          BNE     68$             ;BR IF NEG OFFSET
 4700
 4701  022730  020237  001356                  CMP     R2,CYLDIF       ;CHECK POS OFFSET
 4702  022734  001406                          BEQ     69$
 4703  022736  104115                          ERROR   115             ;OFFSET IN A2 NOT = RKASOF
 4704  022740  000404                          BR      69$             ;AFTER DRIVE CLEAR CMD
 4705
 4706  022742  020137  001356          68$:    CMP     R1,CYLDIF       ;CHECK NEG OFFSET
 4707  022746  001401                          BEQ     69$
 4708  022750  104115                          ERROR   115             ;OFFSET IN A2 NOT = RKASOF
 4709                                                                  ;AFTER DRIVE CLEAR CMD
 4710  022752                          69$:
 4711
 4712  022752  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
 4713  022760  013737  001410  007376          MOV     T50,TEMP1       ;SETUP TIMEOUT
 4714  022766  004737  032230                  JSR     PC,FRDY         ;FIND RDY
 4715  022772  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 4716
 4717  022774  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
 4718  023002  004737  032640                  JSR     PC,FATT2        ;FIND ATTN
 4719  023006  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 4720
 4721  023010  032737  100000  007340          BIT     #CERR,HCS1
 4722  023016  001401                          BEQ     70$
 4723  023020  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 4724
 4725  023022                          70$:
 4726
 4727
 4728  023022  032737  002000  007366          BIT     #D.OFF,HMR2
 4729  023030  001001                          BNE     7$
 4730  023032  104045                          ERROR   45              ;OFFSET BIT CLEARED IN RKMR2 AFTER SEEK TO SELF.
 4731
 4732  023034                          7$:
 4733
 4734  023034  012737  052340  007430          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4735  023042  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
 4736  023046  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4737  023054  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4738  023062  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
 4739  023066  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4740  023074  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4741
 4742  023102  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
```

```
4743  023106  000003                    .WORD   T.A2!T.B2!0          ;& MSGS SPECIFIED HERE
4744  023110  104140                    ERROR   140                  ;MSG A0 ERROR AFTER SEEK TO SELF
4745  023112  104141                    ERROR   141                  ;MSH B0 ERROR
4746  023114  104142                    ERROR   142                  ;MSG A1 ERROR
4747  023116  104143                    ERROR   143                  ;MSG B1 ERROR
4748  023120  010265  000016            MOV     R2,RKASOF(R5)        ;REFRESH RKASOF
4749
4750  023124  032702  000200            BIT     #BIT7,R2
4751  023130  001005                    BNE     71$                  ;BR IF NEG OFFSET
4752
4753  023132  020237  001356            CMP     R2,CYLDIF            ;CHECK POS OFFSET
4754  023136  001406                    BEQ     72$
4755  023140  104123                    ERROR   123                  ;OFFSET IN A2 NOT = RKASOF
4756  023142  000404                    BR      72$                  ;AFTER SEEK TO SELF
4757
4758  023144  020137  001356    71$:    CMP     R1,CYLDIF            ;CHECK NEG OFFSET
4759  023150  001401                    BEQ     72$
4760  023152  104123                    ERROR   123                  ;OFFSET IN A2 NOT = RKASOF
4761                                                                 ;AFTER SEEK TO SELF
4762  023154                    72$:
4763
4764  023154  004737  034140            JSR     PC,SUBCLR
4765  023160  104024                    ERROR   24                   ;CERR AFTER SCLR
4766
4767  023162  012737  000012  001346    MOV     #10.,TOCYL
4768  023170  012765  000012  000020    MOV     #10.,RKDC(R5)        ;SETUP CYL 10
4769                                                                 ;DO ACTUAL IMPLIED SEEK TO CYL 10 TO VERIFY
4770                                                                 ;OFFSET BIT IN RKMR2 CLEARED
4771
4772
4773  023176  012700  001712            MOV     #RHTAB,R0
4774  023202  012765  000025  000000    MOV     #<RDHEAD>,RKCS1(R5)  ;READ HEADER CMD
4775  023210  013737  001420  007376    MOV     T50000,TEMP1         ;SETUP  TIMEOUT
4776  023216  004737  032230            JSR     PC,FRDY              ;FIND RDY
4777  023222  104171                    ERROR   171                  ;NO RDY AFTER READ HEADER CMD
4778  023224  032737  100000  007340    BIT     #CERR,HCS1
4779  023232  001405                    BEQ     74$
4780  023234  104174                    ERROR   174                  ;CERR AFTER READ HEADER CMD
4781  023236  104401  045277            TYPE    ,MSG26               ;ABORTING BAL OF TESTS
4782  023242  000137  031514            JMP     $EOP
4783
4784  023246  016520  000024    74$:    MOV     RKDB(R5),(R0)+       ;1'ST WORD FROM SILO TO RHTAB
4785  023252  016520  000024            MOV     RKDB(R5),(R0)+       ;2'ND WORD
4786  023256  016520  000024            MOV     RKDB(R5),(R0)+       ;3'RD WORD
4787
4788
4789  023262  032765  100000  000010    BIT     #DLT,RKCS2(R5)
4790  023270  001407                    BEQ     75$
4791  023272  004737  033566            JSR     PC,GSTAT
4792  023276  104173                    ERROR   173                  ;DLT AFTER READ HEADER CMD
4793  023300  104401  045277            TYPE    ,MSG26               ;ABORTING BAL OF TESTS
4794  023304  000137  031514            JMP     $EOP
4795  023310                    75$:
4796
4797  023310  012737  010340  007430    MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4798  023316  005037  007432            CLR     E.B0                 ;EXPECTED MSG B0
```

```
 4799  023322  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4800  023330  012737  000001  007436          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
 4801  023336  005037  007440                  CLR     E.A2             ;EXPECTED MSG A2
 4802  023342  012737  000002  007442          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 4803  023350  012737  000003  007446          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 4804
 4805  023356  004737  032752                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 4806  023362  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
 4807  023364  104301                          ERROR   301              ;MSG A0 ERROR AFTER READ HEADER CMD
 4808  023366  104271                          ERROR   271              ;MSH B0 ERROR
 4809  023370  104302                          ERROR   302              ;MSG A1 ERROR
 4810  023372  104272                          ERROR   272              ;MSG B1 ERROR
 4811
 4812  023374  023737  001712  001346          CMP     RHTAB,TOCYL      ;CHECK WORD 0 ONLY, CYL#
 4813  023402  001401                          BEQ     73$              ;BR IF SAME
 4814  023404  104051                          ERROR   51               ;WRONG CYL# ON HEADER
 4815  023406                          73$:
 4816
 4817
 4818  023406  032737  002000  007366          BIT     #D.OFF,HMR2
 4819  023414  001401                          BEQ     9$
 4820  023416  104101                          ERROR   101              ;OFFSET NOT CLEARED AFTER READ HEADER WITH MOVEMENT
 4821
 4822  023420  023727  001360  000012  9$:     CMP     CYLADD,#10.
 4823  023426  001401                          BEQ     10$
 4824  023430  104122                          ERROR   122              ;DID NOT GO TO CYL 10
 4825
 4826  023432  005737  001356          10$:    TST     CYLDIF
 4827  023436  001401                          BEQ     16$
 4828  023440  104101                          ERROR   101              ;OFFSET NOT CLEARED IN RKMR2
 4829
 4830  023442  004737  034140          16$:    JSR     PC,SUBCLR
 4831  023446  104024                          ERROR   24               ;CERR AFTER SCLR
 4832
 4833  023450  012765  000017  000000          MOV     #SEEK,RKCS1(R5)  ;SEEK CMD
 4834  023456  013737  001410  007376          MOV     T50,TEMP1        ;SETUP TIMEOUT
 4835  023464  004737  032230                  JSR     PC,FRDY          ;FIND RDY
 4836  023470  104131                          ERROR   131              ;NO RDY AFTER SEEK CMD
 4837
 4838  023472  013737  001420  007376          MOV     T50000,TEMP1     ;SETUP TIMEOUT
 4839  023500  004737  032640                  JSR     PC,FATT2         ;FIND ATTN
 4840  023504  104132                          ERROR   132              ;NO ATTN AFTER SEEK CMD
 4841
 4842  023506  032737  100000  007340          BIT     #CERR,HCS1
 4843  023514  001401                          BEQ     76$
 4844  023516  104210                          ERROR   210              ;CERR AFTER SEEK CMD
 4845
 4846  023520                          76$:
 4847
 4848
 4849  023520  032702  000200                  BIT     #BIT7,R2         ;SEE IF DOING NEG OFFSETS
 4850  023524  001014                          BNE     18$              ;BR IF YES
 4851
 4852  023526  005202                          INC     R2
 4853  023530  020227  000061                  CMP     R2,#61           ;SEE IF JUST DID MAX POS OFFSET
 4854  023534  001402                          BEQ     20$              ;BR IF YES
```

```
                                        JMP     1$              ;ELSE DO NEXT POS OFFSET
                                
                        20$:    MOV     #201,R2         ;SETUP NEG OFFSET FOR RKASOF
                                MOV     #101,R1         ;SETUP NEG OFFSET OFOR MSG A
                                JMP     1$              ;DO NEG OFFSET
                                
                        18$:    INC     R1
                                INC     R2
                                CMP     R2,#261         ;SEE IF ALL NEG OFFSETS DONE
                                BEQ     TST14           ;;GO TO NEXT TST
                                JMP     1$              ;DO ANOTHER
                                
;;*****************************************************************
;*TEST 14        TEST READ DATA AT ALL HEAD OFFSET POSITIONS
;*
;*      THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY
;*      WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN
;*      PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET
;*      POSITIONS UNTIL A FAILURE OCCURES.  THE FAILING OFFSET POSITIONS
;*      ARE TYPED OUT.
;*      OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10.
;*
;*      ALL HEADS ARE TESTED AT CYLINDER 0
;*      IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT
;*
;*              A. HEADS DID NOT MOVE AT ALL
;*      OR      B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP
;*                 ARE EXCEPTIONALLY GOOD.
;*
;*      NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR,
;*      BUT AN INDICATION OF SURFACE, HEAD & R/W ELECTRONICS QUALITY ONLY.
;*
;;*****************************************************************
000004  TST14:  SCOPE
001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
001100          MOV     #STACK,SP       ;RESTORE STK PTR
                
034140          JSR     PC,SUBCLR
104024          ERROR   24              ;CERR AFTER SCLR
                
001402          CLR     SECTOR
000006  11$:    MOV     SECTOR,RKDA(R5)
000004          MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
000010          BIS     #BAI,RKCS2(R5)  ;BUS ADDR INCR INHIB
000002          MOV     #-256.,RKWC(R5) ;SECTOR 0 ONLY
                                        ;WILL DO IMPLIED SEEK TO CYL 0
                                        ;WAS ON CYL 1 FROM LAST TEST
                
000000          MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
032230          JSR     PC,FRDY         ;FIND RDY
104011          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
033566          JSR     PC,GSTAT        ;GET FRESH STATUS
007340          BIT     #CERR,HCS1
001465          BEQ     67$             ;BR IF NO ERRORS
```

```
 4911  023710  032737  000200  007354          BIT     #BSE,HER          ;SEE IF BAD SECTOR FLAG
 4912  023716  001421                          BEQ     65$               ;BR IF NO
 4913  023720  004737  035610                  JSR     PC,TRUERR         ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
 4914  023724  000455                          BR      66$               ;RETURN HERE IF NO
 4915
 4916  023726  005237  001402                  INC     SECTOR            ;RETURN HERE IF YES
 4917  023732  023727  001402  000012          CMP     SECTOR,#10.       ;ARE 10 CONSEC. SECTORS BAD
 4918  023740  001003                          BNE     64$               ;BR IF NO
 4919  023742  104046                          ERROR   46                ;ABORTING TEST DETECTED 10 BAD SECTORS
 4920  023744  000137  025406                  JMP     10$               ;BYPASS TEST
 4921  023750  012765  100000  000000  64$:    MOV     #CCLR,RKCS1(R5)   ;TRY ANOTHER SECTOR
 4922  023756  000137  023622                  JMP     11$
 4923  023762  104012                  65$:    ERROR   12                ;CERR WITH WRITE DATA CMD
 4924
 4925  023764  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4926  023772  005037  007432                  CLR     E.B0              ;EXPECTED MSG B0
 4927  023776  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4928  024004  012737  000001  007436          MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
 4929  024012  005037  007440                  CLR     E.A2              ;EXPECTED MSG A2
 4930  024016  012737  000002  007442          MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
 4931  024024  012737  000003  007446          MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
 4932
 4933  024032  004737  032752                  JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
 4934  024036  000003                          .WORD   T.A2!T.B2!0               ;& MSGS SPECIFIED HERE
 4935  024040  104052                          ERROR   52                ;MSG A0 ERROR AFTER WRITE DATA CMD
 4936  024042  104023                          ERROR   23                ;MSH B0 ERROR
 4937  024044  104053                          ERROR   53                ;MSG A1 ERROR
 4938  024046  104025                          ERROR   25                ;MSG B1 ERROR
 4939  024050  104401  045277                  TYPE    ,MSG26            ;ABORTING BAL OF TESTS
 4940  024054  000137  031514                  JMP     $EOP
 4941  024060  104063                  66$:    ERROR   63                ;BAD SECTOR NOT LISTED IN TABLE
 4942  024062                           67$:
 4943
 4944  024062  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4945  024070  005037  007432                  CLR     E.B0              ;EXPECTED MSG B0
 4946  024074  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4947  024102  012737  000001  007436          MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
 4948  024110  005037  007440                  CLR     E.A2              ;EXPECTED MSG A2
 4949  024114  012737  000002  007442          MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
 4950  024122  012737  000003  007446          MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
 4951
 4952  024130  004737  032752                  JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
 4953  024134  000003                          .WORD   T.A2!T.B2!0               ;& MSGS SPECIFIED HERE
 4954  024136  104052                          ERROR   52                ;MSG A0 ERROR AFTER WRITE DATA CMD
 4955  024140  104023                          ERROR   23                ;MSH B0 ERROR
 4956  024142  104053                          ERROR   53                ;MSG A1 ERROR
 4957  024144  104025                          ERROR   25                ;MSG B1 ERROR
 4958  024146  012765  001464  000004          MOV     #DATA1,RKBA(R5)
 4959  024154  052765  000020  000010          BIS     #BAI,RKCS2(R5)
 4960  024162  012765  177400  000002          MOV     #-256.,RKWC(R5)
 4961  024170  013765  001402  000006          MOV     SECTOR,RKDA(R5)
 4962
 4963  024176  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
 4964  024204  013737  001420  007376          MOV     TS0000,TEMP1      ;SETUP TIMEOUT
 4965  024212  004737  032230                  JSR     PC,FRDY           ;FIND RDY
 4966  024216  104015                          ERROR   15                ;NO RDY AFTER WRITE CHECK CMD
```

```
                                                JSR     PC,GSTAT        ;GET FRESH STATUS
                                                BIT     #CERR,HCS1
                                                BEQ     69$
                                                BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
                                                BEQ     68$
                                                MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
                                                MOV     DATA1,WD2               ;EXPECTED WORD FOR TYPEOUT
                                                ERROR   16              ;WCE AFTER WRITE CMD
                                                BR      69$

                                        68$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD

                                                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   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
                                                ERROR   31              ;MSH B0 ERROR
                                                ERROR   60              ;MSG A1 ERROR
                                                ERROR   32              ;MSG B1 ERROR
                                                TYPE    ,MSG26          ;ABORTING BAL OF TESTS
                                                JMP     $EOP

                                        69$:

                                                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   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
                                                ERROR   31              ;MSH B0 ERROR
                                                ERROR   60              ;MSG A1 ERROR
                                                ERROR   32              ;MSG B1 ERROR

                                                TYPE    ,MSG8           ;READ WITH OFFSET TEST
                                                CLR     R1              ;HEAD #

                                        9$:     MOV     #1,R0           ;INIT OFFSET COUNTER
                                                TYPE    ,MSG9           ;HEAD #
                                                MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
                                                                        ;;TYPE HEAD #
                                                TYPOS                   ;;GO TYPE--OCTAL ASCII
                                                .BYTE   1               ;;TYPE 1 DIGIT(S)
                                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
5023  024472  104401  001205              TYPE    ,$CRLF
5024
5025  024476  005037  001442        1$:   CLR     OFFERR          ;WRITE CHECK ERROR FLAG
5026
5027  024502  004737  034140              JSR     PC,SUBCLR
5028  024506  104024                      ERROR   24              ;CERR AFTER SCLR
5029
5030  024510  010065  000016              MOV     R0,RKASOF(R5)   ;OFFSET VALUE
5031  024514  000301                      SWAB    R1
5032  024516  010165  000006              MOV     R1,RKDA(R5)     ;HEAD NO.
5033  024522  000301                      SWAB    R1
5034
5035  024524  012737  024620  001176      MOV     #70$,$ESCAPE
5036  024532  012765  000015  000000      MOV     #OFFSET,RKCS1(R5)  ;OFFSET CMD
5037  024540  013737  001414  007376      MOV     T100,TEMP1      ;SETUP TIMEOUT
5038  024546  004737  032230              JSR     PC,FRDY
5039  024552  104033                      ERROR   33              ;NO RDY AFTER OFFSET CMD
5040
5041  024554  012737  032140  007430      MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5042  024562  005037  007432              CLR     E.B0
5043  024566  012737  001720  007434      MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5044  024574  012737  000001  007436      MOV     #1,E.B1
5045
5046  024602  004737  032752              JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5047  024606  000000                      .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5048  024610  104035                      ERROR   35              ;MSG A0 ERROR DURING OFFSET CMD
5049  024612  104061                      ERROR   61              ;MSH B0 ERROR
5050  024614  104036                      ERROR   36              ;MSG A1 ERROR
5051  024616  104062                      ERROR   62              ;MSG B1 ERROR
5052
5053  024620  005037  001176        70$:  CLR     $ESCAPE
5054  024624  013737  001416  007376      MOV     T5000,TEMP1     ;SETUP TIMEOUT
5055  024632  004737  032640              JSR     PC,FATT2        ;FIND ATTN
5056  024636  104034                      ERROR   34              ;NO ATTN AFTER OFFSET CMD
5057
5058
5059  024640  012737  052340  007430      MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5060  024646  005037  007432              CLR     E.B0            ;EXPECTED MSG B0
5061  024652  012737  001720  007434      MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5062  024660  012737  000001  007436      MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5063  024666  005037  007440              CLR     E.A2            ;EXPECTED MSG A2
5064  024672  012737  000002  007442      MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5065  024700  012737  000003  007446      MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5066
5067  024706  004737  032752              JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5068  024712  000003                      .WORD   T.A2!T.B2!0             ;& MSG$ SPECIFIED HERE
5069  024714  104260                      ERROR   260             ;MSG A0 ERROR AFTER OFFSET CMD
5070  024716  104261                      ERROR   261             ;MSH B0 ERROR
5071  024720  104037                      ERROR   37              ;MSG A1 ERROR
5072  024722  104040                      ERROR   40              ;MSG B1 ERROR
5073
5074
5075  024724  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
5076  024732  013765  001222  000010      MOV     $UNIT,RKCS2(R5) ;DRIVE#
5077  024740  012765  000005  000000      MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5078  024746  013737  001406  007376      MOV     T10,TEMP1       ;SETUP TIMEOUT
```

```
5079  024754  004737  032230          JSR     PC,FRDY         ;FIND RDY
5080  024760  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5081  024762  004737  032512          JSR     PC,TSTATN       ;TEST FOR ATTN
5082  024766  000401                  BR      71$
5083  024770  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5084  024772                  71$:
5085
5086  024772  012765  001464  000004  MOV     #DATA1,RKBA(R5)
5087  025000  052765  000020  000010  BIS     #BAI,RKCS2(R5)
5088  025006  012765  177400  000002  MOV     #-256.,RKWC(R5)
5089  025014  013765  001402  000006  MOV     SECTOR,RKDA(R5)
5090  025022  012765  000031  000000  MOV     #WRTCHK,RKCS1(R5)  ;WRITE CHECK CMD
5091  025030  012737  141520  007376  MOV     #50000.,TEMP1   ;SETUP TIMEOUT
5092  025036  004737  032230          JSR     PC,FRDY         ;FIND RDY
5093  025042  104015                  ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
5094  025044  004737  033566          JSR     PC,GSTAT        ;GET FRESH STATUS
5095  025050  032737  040000  007342  BIT     #WCE,HCS2
5096  025056  001421                  BEQ     2$
5097
5098  025060  016537  000024  001436  MOV     RKDB(R5),WD1    ;GET MISCOMPARED WORD
5099  025066  005237  001442          INC     OFFERR          ;BAD WRITE CHK ERROR=SET ERR FLG.
5100
5101  025072  005737  001442          TST     OFFERR
5102  025076  001411                  BEQ     2$
5103  025100  104401  045543          TYPE    ,MSG39          ;WRITE CHECK FAILURE AT OFFSET
5104  025104  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
5105                                                          ;;TYPE OFFSET VALUE
5106  025106  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
5107  025110     006                  .BYTE   6               ;;TYPE 6 DIGITS
5108  025111     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
5109  025112  104401  001205          TYPE    ,$CRLF
5110  025116  104401  001205          TYPE    ,$CRLF
5111
5112  025122  032700  000200  2$:     BIT     #BIT7,R0        ;SEE IF OFFSET IS + OR -
5113  025126  001023                  BNE     5$              ;BR IF - OFFSET
5114
5115  025130  020027  000060          CMP     R0,#60
5116  025134  001412                  BEQ     4$
5117  025136  005737  001442          TST     OFFERR
5118  025142  001404                  BEQ     3$
5119  025144  012700  000200  8$:     MOV     #200,R0         ;SETUP FOR NEG OFFSET
5120  025150  000137  024476          JMP     1$
5121
5122  025154  005200          3$:     INC     R0
5123  025156  000137  024476          JMP     1$
5124
5125  025162  005737  001442  4$:     TST     OFFERR
5126  025166  001366                  BNE     8$              ;DO NEG OFFSETS
5127  025170  104401  045405          TYPE    ,MSG37          ;NO WRITE CHECK ERROR AT MAX POS OFFSET
5128                                                          ;NOTE! EITHER HEADS DID NOT MOVE
5129                                                          ;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
5130  025174  000763                  BR      8$              ;DO NEG OFFSETS
5131
5132  025176  020027  000260  5$:     CMP     R0,#260
5133  025202  001404                  BEQ     6$
5134  025204  005737  001442          TST     OFFERR
```

```
5135  025210  001100                        BNE     TST15           ;;GO TO NEXT TST
5136  025212  000760                        BR      3$
5137
5138  025214  005737  001442          6$:   TST     OFFERR
5139  025220  001002                        BNE     7$
5140  025222  104401  045463                TYPE    ,MSG38          ;NO WRITE CHECK ERROR AT MAX NEG OFFSET
5141                                                                  ;NOTE! EITHER HEADS DID NOT MOVE
5142                                                                  ;OR READ/WRITE AMP IS EXECEPTIONALLY GOOD.
5143  025226                          7$:
5144
5145  025226  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
5146  025234  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
5147  025242  012765  000013  000000        MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
5148                                                                  ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5149                                                                  ;IN RKMR2 & RKMR3 RESP.
5150  025250  013737  001406  007376        MOV     T10,TEMP1       ;SETUP TIMEOUT
5151  025256  004737  032230                JSR     PC,FRDY         ;FIND RDY
5152  025262  104124                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD
5153
5154  025264  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5155  025272  004737  033566                JSR     PC,GSTAT
5156  025276  032737  020000  007366        BIT     #D.RTZ,HMR2
5157  025304  001001                        BNE     72$
5158  025306  104244                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
5159  025310  013737  001406  007400  72$:  MOV     T10,TEMP2       ;SETUP TIMEOUT
5160  025316  004737  032544                JSR     PC,FATT1        ;FIND ATTN
5161  025322  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD
5162
5163  025324  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
5164  025332  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
5165  025340  012765  000005  000000        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5166  025346  013737  001406  007376        MOV     T10,TEMP1       ;SETUP TIMEOUT
5167  025354  004737  032230                JSR     PC,FRDY         ;FIND RDY
5168  025360  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5169  025362  004737  032512                JSR     PC,TSTATN       ;TEST FOR ATTN
5170  025366  000401                        BR      73$
5171  025370  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5172  025372                          73$:
5173
5174
5175  025372  005201                        INC     R1              ;HEAD CTR
5176  025374  020127  000003                CMP     R1,#3           ;SEE IF ALL HEADS DONE
5177  025400  001402                        BEQ     10$             ;BR IF YES
5178  025402  000137  024454                JMP     9$              ;ELSE REPEAT ALL FOR NEXT HEAD
5179  025406  104401  044723          10$:  TYPE    ,MSG12          ;OFFSET FAILURES ARE NOT ERRORS
5180                                        ;;*****************************************************************
5181                                        ;*TEST 15       WRITE WITH HEADS OFFSET
5182                                        ;*
5183                                        ;*      THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
5184                                        ;*      WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
5185                                        ;*      & THE DRIVE WILL WRITE
5186                                        ;*      SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
5187                                        ;*      THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY
5188                                        ;*
5189                                        ;;*****************************************************************
5190  025412  000004                    TST15:  SCOPE
```

```
5191  025414  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5192  025422  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5193
5194  025426  012700  000260                  MOV     #260,R0         ;MAX NEG OFFSET
5195
5196  025432  004737  034140          1$:     JSR     PC,SUBCLR
5197  025436  104024                          ERROR   24              ;CERR AFTER SCLR
5198
5199  025440  010065  000016                  MOV     R0,RKASOF(R5)   ;SET OFFSET
5200
5201  025444  012737  025540  001176          MOV     #64$,$ESCAPE
5202  025452  012765  000015  000000          MOV     #OFFSET,RKCS1(R5)  ;OFFSET CMD
5203  025460  013737  001414  007376          MOV     T100,TEMP1      ;SETUP TIMEOUT
5204  025466  004737  032230                  JSR     PC,FRDY
5205  025472  104033                          ERROR   33              ;NO RDY AFTER OFFSET CMD
5206
5207  025474  012737  032140  007430          MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5208  025502  005037  007432                  CLR     E.B0
5209  025506  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5210  025514  012737  000001  007436          MOV     #1,E.B1
5211
5212  025522  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5213  025526  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5214  025530  104035                          ERROR   35              ;MSG A0 ERROR DURING OFFSET CMD
5215  025532  104061                          ERROR   61              ;MSH B0 ERROR
5216  025534  104036                          ERROR   36              ;MSG A1 ERROR
5217  025536  104062                          ERROR   62              ;MSG B1 ERROR
5218
5219  025540  005037  001176          64$:    CLR     $ESCAPE
5220  025544  013737  001416  007376          MOV     T5000,TEMP1     ;SETUP TIMEOUT
5221  025552  004737  032640                  JSR     PC,FATT2        ;FIND ATTN
5222  025556  104034                          ERROR   34              ;NO ATTN AFTER OFFSET CMD
5223
5224
5225  025560  012737  052340  007430          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5226  025566  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5227  025572  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5228  025600  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5229  025606  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5230  025612  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5231  025620  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5232
5233  025626  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5234  025632  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5235  025634  104260                          ERROR   260             ;MSG A0 ERROR AFTER OFFSET CMD
5236  025636  104261                          ERROR   261             ;MSH B0 ERROR
5237  025640  104037                          ERROR   37              ;MSG A1 ERROR
5238  025642  104040                          ERROR   40              ;MSG B1 ERROR
5239
5240
5241  025644  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5242  025652  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5243  025660  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5244  025666  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
5245  025674  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5246  025700  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
```

```
5247  025702  004737  032512          JSR     PC,TSTATN       ;TEST FOR ATTN
5248  025706  000401                  BR      65$
5249  025710  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5250  025712                  65$:
5251
5252
5253  025712  005037  001402          CLR     SECTOR
5254  025716  013765  001402  000006  4$:     MOV     SECTOR,RKDA(R5)
5255  025724  012765  001460  000004          MOV     #DATAO,RKBA(R5) ;WRITE ALL 0'S
5256  025732  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUS ADDR INCR INH
5257  025740  012765  177400  000002          MOV     #-256.,RKWC(R5) ;FULL SECTOR
5258
5259  025746  012765  000023  000000          MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
5260  025754  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5261  025762  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5262  025766  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
5263  025770  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
5264  025774  032737  100000  007340          BIT     #CERR,HCS1
5265  026002  001465                          BEQ     69$             ;BR IF NO ERRORS
5266
5267  026004  032737  000200  007354          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
5268  026012  001421                          BEQ     67$             ;BR IF NO
5269  026014  004737  035610                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
5270  026020  000455                          BR      68$             ;RETURN HERE IF NO
5271
5272  026022  005237  001402                  INC     SECTOR          ;RETURN HERE IF YES
5273  026026  023727  001402  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
5274  026034  001003                          BNE     66$             ;BR IF NO
5275  026036  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
5276  026040  000137  026760                  JMP     3$              ;BYPASS TEST
5277  026044  012765  100000  000000  66$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
5278  026052  000137  025716                  JMP     4$
5279  026056  104012                  67$:    ERROR   12              ;CERR WITH WRITE DATA CMD
5280
5281  026060  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5282  026066  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5283  026072  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5284  026100  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5285  026106  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5286  026112  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5287  026120  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5288
5289  026126  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5290  026132  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5291  026134  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5292  026136  104023                          ERROR   23              ;MSH B0 ERROR
5293  026140  104053                          ERROR   53              ;MSG A1 ERROR
5294  026142  104025                          ERROR   25              ;MSG B1 ERROR
5295  026144  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5296  026150  000137  031514                  JMP     $EOP
5297  026154  104063                  68$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
5298  026156                  69$:
5299
5300  026156  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5301  026164  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5302  026170  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
```

```
5303  026176  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5304  026204  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5305  026210  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5306  026216  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5307
5308  026224  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5309  026230  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5310  026232  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5311  026234  104023                          ERROR   23              ;MSH B0 ERROR
5312  026236  104053                          ERROR   53              ;MSG A1 ERROR
5313  026240  104025                          ERROR   25              ;MSG B1 ERROR
5314
5315  026242  005737  001356                  TST     CYLDIF          ;SEE IF MSG A2=0
5316  026246  001401                          BEQ     70$             ;BR IF YES
5317  026250  104104                          ERROR   104             ;MSG A2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5318  026252  005737  001360          70$:    TST     CYLADD          ;SEE IF MSG B2=0
5319  026256  001401                          BEQ     71$             ;BR IF YES
5320  026260  104105                          ERROR   105             ;MSG B2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5321  026262                          71$:
5322
5323  026262  104415                          SCOP1
5324  026264  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5325
5326  026270  004737  034140                  JSR     PC,SUBCLR
5327  026274  104024                          ERROR   24              ;CERR AFTER SCLR
5328
5329
5330  026276  012765  001460  000004          MOV     #DATAO,RKBA(R5)
5331  026304  052765  000020  000010          BIS     #BAI,RKCS2(R5)
5332  026312  012765  177400  000002          MOV     #-256.,RKWC(R5)
5333  026320  013765  001402  000006          MOV     SECTOR,RKDA(R5)
5334
5335  026326  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
5336  026334  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5337  026342  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5338  026346  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
5339  026350  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
5340  026354  032737  100000  007340          BIT     #CERR,HCS1
5341  026362  001453                          BEQ     73$
5342  026364  032737  040000  007342          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
5343  026372  001410                          BEQ     72$
5344  026374  016537  000024  001436          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
5345  026402  013737  001460  001440          MOV     DATAO,WD2               ;EXPECTED WORD FOR TYPEOUT
5346  026410  104016                          ERROR   16              ;WCE AFTER WRITE CMD
5347  026412  000437                          BR      73$
5348
5349  026414  104022                  72$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
5350
5351  026416  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5352  026424  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5353  026430  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5354  026436  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5355  026444  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5356  026450  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5357  026456  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5358
```

```
5359  026464  004737  032752          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5360  026470  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5361  026472  104057                  ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5362  026474  104031                  ERROR   31              ;MSH B0 ERROR
5363  026476  104060                  ERROR   60              ;MSG A1 ERROR
5364  026500  104032                  ERROR   32              ;MSG B1 ERROR
5365  026502  104401  045277          TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5366  026506  000137  031514          JMP     $EOP
5367
5368  026512                  73$:
5369
5370  026512  012737  010340  007430  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5371  026520  005037  007432          CLR     E.B0            ;EXPECTED MSG B0
5372  026524  012737  001720  007434  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5373  026532  012737  000001  007436  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5374  026540  005037  007440          CLR     E.A2            ;EXPECTED MSG A2
5375  026544  012737  000002  007442  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5376  026552  012737  000003  007446  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5377
5378  026560  004737  032752          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5379  026564  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5380  026566  104057                  ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5381  026570  104031                  ERROR   31              ;MSH B0 ERROR
5382  026572  104060                  ERROR   60              ;MSG A1 ERROR
5383  026574  104032                  ERROR   32              ;MSG B1 ERROR
5384
5385  026576  020027  000260          CMP     R0,#260
5386  026602  001004                  BNE     2$              ;BR IF JUST DID POS OFFSET
5387  026604  012700  000060          MOV     #60,R0          ;ELSE SETUP FOR POS OFFSET
5388  026610  000137  025432          JMP     1$
5389
5390  026614                  2$:
5391
5392  026614  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5393  026622  013765  001222  000010  MOV     $UNIT,RKCS2(R5)
5394  026630  012765  000013  000000  MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
5395                                          ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5396                                          ;IN RKMR2 & RKMR3 RESP.
5397  026636  013737  001406  007376  MOV     T10,TEMP1       ;SETUP TIMEOUT
5398  026644  004737  032230          JSR     PC,FRDY         ;FIND RDY
5399  026650  104124                  ERROR   124             ;RDY NOT SET AFTER RECAL CMD
5400
5401  026652  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5402  026660  004737  033566          JSR     PC,GSTAT
5403  026664  032737  020000  007366  BIT     #D.RTZ,HMR2
5404  026672  001001                  BNE     74$
5405  026674  104244                  ERROR   244             ;RTZ NOT SET DURING RECAL CMD
5406  026676  013737  001406  007400  74$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
5407  026704  004737  032544          JSR     PC,FATT1        ;FIND ATTN
5408  026710  104055                  ERROR   55              ;NO ATTN AFTER RECAL CMD
5409
5410  026712  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5411  026720  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5412  026726  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5413  026734  013737  001406  007376  MOV     T10,TEMP1       ;SETUP TIMEOUT
5414  026742  004737  032230          JSR     PC,FRDY         ;FIND RDY
```

```
                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                        JSR     PC,TSTATN       ;TEST FOR ATTN
                        BR      75$
                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                75$:

                3$:

                ;;*****************************************************************
                ;*TEST 16       TEST CURRENT CROSS-OVER CYLINDERS
                ;*
                ;*      THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF
                ;*      CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:
                ;*
                ;*      SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y
                ;*      WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.
                ;*
                ;*      A WRITE CHECK IS THEN PREFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.
                ;*      THIS TEST IS PERFORMED FOR ALL 3 HEADS.
                ;*
                ;*      CYLINDER X:  63  127  191  255  319  383
                ;*      CYLINDER Y:  64  128  192  256  320  384
                ;*
                ;;*****************************************************************
026760 000004   TST16:  SCOPE
026762 012737 000001 001174     MOV     #1,$TIMES       ;;DO 1 ITERATION
026770 012706 001100            MOV     #STACK,SP
026774 012700 001422            MOV     #CYL,R0         ;CYL ADDR TABLE
027000 004737 034140    1$:     JSR     PC,SUBCLR
027004 104024                   ERROR   24              ;CERR AFTER SCLR
027006 011065 000020            MOV     (R0),RKDC(R5)   ;CYL #
027012 012765 001466 000004     MOV     #DPAT1,RKBA(R5) ;DATA PATTERN
027020 052765 000020 000010     BIS     #BAI,RKCS2(R5)  ;BUSS ADDR INCREMENT INHIBIT
027026 012765 076000 000002     MOV     #-6*22.*256.,RKWC(R5) ;WORD COUNT TO SPIRAL & FILL 2 CYLINDERS
027034 012765 000023 000000     MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
027042 013737 001420 007376     MOV     T50000,TEMP1    ;SETUP TIMEOUT
027050 004737 032230            JSR     PC,FRDY         ;FIND RDY
027054 104011                   ERROR   11              ;NO RDY AFTER WRITE DATA CMD
027056 004737 033566            JSR     PC,GSTAT        ;GET FRESH STATUS
027062 032737 100000 007340     BIT     #CERR,HCS1
027070 001451                   BEQ     67$             ;BR IF NO ERRORS
027072 032737 000200 007354     BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
027100 001405                   BEQ     65$             ;BR IF NO
027102 004737 035610            JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
027106 000441                   BR      66$             ;RETURN HERE IF NO
027110 000137 027576            JMP     3$              ;RET HERE IF YES
027114 104012           65$:    ERROR   12              ;CERR WITH WRITE DATA CMD
027116 012737 010340 007430     MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
027124 005037 007432            CLR     E.B0            ;EXPECTED MSG B0
```

```
5471  027130  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5472  027136  012737  000001  007436          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5473  027144  005037  007440                  CLR     E.A2                ;EXPECTED MSG A2
5474  027150  012737  000002  007442          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5475  027156  012737  000003  007446          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5476
5477  027164  004737  032752                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5478  027170  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
5479  027172  104052                          ERROR   52                  ;MSG A0 ERROR AFTER WRITE DATA CMD
5480  027174  104023                          ERROR   23                  ;MSH B0 ERROR
5481  027176  104053                          ERROR   53                  ;MSG A1 ERROR
5482  027200  104025                          ERROR   25                  ;MSG B1 ERROR
5483  027202  104401  045277                  TYPE    ,MSG26              ;ABORTING BAL OF TESTS
5484  027206  000137  031514                  JMP     $EOP
5485  027212  104063                  66$:    ERROR   63                  ;BAD SECTOR NOT LISTED IN TABLE
5486  027214                           67$:
5487
5488  027214  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5489  027222  005037  007432                  CLR     E.B0                ;EXPECTED MSG B0
5490  027226  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5491  027234  012737  000001  007436          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5492  027242  005037  007440                  CLR     E.A2                ;EXPECTED MSG A2
5493  027246  012737  000002  007442          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5494  027254  012737  000003  007446          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5495
5496  027262  004737  032752                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5497  027266  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
5498  027270  104052                          ERROR   52                  ;MSG A0 ERROR AFTER WRITE DATA CMD
5499  027272  104023                          ERROR   23                  ;MSH B0 ERROR
5500  027274  104053                          ERROR   53                  ;MSG A1 ERROR
5501  027276  104025                          ERROR   25                  ;MSG B1 ERROR
5502
5503  027300  011065  000020                  MOV     (R0),RKDC(R5)
5504  027304  012765  001466  000004          MOV     #DPAT1,RKBA(R5)
5505  027312  052765  000020  000010          BIS     #BAI,RKCS2(R5)
5506  027320  012765  076000  000002          MOV     #-6*22.*256.,RKWC(R5)
5507
5508  027326  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
5509  027334  013737  001420  007376          MOV     T50000,TEMP1        ;SETUP TIMEOUT
5510  027342  004737  032230                  JSR     PC,FRDY             ;FIND RDY
5511  027346  104015                          ERROR   15                  ;NO RDY AFTER WRITE CHECK CMD
5512  027350  004737  033566                  JSR     PC,GSTAT            ;GET FRESH STATUS
5513  027354  032737  100000  007340          BIT     #CERR,HCS1
5514  027362  001453                          BEQ     69$
5515  027364  032737  040000  007342          BIT     #WCE,HCS2           ;SEE IF WRITE CHECK ERROR
5516  027372  001410                          BEQ     68$
5517  027374  016537  000024  001436          MOV     RKDB(R5),WD1        ;ACTUAL WORD FOR PRINTOUT
5518  027402  013737  001466  001440          MOV     DPAT1,WD2                   ;EXPECTED WORD FOR TYPEOUT
5519  027410  104016                          ERROR   16                  ;WCE AFTER WRITE CMD
5520  027412  000437                          BR      69$
5521
5522  027414  104022                  68$:    ERROR   22                  ;CERR AFTER WRITE CHECK CMD
5523
5524  027416  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5525  027424  005037  007432                  CLR     E.B0                ;EXPECTED MSG B0
5526  027430  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
```

```
5527  027436  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5528  027444  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5529  027450  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5530  027456  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5531
5532  027464  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5533  027470  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5534  027472  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5535  027474  104031                          ERROR   31              ;MSH B0 ERROR
5536  027476  104060                          ERROR   60              ;MSG A1 ERROR
5537  027500  104032                          ERROR   32              ;MSG B1 ERROR
5538  027502  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5539  027506  000137  031514                  JMP     $EOP
5540
5541  027512                          69$:
5542
5543  027512  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG AC
5544  027520  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5545  027524  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5546  027532  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5547  027540  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5548  027544  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5549  027552  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5550
5551  027560  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5552  027564  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5553  027566  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5554  027570  104031                          ERROR   31              ;MSH B0 ERROR
5555  027572  104060                          ERROR   60              ;MSG A1 ERROR
5556  027574  104032                          ERROR   32              ;MSG B1 ERROR
5557  027576  022027  000577          3$:     CMP     (R0)+,#383.     ;ALL CYLINDERS DONE?
5558  027602  001402                          BEQ     4$              ;BR IF YES
5559  027604  000137  027000                  JMP     1$              ;ELSE REPEAT
5560
5561  027610                          4$:
5562
5563  027610  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5564  027616  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
5565  027624  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
5566                                          ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5567                                          ;IN RKMR2 & RKMR3 RESP.
5568  027632  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
5569  027640  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5570  027644  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
5571
5572  027646  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5573  027654  004737  033566                  JSR     PC,GSTAT
5574  027660  032737  020000  007366          BIT     #D.RTZ,HMR2
5575  027666  001001                          BNE     70$
5576  027670  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
5577  027672  013737  001406  007400  70$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
5578  027700  004737  032544                  JSR     PC,FATT1        ;FIND ATTN
5579  027704  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
5580
5581  027706  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5582  027714  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
```

```
5583  027722  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5584  027730  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
5585  027736  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5586  027742  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5587  027744  004737  032512                  JSR     PC,TSTATN       ;TEST FOR ATTN
5588  027750  000401                          BR      71$
5589  027752  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5590  027754                          71$:
5591
5592
5593  027754                          2$:
5594
5595                                  ;;*****************************************************************
5596                                  ;*TEST 17       TEST HEAD SWITCHING TIME
5597                                  ;*
5598                                  ;*      TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.
5599                                  ;*
5600                                  ;*      1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS
5601                                  ;*         TO SECTOR 25(8) IS ISSUED.
5602                                  ;*      2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL
5603                                  ;*         A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
5604                                  ;*      3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN
5605                                  ;*         THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO
5606                                  ;*         SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS
5607                                  ;*
5608                                  ;*      THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2
5609                                  ;*
5610                                  ;*      THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT
5611                                  ;*
5612                                  ;;*****************************************************************
5613  027754  000004                  TST17:  SCOPE
5614  027756  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5615  027764  012706  001100                  MOV     #STACK,SP
5616
5617  027770  005737  007506                  TST     DOTIM           ;BYPASS THIS TEST IF
5618  027774  001001                          BNE     1$              ;NEITHER L OR P CLOCK PRESENT
5619  027776  000520                          BR      TST20           ;;GO TO NEXT TEST
5620
5621  030000  012737  000025  007406  1$:     MOV     #25,TEMP5       ;HEAD 0, SECTOR 21 TO BE PUT IN RKDA
5622  030006  004737  034140          2$:     JSR     PC,$UBCLR
5623  030012  104024                          ERROR   24              ;CERR AFTER SCLR
5624
5625  030014  004737  034710                  JSR     PC,FSEC23       ;FIND SECTOR 23
5626  030020  104106                          ERROR   106             ;CANNOT FIND SECTOR 23
5627  030022  000506                          BR      TST20           ;;GO TO NEXT TEST
5628
5629  030024  012765  001462  000004          MOV     #DATAO1,RKBA(R5) ;DATA O101
5630  030032  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUSS ADDR INCREMENT INHIBIT
5631  030040  012765  177000  000002          MOV     #-512.,RKWC(R5) ;WORD COUNT
5632  030046  013765  007406  000006          MOV     TEMP5,RKDA(R5)  ;HEAD & SECTOR
5633  030054  012765  000023  000000          MOV     #WRDATA,RKCS1(R5) ;WRITE DATA CMD
5634
5635  030062  012727  002000  007376          MOV     #2000,#TEMP1
5636  030070  004737  032720                  JSR     PC,DLY          ;DO DELAY
5637
5638  030074  032765  000200  000000  7$:     BIT     #RDY,RKCS1(R5)  ;LOOK FOR CONTROLLER READY
```

```
5639  030102  001006                        BNE     8$
5640  030104  004737  032230                JSR     PC,FRDY         ;FIND RDY AND GET FRESH STATUS
5641  030110  104011                        ERROR   11              ;NO RDY AFTER SEL DRV CMD
5642  030112  004737  033566                JSR     PC,GSTAT
5643  030116  104107                        ERROR   107             ;HEAD SWITCHING LONGER THAN DELAY
5644
5645  030120  032737  100000  007340  8$:   BIT     #CERR,HCS1
5646  030126  001444                        BEQ     3$
5647  030130  104012                        ERROR   12              ;CERR AFTER WRITE DATA
5648
5649  030132  012737  010340  007430        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5650  030140  005037  007432                CLR     E.B0            ;EXPECTED MSG B0
5651  030144  012737  001720  007434        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5652  030152  012737  000001  007436        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5653  030160  005037  007440                CLR     E.A2            ;EXPECTED MSG A2
5654  030164  012737  000002  007442        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5655  030172  012737  000003  007446        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5656
5657  030200  004737  032752                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5658  030204  000000                        .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5659  030206  104052                        ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5660  030210  104023                        ERROR   23              ;MSH B0 ERROR
5661  030212  104053                        ERROR   53              ;MSG A1 ERROR
5662  030214  104025                        ERROR   25              ;MSG B1 ERROR
5663
5664  030216  023727  007406  000425        CMP     TEMP5,#425      ;HEAD 1,SECTOR 21 DONE?
5665  030224  001405                        BEQ     TST20           ;;GO TO NEXT TEST
5666  030226  012737  000425  007406        MOV     #425,TEMP5
5667  030234  000137  030006                JMP     2$              ;ELSE REPEAT FOR HEAD 1, SECTOR 21
5668  030240                          3$:
5669
5670                                  ;;*****************************************************************
5671                                  ;*TEST 20       DRIVE OFF TRACK TEST
5672                                  ;*
5673                                  ;*      THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND
5674                                  ;*      THE ALLOTED 3MS.
5675                                  ;*
5676                                  ;*      1.  INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS
5677                                  ;*          (UNIQUE TO EACH CYLINDER)
5678                                  ;*      2.  A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO 1.
5679                                  ;*          AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE
5680                                  ;*          HEADS WILL IMMEDIATELY BE WRITTEN.
5681                                  ;*      3.  IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS,
5682                                  ;*          DRIVE OFF TRACK ERROR WILL SET.
5683                                  ;*
5684                                  ;*      IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDERS.
5685                                  ;*      100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
5686                                  ;*      BEFORE DOING THE NEXT CYLINDER
5687                                  ;*
5688                                  ;;*****************************************************************
5689  030240  000004                  TST20:  SCOPE
5690  030242  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5691  030250  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5692  030254  005237  007324                  INC     BADHDR          ;USED FOR VALID HALT
5693  030260  005037  001346                  CLR     TOCYL
5694  030264  012737  100000  007406          MOV     #BIT15,TEMP5
```

```

030272  004737  034140           1$:     JSR     PC,SUBCLR
030276  104024                           ERROR   24                      ;CERR AFTER SCLR

030300  012700  001506                   MOV     #HDTAB,R0               ;FORMAT HEADERS ON ALL MAJOR CYL.

030304  013720  001346           2$:     MOV     TOCYL,(R0)+             ;HEADER WORD 0: CYL #
030310  012720  140000                   MOV     #140000,(R0)+           ;HEADER WORD 1: ALL SECTOR 0
030314  012710  140000                   MOV     #140000,(R0)            ;HEADER WORD 2: XOR OF 0 & 1
030320  053720  001346                   BIS     TOCYL,(R0)+             ;ADD CYL # TO WORD 2

030324  020027  001712                   CMP     R0,#HDTAB+132.          ;ALL 22 SECTORS DONE? (22X6=132)
030330  001365                           BNE     2$                      ;BR IF NO

030332  012765  001506  000004           MOV     #HDTAB,RKBA(R5)
030340  012765  177676  000002           MOV     #-66.,RKWC(R5)
030346  013765  001346  000020           MOV     TOCYL,RKDC(R5)

030354  012765  000027  000000           MOV     #<WRHEAD>,RKCS1(R5)         ;WRITE HEADER CMD
030362  013737  001420  007376           MOV     T50000,TEMP1            ;SETUP  TIMEOUT
030370  004737  032230                   JSR     PC,FRDY                 ;FIND RDY
030374  104200                           ERROR   200                     ;NO RDY AFTER WRITE HEADER CMD
030376  004737  033566                   JSR     PC,GSTAT                ;GET FRESH STATUS
030402  032737  100000  007340           BIT     #CERR,HCS1
030410  001405                           BEQ     64$
030412  104201                           ERROR   201                     ;CERR AFTER WRITE HEADER CMD
030414  104401  045277                   TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
030420  000137  031514                   JMP     $EOP
030424                           64$:



030424  006137  007406                   ROL     TEMP5                   ;SET CARRY ONLY ONCE
030430  006137  001346                   ROL     TOCYL                   ;SELECT NEXT MAJOR CYL
030434  023727  001346  001000           CMP     TOCYL,#1000             ;ALL MAJOR CYL FORMATTED?
030442  001313                           BNE     1$                      ;BR IF NO
030444  005065  000020                   CLR     RKDC(R5)                ;SETUP TO RETURN TO CYL 0

030450  012765  000017  000000           MOV     #SEEK,RKCS1(R5)         ;SEEK CMD
030456  013737  001410  007376           MOV     T50,TEMP1               ;SETUP TIMEOUT
030464  004737  032230                   JSR     PC,FRDY                 ;FIND RDY
030470  104131                           ERROR   131                     ;NO RDY AFTER SEEK CMD

030472  013737  001420  007376           MOV     T50000,TEMP1            ;SETUP TIMEOUT
030500  004737  032640                   JSR     PC,FATT2                ;FIND ATTN
030504  104132                           ERROR   132                     ;NO ATTN AFTER SEEK CMD

030506  032737  100000  007340           BIT     #CERR,HCS1
030514  001401                           BEQ     65$
030516  104210                           ERROR   210                     ;CERR AFTER SEEK CMD

030520                           65$:

030520  012737  031226  001176           MOV     #FORM,$ESCAPE
030526  005000                           CLR     R0                      ;ITERATION COUNTER
030530  012737  000001  001346           MOV     #1,TOCYL                ;SETUP TO CYL #
030536  005037  001344                   CLR     FRCYL
```

```
5751
5752  030542  104415                          SCOP1
5753  030544  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5754
5755  030550  013737  001346  001354          MOV     TOCYL,CALDIF    ;SETUP FOR ERROR PRINTOUT
5756
5757  030556  004737  034140          3$:     JSR     PC,SUBCLR
5758  030562  104024                          ERROR   24              ;CERR AFTER SCLR
5759
5760  030564  013765  001346  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
5761  030572  012765  001464  000004          MOV     #DATA1,RKBA(R5) ;ALL 1'S
5762  030600  052765  000020  000010          BIS     #BAI,RKCS2(R5)
5763  030606  012765  177400  000002          MOV     #-256.,RKWC(R5) ; SECTOR TO BE ALL 1'S
5764
5765  030614  012765  000023  000000          MOV     #WRDATA,RKCS1(R5) ;WRITE DATA CMD
5766  030622  013737  001420  007376          MOV     T50000,TEMP1
5767  030630  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5768  030634  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD.
5769
5770  030636  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
5771  030642  032737  020000  007370          BIT     #D.DROT,HMR3    ;SEE IF DRIVE OFF TRACK
5772  030650  001401                          BEQ     5$
5773  030652  104112                          ERROR   112             ;DRIVE OFF TRACK AFTER WRITE DATA CMD
5774
5775  030654  032737  100000  007340  5$:     BIT     #CERR,HCS1
5776  030662  001401                          BEQ     6$
5777  030664  104012                          ERROR   12              ;CERR SET AFTER WRITE DATA CMD
5778
5779  030666                          6$:
5780
5781  030666  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5782  030674  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5783  030700  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5784  030706  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5785  030714  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5786  030720  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5787  030726  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5788
5789  030734  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5790  030740  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5791  030742  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5792  030744  104023                          ERROR   23              ;MSH B0 ERROR
5793  030746  104053                          ERROR   53              ;MSG A1 ERROR
5794  030750  104025                          ERROR   25              ;MSG B1 ERROR
5795  030752  023737  001360  001346          CMP     CYLADD,TOCYL
5796  030760  001401                          BEQ     7$
5797  030762  104113                          ERROR   113             ;CYL ADDR IN RKMR3 NOT = RKDC
5798
5799  030764                          7$:
5800  030764  104415                          SCOP1
5801  030766  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5802
5803  030772  004737  034140                  JSR     PC,SUBCLR
5804  030776  104024                          ERROR   24              ;CERR AFTER SCLR
5805
5806                                          ;RETURN TO CYL 0
```

```
5807  031000  012765  001464  000004          MOV     #DATA1,RKBA(R5)
5808  031006  052765  000020  000010          BIS     #BAI,RKCS2(R5)
5809  031014  012765  177400  000002          MOV     #-256.,RKWC(R5)
5810
5811  031022  012765  000023  000000          MOV     #WRDATA,RKCS1(R5)
5812  031030  013737  001420  007376          MOV     T50000,TEMP1
5813  031036  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5814  031042  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
5815
5816  031044  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
5817  031050  032737  020000  007370          BIT     #D.DROT,HMR3
5818  031056  001401                          BEQ     8$
5819  031060  104112                          ERROR   112             ;DRIVE OFF TRACK AFTER WRITE DATA CMD
5820
5821  031062  032737  100000  007340  8$:     BIT     #CERR,HCS1
5822  031070  001401                          BEQ     9$
5823  031072  104012                          ERROR   12              ;CERR AFTER WRITE DATA CMD
5824
5825  031074                          9$:
5826
5827  031074  012737  010340  007430          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5828  031102  005037  007432                  CLR     E.B0            ;EXPECTED MSG B0
5829  031106  012737  001720  007434          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5830  031114  012737  000001  007436          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5831  031122  005037  007440                  CLR     E.A2            ;EXPECTED MSG A2
5832  031126  012737  000002  007442          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5833  031134  012737  000003  007446          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5834
5835  031142  004737  032752                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5836  031146  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5837  031150  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5838  031152  104023                          ERROR   23              ;MSH B0 ERROR
5839  031154  104053                          ERROR   53              ;MSG A1 ERROR
5840  031156  104025                          ERROR   25              ;MSG B1 ERROR
5841  031160  005737  001360                  TST     CYLADD
5842  031164  001401                          BEQ     10$
5843  031166  104042                          ERROR   42              ;NOT BACK TO CYL 0
5844
5845  031170  005200                  10$:    INC     R0
5846  031172  020027  000144                  CMP     R0,#100.        ;ALL ITERATIONS DONE?
5847  031176  001402                          BEQ     13$             ;BR IF YES
5848  031200  000137  030556                  JMP     3$              ;ELSE DO AGAIN
5849
5850  031204  005000                  13$:    CLR     R0              ;RESET ITERATION CTR
5851  031206  006337  001346                  ASL     TOCYL
5852  031212  023727  001346  001000          CMP     TOCYL,#1000     ;ALL MAJOR CYL DONE?
5853  031220  001402                          BEQ     FORM            ;BR IF YES
5854  031222  000137  030556                  JMP     3$              ;ELSE DO NEXT CYL
5855
5856  031226  005037  001346          FORM:   CLR     TOCYL           ;RESTORE TO ORIG 22 SECTOR FORMAT.
5857  031232  012737  100000  007406          MOV     #BIT15,TEMP5
5858
5859  031240  004737  034140          12$:    JSR     PC,SUBCLR
5860  031244  104024                          ERROR   24              ;CERR AFTER SCLR
5861
5862  031246  012765  001506  000004          MOV     #HDTAB,RKBA(R5) ;REFORMAT ALL MAJOR CYLINDERS
```

```
5863  031254  012765  177676  000002          MOV     #-66.,RKWC(R5)
5864  031262  013765  001346  000020          MOV     TOCYL,RKDC(R5)
5865
5866  031270  013737  001346  001362          MOV     TOCYL,CALADD    ;SETUP
5867  031276  012737  000000  001444          MOV     #0,HEAD         ;TO FILL
5868  031304  012737  000000  001452          MOV     #0,FORMAT       ;HEADER
5869  031312  004737  035122                  JSR     PC,FHDTAB       ;TABLE
5870
5871
5872  031316  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
5873  031324  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5874  031332  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5875  031336  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5876  031340  004737  033566                  JSR     PC,GSTAT        ;GET FRESH STATUS
5877  031344  032737  100000  007340          BIT     #CERR,HCS1
5878  031352  001405                          BEQ     64$
5879  031354  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
5880  031356  104401  045277                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5881  031362  000137  031514                  JMP     $EOP
5882  031366                          64$:
5883
5884
5885  031366  006137  007406                  ROL     TEMP5
5886  031372  006137  001346                  ROL     TOCYL
5887  031376  023727  001346  001000          CMP     TOCYL,#1000     ;ALL MAJOR CYL REFORMATTED?
5888  031404  001315                          BNE     12$             ;BR IF NO
5889
5890  031406  005065  000020                  CLR     RKDC(R5)        ;SETUP TO RETURN TO CYL 0
5891  031412  005037  001176                  CLR     $ESCAPE
5892
5893  031416  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
5894  031424  013737  001410  007376          MOV     T50,TEMP1       ;SETUP TIMEOUT
5895  031432  004737  032230                  JSR     PC,FRDY         ;FIND RDY
5896  031436  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
5897
5898  031440  013737  001420  007376          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5899  031446  004737  032640                  JSR     PC,FATT2        ;FIND ATTN
5900  031452  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
5901
5902  031454  032737  100000  007340          BIT     #CERR,HCS1
5903  031462  001401                          BEQ     65$
5904  031464  104210                          ERROR   210             ;CERR AFTER SEEK CMD
5905
5906  031466                          65$:
5907
5908
5909  031466  005737  007326                  TST     HPEND           ;SEE IF HALT PENDING
5910  031472  001406                          BEQ     4$              ;BR IF NO
5911  031474  005037  007326                  CLR     HPEND           ;CLEAR FOR FUTURE FORMATTING
5912  031500  005037  007324                  CLR     BADHDR          ;HEADERS NOW OK
5913  031504  000137  036172                  JMP     STOP            ;GO BACK & HALT CPU
5914
5915  031510  005037  007324          4$:     CLR     BADHDR          ;HEADERS NOW OK
5916
```

```
                        .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
031514                  $EOP:
031514  000004                  SCOPE
031516  012737  000001  001174  $TIMES
031524  012706  001100          MOV     #STACK,SP
031530  005237  001220          INC     $DEVCT          ;INCR COUNT FOR # DRIVES CHECKED
031534  023737  007460  001220  CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
031542  001403                  BEQ     $EOP1+2         ;BR IF YES
031544  000137  014764          JMP     NUDRV           ;ELSE TEST NEXT DRIVE PRESENT
031550  000004          $EOP1:  SCOPE
031552  005037  001102          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
031556  005037  001174          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
031562  005237  001216          INC     $PASS           ;;INCREMENT THE PASS NUMBER
031566  042737  100000  001216  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
031574  005327                  DEC     (PC)+           ;;LOOP?
031576  000001          $EOPCT: .WORD   1
031600  003022                  BGT     $DOAGN          ;;YES
031602  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
031604  000001          $ENDCT: .WORD   1
031606  031576                  $EOPCT
031610  104401  031655          TYPE    ,$ENDMG         ;;TYPE "END PASS #"
031614  013746  001216          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
031620  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
031622  104401  031652          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
031626  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
031632  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
031634  000005                  RESET                   ;;CLEAR THE WORLD
031636  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
031640  000240                  NOP                     ;;SAVE ROOM
031642  000240                  NOP                     ;;FOR
031644  000240                  NOP                     ;;ACT11
031646                  $DOAGN:
031646  000137                  JMP     @(PC)+          ;;RETURN
031650  013340          $RTNAD: .WORD   ST5
031652    377     377     000   $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
031655    015   042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
031662  050040  051501  020123
031670  000043
```

```
                                        .SBTTL  SUBROUTINES
                                        ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                        ;
031672  012700  007450          CLRFLG: MOV     #DDUMP,R0
031676  012701  177757                  MOV     #-17.,R1
031702  005020                  1$:     CLR     (R0)+
031704  005201                          INC     R1
031706  001375                          BNE     1$
031710  000207                          RTS     PC
                                        ;
                                        ;TYPE PROGRAM ID IF FTITLE=0
                                        ;
031712  005737  001340          TITLE:  TST     FTITLE
031716  001024                          BNE     1$
031720  005237  001340                  INC     FTITLE
031724  104401  043352                  TYPE    ,MSG1                   ;PROGRAM ID
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
031730  005737  000042                  TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
031734  001012                          BNE     64$                     ;;BRANCH IF YES
031736  123727  001230  000001          CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
031744  001406                          BEQ     64$                     ;;BRANCH IF YES
031746  023727  001140  000176          CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
031754  001005                          BNE     65$                     ;;BRANCH IF NO
031756  104406                          GTSWR                           ;;GET SOFT-SWR SETTINGS
031760  000403                          BR      65$
031762  112737  000001  001134  64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
031770                          65$:
031770  000207                  1$:     RTS     PC
                                        ;
                                        ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
                                        ;DRIVS, DRIV0-DRIV7 REGISTERS APPROPRIATELY
                                        ;
032772  104411                  GDRVS:  RDLIN
032774  012600                          MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
032776  012701  177770                  MOV     #-8.,R1                 ;SET UP COUNT
033002  112002                  1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
033004  042702  177400                  BIC     #177400,R2              ;MASK HI BYTE
033010  012703  007462                  MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
033014  012704  000060                  MOV     #60,R4
033020  020402                  2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
033022  001415                          BEQ     3$                      ;BRANCH IF YES
033024  005723                          TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
033026  005204                          INC     R4
033030  020427  000070                  CMP     R4,#70
033034  001371                          BNE     2$                      ;S/B 0-7 OR TERMINATOR
033036  005702                          TST     R2
033040  001022                          BNE     4$
033042  020127  177770                  CMP     R1,#-8.
033046  001426                          BEQ     6$                      ;DEFAULT ALL DRIVES
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
6019  032050  005037  007510          7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
6020  032054  000207                          RTS     PC              ;FOUND TERMINATOR, EXIT
6021
6022  032056  005213                  3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
6023  032060  005237  007460                  INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
6024  032064  112002                          MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
6025  032066  042702  177400                  BIC     #177400,R2      ;MASK
6026  032072  022702  000054                  CMP     #54,R2          ;IS IT A COMMA?
6027  032076  001407                          BEQ     5$              ;YES, GO TO NEXT WORD.
6028  032100  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
6029  032102  001001                          BNE     4$              ;IF NOT, SOMETHING WRONG.
6030  032104  000761                          BR      7$              ;FOUND TERMINATOR, EXIT
6031
6032  032106  104401  046062          4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
6033  032112  000137  012546                  JMP     PRGSRT          ;START ALL OVER
6034
6035  032116  005201                  5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
6036  032120  001330                          BNE     1$              ;DRIVES TYPED IN.
6037  032122  000771                          BR      4$              ;IF MORE, HAVE ERROR.
6038
6039  032124  005237  007510          6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
6040  032130  000207                          RTS     PC              ;EXIT.
6041
6042                                          ;
6043                                          ;ROUTINE TO INPUT RKBAS OR DEFAULT.
6044                                          ;
6045
6046  032132  104412                  GBA:    RDOCT
6047  032134  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
6048  032136  005700                          TST     R0
6049  032140  001403                          BEQ     1$              ;BRANCH IF DEFAULT.
6050  032142  010037  001264                  MOV     R0,$BASE
6051  032146  000207                          RTS     PC
6052  032150  012737  177440  001264  1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
6053  032156  000207                          RTS     PC
6054
6055                                          ;
6056                                          ;ROUTINE TO INPUT RKVEC OR DEFAULT
6057                                          ;
6058
6059  032160  104412                  GINT:   RDOCT
6060  032162  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
6061  032164  005700                          TST     R0
6062  032166  001405                          BEQ     1$              ;BRANCH IF DEFAULT
6063  032170  010037  001314                  MOV     R0,RKVEC
6064  032174  004737  032212          2$:     JSR     PC,SETINT
6065  032200  000207                          RTS     PC
6066  032202  012737  000210  001314  1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
6067  032210  000771                          BR      2$
6068
6069                                          ;
6070                                          ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
6071                                          ;
6072
6073  032212  013700  001314          SETINT: MOV     RKVEC,R0
6074  032216  012720  036652                  MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
6075  032222  013710  001316                  MOV     RKPRI,(R0)      ;PR5 TO RKVEC+2
6076  032226  000207                          RTS     PC
6077
6078
6079                                  ;
6080                                  ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
6081                                  ;ENTER WITH A COUNT IN TEMP1
6082                                  ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
6083                                  ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
6084                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6085                                  ;
6086  032230  032765  000200  000000  FRDY:   BIT     #RDY,RKCS1(R5)
6087  032236  001010                          BNE     1$
6088  032240  005337  007376                  DEC     TEMP1
6089  032244  001371                          BNE     FRDY
6090  032246  004737  032364                  JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
6091  032252  004737  033504                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
6092  032256  000207                          RTS     PC              ;NO RDY, EXIT
6093  032260  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
6094  032264  004737  032364                  JSR     PC,HOLD
6095  032270  004737  033504                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
6096  032274  000207                          RTS     PC
6097                                  ;
6098                                  ;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
6099                                  ;
6100  032276  032765  000200  000000  FRDY1:  BIT     #RDY,RKCS1(R5)
6101  032304  001014                          BNE     1$
6102  032306  005337  007376                  DEC     TEMP1
6103  032312  001371                          BNE     FRDY1
6104  032314  016537  000034  007366          MOV     RKMR2(R5),HMR2
6105  032322  016537  000036  007370          MOV     RKMR3(R5),HMR3
6106  032330  004737  033504                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
6107  032334  000207                          RTS     PC              ;NO RDY, EXIT
6108  032336  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
6109  032342  016537  000034  007366          MOV     RKMR2(R5),HMR2
6110  032350  016537  000036  007370          MOV     RKMR3(R5),HMR3
6111  032356  004737  033504                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
6112  032362  000207                          RTS     PC
6113
6114                                  ;
6115                                  ;STORE ALL RK611 REGISTERS IN HOLDING REGS
6116                                  ;
6117
6118  032364  016537  000000  007340  HOLD:   MOV     RKCS1(R5),HCS1
6119  032372  016537  000010  007342          MOV     RKCS2(R5),HCS2
6120  032400  016537  000002  007344          MOV     RKWC(R5),HWC
6121  032406  016537  000004  007346          MOV     RKBA(R5),HBA
6122  032414  016537  000006  007350          MOV     RKDA(R5),HDA
6123  032422  016537  000012  007352          MOV     RKDS(R5),HDS
6124  032430  016537  000014  007354          MOV     RKER(R5),HER
6125  032436  016537  000016  007356          MOV     RKASOF(R5),HASOF
6126  032444  016537  000020  007360          MOV     RKDC(R5),HDC
6127  032452  016537  000026  007364          MOV     RKMR1(R5),HMR1
6128  032460  016537  000034  007366          MOV     RKMR2(R5),HMR2
6129  032466  016537  000036  007370          MOV     RKMR3(R5),HMR3
6130  032474  016537  000030  007372          MOV     RKECPS(R5),HPOS
```

```
6131  032502  016537  000032  007374          MOV     RKECPT(R5),HPAT
6132  032510  000207                          RTS     PC
6133
6134                                  ;
6135                                  ;ROUTINE TO CHECK FOR CORRECT ATTN
6136                                  ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
6137                                  ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
6138                                  ;
6139  032512  010446                  TSTATN: MOV     R4,-(SP)        ;SAV R4
6140  032514  013704  001222                  MOV     $UNIT,R4
6141  032520  136437  007330  007357          BITB    ATTN(R4),HASOF+1
6142  032526  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
6143  032530  012604                          MOV     (SP)+,R4        ;RESTOR R4
6144  032532  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
6145  032536  000207                          RTS     PC
6146  032540  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
6147  032542  000207                          RTS     PC
6148
6149                                  ;
6150                                  ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
6151                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
6152                                  ;RETURN IF NO ATTN (ERROR CONDITION)
6153                                  ;RETURN +2 IF ATTN FOUND
6154                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6155                                  ;
6156
6157  032544  010446                  FATT1:  MOV     R4,-(SP)        ;SAV R4
6158  032546  012737  177777  007376  3$:     MOV     #-1,TEMP1
6159  032554  013704  001222                  MOV     $UNIT,R4
6160  032560  136465  007330  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
6161  032566  001014                          BNE     2$
6162  032570  005337  007376                  DEC     TEMP1
6163  032574  001371                          BNE     1$
6164  032576  005337  007400                  DEC     TEMP2
6165  032602  001361                          BNE     3$
6166  032604  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
6167  032610  004737  033566                  JSR     PC,GSTAT        ;GET LATEST STATUS
6168  032614  012604                          MOV     (SP)+,R4        ;RESTOR R4
6169  032616  000207                          RTS     PC
6170  032620  005065  000026          2$:     CLR     RKMR1(R5)
6171  032624  004737  033566                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
6172  032630  012604                          MOV     (SP)+,R4        ;RESTOR R4
6173  032632  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
6174  032636  000207                          RTS     PC
6175
6176                                  ;
6177                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
6178                                  ;ENTER WITH COUNT IN TEMP1
6179                                  ;RETURN IF NO ATTN (ERROR)
6180                                  ;RETURN +2 IF ATTN FOUND
6181                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6182                                  ;
6183
6184  032640  010446                  FATT2:  MOV     R4,-(SP)        ;SAV R4
6185  032642  013704  001222          2$:     MOV     $UNIT,R4
6186  032646  136465  007330  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
6187  032654  001011                  BNE     1$
6188  032656  005337  007376          DEC     TEMP1
6189  032662  001367                  BNE     2$
6190  032664  005065  000026          CLR     RKMR1(R5)       ;SELECT WORD 0
6191  032670  004737  033566          JSR     PC,GSTAT        ;GET LATEST STATUS.
6192  032674  012604                  MOV     (SP)+,R4        ;RESTOR R4
6193  032676  000207                  RTS     PC
6194  032700  005065  000026  1$:     CLR     RKMR1(R5)
6195  032704  004737  033566          JSR     PC,GSTAT
6196  032710  012604                  MOV     (SP)+,R4        ;RESTOR R4
6197  032712  062716  000002          ADD     #2,(SP)         ;SKIP OVER ERROR
6198  032716  000207                  RTS     PC
6199                                  ;
6200                                  ;ENTER WITH A COUNT IN TEMP1
6201                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
6202                                  ;WHEN COUNT IS 0. BASED ON AN 11/05
6203                                  ;
6204  032720  005737  007376  DLY:    TST     TEMP1           ;5.6 US
6205  032724  001403                  BEQ     1$              ;1.9 US
6206  032726  005337  007376          DEC     TEMP1           ;6.8 US
6207  032732  000772                  BR      DLY             ;2.5 US
6208  032734  000207          1$:     RTS     PC              ;3.8 US
6209                                  ;
6210                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
6211                                  ;
6212
6213  032736  104401  045166  BYP:    TYPE    ,MSG14          ;BYPASS DRIVE
6214  032742  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
6215                                                          ;;TYPE DR#
6216  032744  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
6217  032746     001                  .BYTE   1               ;;TYPE 1 DIGIT(S)
6218  032747     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
6219  032750  000207                  RTS     PC
6220                                  ;
6221                                  ;THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
6222                                  ;
6223  032752  017637  000000  001504  CHKMSG: MOV     @(SP),CHKFLG    ;PASS MSGS TO BE TESTED
6224  032760  062716  000002          ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
6225  032764  004737  033630          JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS
6226
6227  032770  053737  001222  007430  BIS     $UNIT,E.A0      ;SET UNIT #
6228  032776  053737  001222  007434  BIS     $UNIT,E.A1
6229  033004  053737  001222  007440  BIS     $UNIT,E.A2
6230  033012  053737  001222  007444  BIS     $UNIT,E.A3
6231
6232  033020  013746  007376          MOV     TEMP1,-(SP)     ;SAVE TEMP1
6233
6234  033024  013737  007430  007376  MOV     E.A0,TEMP1
6235  033032  004737  036066          JSR     PC,SBPAR        ;GET PARITY FOR MSG A0
6236  033036  013737  007376  007430  MOV     TEMP1,E.A0
6237
6238  033044  013737  007434  007376  MOV     E.A1,TEMP1
6239  033052  004737  036066          JSR     PC,SBPAR        ;GET PARITY FOR MSG A1
6240  033056  013737  007376  007434  MOV     TEMP1,E.A1
6241
6242  033064  013737  007440  007376  MOV     E.A2,TEMP1
```

```
6243  033072  004737  036066            JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
6244  033076  013737  007376  007440    MOV     TEMP1,E.A2
6245
6246  033104  013737  007432  007376    MOV     E.B0,TEMP1
6247  033112  004737  036066            JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
6248  033116  013737  007376  007432    MOV     TEMP1,E.B0
6249
6250  033124  013737  007436  007376    MOV     E.B1,TEMP1
6251  033132  004737  036066            JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
6252  033136  013737  007376  007436    MOV     TEMP1,E.B1
6253
6254  033144  013737  007442  007376    MOV     E.B2,TEMP1
6255  033152  004737  036066            JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
6256  033156  013737  007376  007442    MOV     TEMP1,E.B2
6257
6258  033164  013737  007446  007376    MOV     E.B3,TEMP1
6259  033172  004737  036066            JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
6260  033176  013737  007376  007446    MOV     TEMP1,E.B3
6261
6262  033204  012637  007376            MOV     (SP)+,TEMP1     ;RESTORE TEMP1
6263  033210  013737  001176  001172    MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE
6264
6265  033216  023737  007410  007430    CMP     H.A0,E.A0       ;TEST MSG A0
6266  033224  001411                    BEQ     2$              ;BR IF OK
6267  033226  012737  033240  001176    MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
6268  033234  011646                    MOV     (SP),-(SP)      ;COPY RET ADDR.
6269  033236  000207                    RTS     PC              ;& RETURN TO MAINLINE ERROR
6270
6271  033240  032777  001000  145672  1$:     BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
6272  033246  001107                    BNE     20$             ;& BR IF LOOP ON ERROR
6273  033250  062716  000002          2$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
6274
6275  033254  023737  007412  007432    CMP     H.B0,E.B0       ;TEST MSG B0
6276  033262  001411                    BEQ     5$              ;BR IF OK
6277  033264  012737  033276  001176    MOV     #4$,$ESCAPE     ;ELSE SETUP ESCAPE
6278  033272  011646                    MOV     (SP),-(SP)      ;COPY RET ADDR
6279  033274  000207                    RTS     PC              ;& RETURN TO MAINLINE ERROR
6280
6281  033276  032777  001000  145634  4$:     BIT     #SW9,@SWR       ;RETURN HERE FROM MAINLINE ERROR
6282  033304  001070                    BNE     20$             ;& BR IF LOOP ON ERROR
6283  033306  062716  000002          5$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
6284
6285  033312  023737  007414  007434    CMP     H.A1,E.A1       ;TEST MSG A1
6286  033320  001411                    BEQ     8$              ;BR IF OK
6287  033322  012737  033334  001176    MOV     #7$,$ESCAPE
6288  033330  011646                    MOV     (SP),-(SP)
6289  033332  000207                    RTS     PC
6290
6291  033334  032777  001000  145576  7$:     BIT     #SW9,@SWR
6292  033342  001051                    BNE     20$
6293  033344  062716  000002          8$:     ADD     #2,(SP)
6294
6295  033350  023737  007416  007436    CMP     H.B1,E.B1       ;TEST MSG B1
6296  033356  001411                    BEQ     11$             ;BR IF OK
6297  033360  012737  033372  001176    MOV     #10$,$ESCAPE
6298  033366  011646                    MOV     (SP),-(SP)
```

```
6299  033370  000207                          RTS     PC
6300
6301  033372  032777  001000  145540  10$:    BIT     #SW9,@SWR
6302  033400  001032                          BNE     20$
6303  033402  062716  000002          11$:    ADD     #2,(SP)
6304
6305  033406  032737  000001  001504  12$:    BIT     #T.A2,CHKFLG    ;TEST MSG A2?
6306  033414  001402                          BEQ     13$             ;BR IF NO
6307  033416  004737  034514                  JSR     PC,RCYLD        ;PUT INFO CYLDIF, DO NOT CHECK
6308  033422  032737  000002  001504  13$:    BIT     #T.B2,CHKFLG    ;TEST MSG B2?
6309  033430  001402                          BEQ     14$             ;BR IF NO
6310  033432  004737  034566                  JSR     PC,RCYLA        ;PUT INFO IN CYLADD, DO NOT CHECK
6311
6312  033436  032737  000004  001504  14$:    BIT     #T.B3,CHKFLG    ;TEST MSG B3?
6313  033444  001404                          BEQ     15$
6314  033446  004737  034624                  JSR     PC,RSEC         ;PUT INFO IN SECTOR, DO NOT CHECK
6315  033452  004737  034662                  JSR     PC,RHEAD        ;PUT INFO IN HEADA, DO NOT CHECK
6316
6317  033456  013737  001172  001176  15$:    MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
6318  033464  000207                          RTS     PC
6319
6320  033466  012706  001100          20$:    MOV     #STACK,SP       ;RESET STACK PTR
6321  033472  013737  001172  001176          MOV     $TMP5,$ESCAPE   ;RESTORE ESCAPE
6322  033500  000177  145404                  JMP     @$LPERR
6323                                  ;
6324                                  ;THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
6325                                  ;I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
6326                                  ;
6327  033504  005737  001502          CKCERR: TST     BYPCERR
6328  033510  001025                          BNE     4$
6329  033512  032737  100000  007340          BIT     #CERR,HCS1
6330  033520  001001                          BNE     1$              ;BR IF CERR
6331  033522  000207                          RTS     PC
6332
6333  033524  032737  004000  007340  1$:     BIT     #CTO,HCS1
6334  033532  001402                          BEQ     2$              ;BR IF NOT CTO
6335  033534  104125                          ERROR   125             ;CTO ERROR, MSG A & B INVALID
6336  033536  000207                          RTS     PC
6337
6338  033540  032737  010000  007342  2$:     BIT     #NED,HCS2
6339  033546  001401                          BEQ     3$              ;BR IF NOT NED
6340  033550  104126                          ERROR   126             ;NED ERROR, MSG A & B INVALID
6341
6342  033552  032737  001000  007342  3$:     BIT     #MDS,HCS2
6343  033560  001401                          BEQ     4$
6344  033562  104127                          ERROR   127             ;MDS ERROR, MSG A & B INVALID
6345
6346  033564  000207                  4$:     RTS     PC
6347
6348                                  ;
6349                                  ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
6350                                  ;IT THEN WAITS FOR CONTROLLER READY
6351                                  ;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
6352                                  ;
6353
6354  033566  013746  007376          GSTAT:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
```

```
 6355  033572  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 6356  033600  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;GET STATUS WITH SELECT DRIVE CMD
 6357  033606  013737  001406  007376          MOV     T10,TEMP1
 6358  033614  004737  032230                  JSR     PC,FRDY         ;FIND RDY
 6359  033620  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRIVE CMD
 6360  033622  012637  007376                  MOV     (SP)+,TEMP1     ;RESTOR TEMP1.
 6361  033626  000207                          RTS     PC
 6362                                  ;
 6363                                  ;THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
 6364                                  ;& ALL CONTROLLER REGISTERS.
 6365                                  ;
 6366  033630  013746  007376          GSTAT1: MOV     TEMP1,-(SP)     ;SAVE TEMP1
 6367  033634  004737  032364                  JSR     PC,HOLD         ;GET ALL CONTR REG
 6368  033640  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
 6369  033646  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 6370  033654  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT WORD 3
 6371  033662  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;GET STATUS
 6372  033670  013737  001406  007376          MOV     T10,TEMP1
 6373  033676  004737  032276                  JSR     PC,FRDY1                ;FIND RDY & STORE DRIVE REGS ONLY
 6374  033702  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
 6375  033704  013737  007366  007424          MOV     HMR2,H.A3       ;STORE MSG A3
 6376  033712  013737  007370  007426          MOV     HMR3,H.B3       ;STORE MSG B3
 6377
 6378  033720  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6379  033726  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 6380  033734  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
 6381  033742  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
 6382  033750  013737  001406  007376          MOV     T10,TEMP1
 6383  033756  004737  032276                  JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
 6384  033762  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
 6385  033764  013737  007366  007420          MOV     HMR2,H.A2       ;STORE MSG A2
 6386  033772  013737  007370  007422          MOV     HMR3,H.B2       ;STORE MSG B2
 6387
 6388  034000  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6389  034006  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 6390  034014  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 6391  034022  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
 6392  034030  013737  001406  007376          MOV     T10,TEMP1
 6393  034036  004737  032276                  JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
 6394  034042  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
 6395  034044  013737  007366  007414          MOV     HMR2,H.A1       ;STORE MSG A1
 6396  034052  013737  007370  007416          MOV     HMR3,H.B1       ;STORE MSG B1
 6397
 6398  034060  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6399  034066  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 6400  034074  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;SELECT WORD 0
 6401  034102  013737  001406  007376          MOV     T10,TEMP1
 6402  034110  004737  032276                  JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
 6403  034114  104117                          ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
 6404  034116  013737  007366  007410          MOV     HMR2,H.A0       ;STORE MSG A0
 6405  034124  013737  007370  007412          MOV     HMR3,H.B0       ;STORE MSG B0
 6406
 6407  034132  012637  007376                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
 6408  034136  000207                          RTS     PC
 6409
 6410                                  ;
```

```
                                        ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
                                        ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
                                        ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
                                        ;RETURN IF CERR SET
                                        ;RETURN +2 IF CERR CLEAR
                                        ;
034140  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
034146  013737  001406  007376          MOV     T10,TEMP1
034154  004737  032230                  JSR     PC,FRDY         ;FIND RDY
034160  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
034162  013765  001222  000010          MOV     SUNIT,RKCS2(R5) ;CURRENT DRIVE #
034170  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
034174  004737  033566                  JSR     PC,GSTAT        ;GET STATUS
034200  032737  100000  007340          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
034206  001401                          BEQ     1$
034210  000207                          RTS     PC
034212  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
034216  000207                          RTS     PC
                                        ;
                                        ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                        ;
034220  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
034226  004737  033566                  JSR     PC,GSTAT
034232  013737  007370  001402          MOV     HMR3,SECTOR
034240  042737  177017  001402          BIC     #^C<M.SECT>,SECTOR
034246  006237  001402                  ASR     SECTOR          ;RIGHT JUSTIFY
034252  006237  001402                  ASR     SECTOR          ;SECTOR
034256  006237  001402                  ASR     SECTOR          ;INFO
034262  006237  001402                  ASR     SECTOR
034266  000207                          RTS     PC
                                        ;
                                        ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                        ;
034270  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
034276  004737  033566                  JSR     PC,GSTAT
034302  013737  007366  001356          MOV     HMR2,CYLDIF
034310  042737  160017  001356          BIC     #^C<M.CDIF>,CYLDIF
034316  006237  001356                  ASR     CYLDIF          ;RIGHT JUSTIFY
034322  006237  001356                  ASR     CYLDIF          ;CYL DIFF/OFFSET
034326  006237  001356                  ASR     CYLDIF          ;INFO
034332  006237  001356                  ASR     CYLDIF
034336  023727  001356  000777          CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
034344  001002                          BNE     1$              ;BR IF NOT
034346  005037  001356                  CLR     CYLDIF          ;CLR IF YES
034352  000207                  1$:     RTS     PC
                                        ;
                                        ;QUICK SELECT DRIVE COMMAND TO OBTAIN CYL DIFF
                                        ;
034354  013746  007376          QKCYLD: MOV     TEMP1,-(SP)     ;SAVE TEMP1
034360  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
034366  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)       ;SELECT DRIVE CMD
034374  013737  001406  007376          MOV     T10,TEMP1
034402  032765  000200  000000  1$:     BIT     #RDY,RKCS1(R5)  ;TEST FOR CONT RDY
034410  001004                          BNE     2$              ;BR IF THERE
```

```
                                        DEC     TEMP1
                                        BNE     1$
                                        ERROR   117             ;NO RDY AFTER SEL DRV CMD

                        001356  2$:     MOV     RKMR2(R5),CYLDIF
                        001356          BIC     #^C<M.CDIF>,CYLDIF      ;GET CYL DIFF ONLY (NO SHIFTING)
                                        MOV     (SP)+,TEMP1     ;RESTORE TEMP1
                                        RTS     PC
                                ;
                                ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
                                ;
                        000026  RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
                                        JSR     PC,GSTAT
                        001360          MOV     HMR3,CYLADD
                        001360          BIC     #^C<M.CADD>,CYLADD
                        001360          ASR     CYLADD          ;RIGHT JUSTIFY
                        001360          ASR     CYLADD          ;CYL ADDR
                        001360          ASR     CYLADD          ;INFO
                        001360          ASR     CYLADD
                                        RTS     PC
                                ;
                                ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                ;
                        001356  RCYLD:  MOV     H.A2,CYLDIF
                        001356          BIC     #^C<M.CDIF>,CYLDIF  ;CLEAR UNWANTED INFO
                        001356          ASR     CYLDIF  ;RIGHT JUSTIFY
                        001356          ASR     CYLDIF
                        001356          ASR     CYLDIF
                        001356          ASR     CYLDIF
                        001356  000777  CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
                                        BNE     1$              ;BR IF NO
                        001356          CLR     CYLDIF          ;ELSE CLEAR
                                1$:     RTS     PC
                                ;
                                ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
                                ;
                        001360  RCYLA:  MOV     H.B2,CYLADD
                        001360          BIC     #^C<M.CADD>,CYLADD  ;CLEAR UNWANTED INFO
                        001360          ASR     CYLADD          ;RIGHT JUSTIFY
                        001360          ASR     CYLADD
                        001360          ASR     CYLADD
                        001360          ASR     CYLADD
                                        RTS     PC
                                ;
                                ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                ;
                        001402  RSEC:   MOV     H.B3,SECTOR
                        001402          BIC     #^C<M.SECT>,SECTOR  ;CLEAR UNWANTED INFO
                        001402          ASR     SECTOR          ;RIGHT JUSTIFY
                        001402          ASR     SECTOR
                        001402          ASR     SECTOR
                        001402          ASR     SECTOR
                                        RTS     PC
                                ;
                                ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
                                ;
```

```
6523  034662  013737  007426  001446  RHEAD:  MOV     H.B3,HEADA
6524  034670  042737  170777  001446          BIC     #^C<M.HEAD>,HEADA  ;CLEAR UNWANTED INFO
6525  034676  006237  001446                  ASR     HEADA              ;RIGHT JUSTIFY IT
6526  034702  000337  001446                  SWAB    HEADA
6527  034706  000207                          RTS     PC
6528                                  ;
6529                                  ;FIND SECTOR 23
6530                                  ;RETURN IF NOT FOUND
6531                                  ;RETURN +4 IF FOUND
6532                                  ;
6533  034710  013737  001416  007376  FSEC23: MOV     T5000,TEMP1        ;SETUP TIMEOUT
6534  034716  004737  034220          1$:     JSR     PC,RDSEC           ;READ SECTOR
6535  034722  023727  001402  000023          CMP     SECTOR,#23         ;TEST FOR SECTOR 23(8)
6536  034730  001014                          BNE     2$                 ;BR IF NOT 23(8)
6537
6538  034732  004737  034220                  JSR     PC,RDSEC
6539  034736  023727  001402  000023          CMP     SECTOR,#23
6540  034744  001412                          BEQ     3$                 ;BR IF READ SAME TWICE
6541  034746  004737  034220                  JSR     PC,RDSEC           ;ELSE TRY 1 MORE TIME
6542  034752  023727  001402  000023          CMP     SECTOR,#23
6543  034760  001404                          BEQ     3$                 ;BR IF 23(8)
6544
6545  034762  005337  007376          2$:     DEC     TEMP1
6546  034766  001353                          BNE     1$                 ;TRY AGAIN
6547  034770  000207                          RTS     PC
6548
6549  034772  062716  000004          3$:     ADD     #4,(SP)            ;SKIP OVER ERROR
6550  034776  000207                          RTS     PC
6551                                  ;
6552                                  ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
6553                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
6554                                  ;RETURN IF NOT FOUND
6555                                  ;RETURN+2 IF FOUND - SKIP OVER ERROR
6556                                  ;
6557  035000  012737  177777  007376  FHDHM:  MOV     #-1,TEMP1          ;ALL 1'S
6558  035006  012765  000001  000026          MOV     #1,RKMR1(R5)       ;WORD 1
6559  035014  004737  033566          1$:     JSR     PC,GSTAT
6560  035020  032737  000040  007366          BIT     #D.HDHM,HMR2
6561  035026  001007                          BNE     2$
6562  035030  005337  007376                  DEC     TEMP1
6563  035034  001367                          BNE     1$
6564  035036  005337  007400                  DEC     TEMP2
6565  035042  001356                          BNE     FHDHM
6566  035044  000207                          RTS     PC
6567  035046  062716  000002          2$:     ADD     #2,(SP)            ;SKIP OVER ERROR
6568  035052  000207                          RTS     PC
6569                                  ;
6570                                  ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
6571                                  ;RETURN IF NOT FOUND
6572                                  ;RETURN+2 IF FOUND: SKIP OVER ERROR
6573                                  ;
6574  035054  012737  000372  007376  FLOAD:  MOV     #250.,TEMP1
6575  035062  012765  000001  000026          MOV     #1,RKMR1(R5)       ;WORD 1
6576  035070  004737  033566          1$:     JSR     PC,GSTAT
6577  035074  032737  010000  007366          BIT     #D.LOAD,HMR2
6578  035102  001004                          BNE     2$
```

```
 6579  035104  005337  007376            DEC     TEMP1
 6580  035110  001367                    BNE     1$
 6581  035112  000207                    RTS     PC
 6582  035114  062716  000002    2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 6583  035120  000207                    RTS     PC
 6584                                    ;
 6585                                    ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
 6586                                    ;ENTER WITH CYL # IN 'CALADD'
 6587                                    ;ENTER WITH HEAD # IN 'HEAD'
 6588                                    ;ENTER WITH FORMAT IN 'FORMAT'
 6589                                    ;
 6590  035122  010046            FHDTAB: MOV     R0,-(SP)        ;SAV R0
 6591  035124  010146                    MOV     R1,-(SP)        ;SAV R1
 6592  035126  012700  001506            MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
 6593  035132  005001                    CLR     R1              ;SECTOR COUNTER
 6594  035134  013737  001444  001450    MOV     HEAD,HD1
 6595  035142  006337  001450            ASL     HD1
 6596  035146  006337  001450            ASL     HD1
 6597  035152  006337  001450            ASL     HD1
 6598  035156  006337  001450            ASL     HD1
 6599  035162  006337  001450            ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
 6600  035166  013737  001452  001454    MOV     FORMAT,FMT1
 6601  035174  000337  001454            SWAB    FMT1
 6602  035200  006337  001454            ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
 6603
 6604  035204  013720  001362    1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
 6605  035210  010110                    MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
 6606  035212  053710  001450            BIS     HD1,(R0)        ;             -HEAD NO
 6607  035216  053710  001454            BIS     FMT1,(R0)       ;             -FORMAT
 6608  035222  004737  035302            JSR     PC,SECFLG
 6609
 6610  035226  013737  001362  007376    MOV     CALADD,TEMP1
 6611  035234  011037  007400            MOV     (R0),TEMP2
 6612  035240  043737  001362  007400    BIC     CALADD,TEMP2
 6613  035246  042037  007376            BIC     (R0)+,TEMP1
 6614  035252  053737  007376  007400    BIS     TEMP1,TEMP2
 6615  035260  013720  007400            MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
 6616
 6617  035264  005201                    INC     R1              ;SECTOR CTR
 6618  035266  020127  000026            CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
 6619  035272  001344                    BNE     1$              ;BR IF NO
 6620
 6621  035274  012601                    MOV     (SP)+,R1        ;RESTOR R1
 6622  035276  012600                    MOV     (SP)+,R0        ;RESTOR R0
 6623  035300  000207                    RTS     PC
 6624                                    ;
 6625                                    ;THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
 6626                                    ;TEST & SETS BITS 14 & 15 APPROPRIATLY.
 6627                                    ;
 6628  035302  010246            SECFLG: MOV     R2,-(SP)        ;SAVE R2
 6629  035304  005737  001452            TST     FORMAT
 6630  035310  001016                    BNE     1$              ;BR IF 20 SECTOR FORMAT
 6631  035312  012702  003332            MOV     #BSE22H+8.,R2
 6632  035316  004737  035402            JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
 6633  035322  052710  100000            BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
 6634
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
6635  035326  012702  005332          MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
6636  035332  004737  035402          JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
6637  035336  052710  040000          BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
6638
6639  035342  012602                  MOV     (SP)+,R2        ;ELSE RETURN HERE
6640  035344  000207                  RTS     PC
6641
6642  035346  012702  002332  1$:     MOV     #BSE20H+8.,R2
6643  035352  004737  035402          JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
6644  035356  052710  100000          BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
6645
6646  035362  012702  004332          MOV     #BSE20S+8.,R2
6647  035366  004737  035402          JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
6648  035372  052710  040000          BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
6649
6650  035376  012602                  MOV     (SP)+,R2        ;RESTORE R2
6651  035400  000207                  RTS     PC
6652
6653                                  ;
6654                                  ;THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
6655                                  ;ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
6656                                  ;RETURN IF NO COMPARE
6657                                  ;RETURN+4 IF COMPARE
6658                                  ;
6659  035402  010346          FLGTST: MOV     R3,-(SP)        ;SAVE R3
6660
6661  035404  021227  177777  1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
6662  035410  001002                  BNE     2$              ;BR IF NO
6663  035412  012603                  MOV     (SP)+,R3        ;RESTORE R3
6664  035414  000207                  RTS     PC
6665
6666  035416  022237  001362  2$:     CMP     (R2)+,CALADD    ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
6667  035422  001403                  BEQ     3$
6668  035424  062702  000002          ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
6669  035430  000765                  BR      1$
6670
6671  035432  013703  001444  3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
6672  035436  000303                  SWAB    R3
6673  035440  050103                  BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
6674  035442  022203                  CMP     (R2)+,R3        ;SEE IF SECTOR/HEAD COMPARE
6675                                                          ;& INCR PTR TO NEXT CYL WORD
6676  035444  001401                  BEQ     4$              ;BR IF COMPARE
6677  035446  000756                  BR      1$              ;ELSE TRY NEXT CYL
6678
6679  035450  012603          4$:     MOV     (SP)+,R3        ;RESTORE R3
6680  035452  062716  000004          ADD     #4,(SP)         ;INCREMENT RET ADDR
6681  035456  000207                  RTS     PC
6682
6683                                  ;
6684                                  ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
6685                                  ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
6686                                  ;
6687  035460  010046          SORT:   MOV     R0,-(SP)        ;SAVE R0
6688  035462  010146                  MOV     R1,-(SP)        ;SAVE R1
6689  035464  004737  034220          JSR     PC,RDSEC
6690  035470  062737  000001  001402  ADD     #1,SECTOR
```

```
6691  035476  004737  035566              JSR     PC,MULT6        ;MULT SECTOR BY 6
6692
6693  035502  012700  000204              MOV     #132.,R0
6694  035506  163700  001402              SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
6695  035512  010037  001402              MOV     R0,SECTOR
6696  035516  062737  001712  001402      ADD     #RHTAB,SECTOR   ;SAVE INDEX
6697
6698  035524  062700  001712              ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
6699  035530  012701  002116              MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
6700
6701  035534  012021              1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
6702  035536  020027  002116              CMP     R0,#RHTAB+132.
6703  035542  001374                      BNE     1$
6704
6705  035544  012700  001712              MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
6706  035550  012021              2$:     MOV     (R0)+,(R1)+
6707  035552  020037  001402              CMP     R0,SECTOR
6708  035556  001374                      BNE     2$
6709
6710  035560  012601                      MOV     (SP)+,R1        ;RESTOR R1
6711  035562  012600                      MOV     (SP)+,R0        ;RESTOR R0
6712  035564  000207                      RTS     PC
6713
6714                                      ;
6715                                      ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
6716                                      ;
6717  035566  006337  001402      MULT6:  ASL     SECTOR          ;2 X SECTOR
6718  035572  013746  001402              MOV     SECTOR,-(SP)
6719  035576  006337  001402              ASL     SECTOR          ;4 X SECTOR
6720  035602  062637  001402              ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
6721  035606  000207                      RTS     PC
6722
6723                                      ;
6724                                      ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
6725                                      ;CMD.  IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
6726                                      ;CYLINDER AT CYL 410, TRACK 2.
6727                                      ;
6728                                      ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
6729                                      ;RETURN+2 IF LISTED, SKIP OVER ERROR
6730                                      ;
6731  035610  010446              TRUERR: MOV     R4,-(SP)                ;SAVE R4
6732
6733  035612  032737  010000  007340      BIT     #CFMT,HCS1      ;CHECK FORMAT
6734  035620  001014                      BNE     2$              ;BR FOR 20 SECTOR FORMAT
6735
6736  035622  012704  003332              MOV     #BSE22H+8.,R4
6737  035626  004737  035710              JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
6738  035632  000422                      BR      3$              ;RETURN HERE IF YES
6739
6740  035634  012704  005332              MOV     #BSE22S+8.,R4   ;ELSE RETURN HERE
6741  035640  004737  035710              JSR     PC,TERR1        ;& SEE IF ON SOFTWARE DETECTED TABLE
6742  035644  000415                      BR      3$              ;RETURN HERE IF YES
6743
6744  035646  012604              1$:     MOV     (SP)+,R4        ;RESTORE R4
6745  035650  000207                      RTS     PC              ;RETURN WITHOUT JUMPING OVER ERROR
6746
```

```
 6747  035652  012704  002332          2$:     MOV     #BSE20H+8.,R4
 6748  035656  004737  035710                  JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
 6749  035662  000406                          BR      3$              ;RETURN HERE IF YES
 6750
 6751  035664  012704  004332                  MOV     #BSE20S+8.,R4   ;ELSE RETURN HERE
 6752  035670  004737  035710                  JSR     PC,TERR1        ;SEE IF ON SOFTWARE DETECTED TABLE
 6753  035674  000401                          BR      3$              ;RETURN HERE IF YES
 6754  035676  000763                          BR      1$              ;RETURN HERE IF NO
 6755
 6756  035700  012604                  3$:     MOV     (SP)+,R4        ;RESTORE R4
 6757  035702  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR ON RETURN
 6758  035706  000207                          RTS     PC
 6759
 6760                                  ;
 6761                                  ;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
 6762                                  ;THE BSE TABLE FOR THE ABOVE SUBROUTINE.
 6763                                  ;RETURN IF FOUND ON TABLE
 6764                                  ;RETURN+2 IF NOT FOUND
 6765                                  ;
 6766  035710  021427  177777          TERR1:  CMP     (R4),#-1        ;SEE IF ALL 1'S
 6767  035714  001405                          BEQ     1$              ;BR IF YES, NOT ON TABLE
 6768  035716  022437  007360                  CMP     (R4)+,HDC       ;SEE IF CYL MATCH
 6769  035722  001405                          BEQ     2$              ;BR IF YES
 6770  035724  005724                          TST     (R4)+           ;ELSE ADV TO NEXT CYL WORD
 6771  035726  000770                          BR      TERR1           ;& TRY AGAIN
 6772
 6773  035730  062716  000002          1$:     ADD     #2,(SP)
 6774  035734  000207                          RTS     PC
 6775
 6776  035736  022437  007350          2$:     CMP     (R4)+,HDA       ;SEE IF SECTOR & TRACK MATCH
 6777  035742  001401                          BEQ     3$              ;BR IF YES
 6778  035744  000761                          BR      TERR1           ;OR TRY AGAIN
 6779
 6780  035746  000207                  3$:     RTS     PC
 6781                                  ;
 6782                                  ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
 6783                                  ;
 6784  035750  005037  001372          CLKON:  CLR     TIMUP
 6785  035754  005737  007504                  TST     PCLKF
 6786  035760  001004                          BNE     1$              ;BRANCH IF P-CLOCK PRESENT
 6787  035762  012777  000100  143336          MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
 6788  035770  000207                          RTS     PC
 6789  035772  012777  177777  143322  1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
 6790  036000  012777  000135  143312          MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
 6791  036006  000207                          RTS     PC              ;LINE FREQ & RUN
 6792                                  ;
 6793                                  ;KW11-L & KW11-P INTERRUPT HANDLER
 6794                                  ;
 6795  036010  005037  001372          CLOCK:  CLR     TIMUP
 6796  036014  005337  001366                  DEC     COUNT
 6797  036020  001010                          BNE     1$
 6798  036022  013737  001364  001366          MOV     HZ,COUNT
 6799  036030  005337  001370                  DEC     SEC
 6800  036034  001002                          BNE     1$
 6801  036036  005237  001372                  INC     TIMUP           ;SORRY, TIME IS UP
 6802  036042  000002                  1$:     RTI
```

```
                                        ;
                                        ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
                                        ;
036044  005737  007504        CLKOF:  TST     PCLKF
036050  001003                        BNE     1$              ;BRACH IF P-CLOCK PRESENT
036052  005077  143250                CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
036056  000207                        RTS     PC
036060  005077  143234        1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
036064  000207                        RTS     PC

                                        ;
                                        ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MESSAGE
                                        ;ENTER WITH THE EXPECTED WORD IN TEMP1
                                        ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
                                        ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
                                        ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
                                        ; THE PARITY BIT IS NOT SET IN B
                                        ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
                                        ;SET IN TEMP1
                                        ;
036066  010046                SBPAR:  MOV     R0,-(SP)        ;SAVE R0
036070  010146                        MOV     R1,-(SP)        ;SAVE R1
036072  012700  000021                MOV     #17.,R0         ;SHIFT COUNTER
036076  005001                        CLR     R1              ;COUNT # OF 1'S IN TEMP1
036100  000241                        CLC                     ;CLEAR CARRY

036102  006137  007376        1$:     ROL     TEMP1
036106  103001                        BCC     2$              ;BR IF CARRY CLEAR
036110  005201                        INC     R1              ;COUNT # OF 1'S
036112  005300                2$:     DEC     R0              ;SHIFT COUNTER
036114  001372                        BNE     1$

036116  032701  000001                BIT     #BIT0,R1
036122  001003                        BNE     3$              ;BR IF ODD # IN R0
036124  052737  100000  007376        BIS     #M.PAR,TEMP1    ;SET PARITY BIT
036132  012601                3$:     MOV     (SP)+,R1        ;RESTORE R1
036134  012600                        MOV     (SP)+,R0        ;RESTORE R0
036136  000207                        RTS     PC

                                        ;
                                        ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                        ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                        ; IE: MY LOOP MACRO
                                        ;
036140  032777  001000  142772  SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
036146  001406                        BEQ     1$              ;BR IF NO
036150  105737  001103                TSTB    $ERFLG          ;HAD ERROR?
036154  001403                        BEQ     1$              ;BR IF NO
036156  013716  001110                MOV     $LPERR,(SP)
036162  000002                        RTI

036164  011637  001110        1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
036170  000002                        RTI
                                        ;
                                        ;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
                                        ;IT IS USED TO ALLOW THE OPERATOR TO HLAT THE CPU WHILE INSURING
```

```
                                        ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
                                        ;THE CPU.
                                        ;
036172  022626                  STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT

036174  004737  034140                  JSR     PC,SUBCLR
036200  104024                          ERROR   24              ;CERR AFTER

036202  005737  007322                  TST     UNLD            ;SEE IF HEADS UNLOADED
036206  001434                          BEQ     3$              ;BR IF NO
036210  005737  000042                  TST     42              ;SEE IF MANUAL OR AUTO MODE
036214  001403                          BEQ     1$              ;BR IF MANUAL MODE
036216  104401  045720                  TYPE    ,MSG74          ;PGM ABORT PENDING
036222  000402                          BR      2$
036224  104401  045766          1$:     TYPE    ,MSG75          ;HALT PENDING
036230                          2$:

036230  004737  034140                  JSR     PC,SUBCLR
036234  104024                          ERROR   24              ;CERR AFTER SCLR

036236  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5)  ;START SPINDLE CMD
036244  013737  001406  007376          MOV     T10,TEMP1       ;SET TIMEOUT
036252  004737  032230                  JSR     PC,FRDY         ;FIND RDY
036256  104121                          ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.

036260  013737  001414  007400          MOV     T100,TEMP2      ;SETUP TIMEOUT
036266  004737  032544                  JSR     PC,FATT1        ;FIND ATTN
036272  104074                          ERROR   74              ;NO ATTN AFTER ST SPIN CMD.

036274  005037  007322                  CLR     UNLD

036300  005737  007324          3$:     TST     BADHDR          ;SEE IF HEADERS VALID
036304  001466                          BEQ     4$              ;BR IF YES
036306  005237  007326                  INC     HPEND

036312  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
036320  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
036326  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                        ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                        ;IN RKMR2 & RKMR3 RESP.
036334  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
036342  004737  032230                  JSR     PC,FRDY         ;FIND RDY
036346  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD

036350  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
036356  004737  033566                  JSR     PC,GSTAT
036362  032737  020000  007366          BIT     #D.RTZ,HMR2
036370  001001                          BNE     64$
036372  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
036374  013737  001406  007400  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
036402  004737  032544                  JSR     PC,FATT1        ;FIND ATTN
036406  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD

036410  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
036416  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
6915  036424  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6916  036432  013737  001406  007376          MOV     T10,TEMP1       ;SETUP TIMEOUT
6917  036440  004737  032230                  JSR     PC,FRDY         ;FIND RDY
6918  036444  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6919  036446  004737  032512                  JSR     PC,TSTATN       ;TEST FOR ATTN
6920  036452  000401                          BR      65$
6921  036454  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6922  036456                          65$:
6923
6924
6925  036456  000137  031226                  JMP     FORM            ;WRITE VALID FORMATS
6926
6927  036462  005737  000042          4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
6928  036466  001410                          BEQ     5$              ;BR IF MANUAL MODE
6929  036470  104401  046023                  TYPE    ,MSG76          ;PGM ABORTED
6930  036474  005037  031576                  CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
6931  036500  005037  001176                  CLR     $ESCAPE
6932  036504  000137  031550                  JMP     $EOP1           ;ABORT PGM
6933
6934  036510  104401  046045          5$:     TYPE    ,MSG77          ;CPU HALTED
6935  036514  000000                          HALT
6936  036516  000137  013340                  JMP     ST5             ;START OVER IF CONTINUE PRESSED
6937
6938
6939
6940                                  .SBTTL  UNEXPECTED TIMEOUT HANDLER
6941
6942                                  ;
6943                                  ;THIS ROUTINE IS ENTERED IF THERE IS
6944                                  ;       A.  NON EXISTANT MEMORY (NO SSYN)
6945                                  ;       B.  BOUNDRY ERROR
6946                                  ;       C.  STACK OVERFLOW
6947                                  ;
6948
6949  036522  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
6950  036524  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE
6951  036526  032777  020000  142404          BIT     #SW13,@SWR      ;INHIBIT ERR TYPOUT?
6952  036534  001005                          BNE     1$              ;YES, DON'T TYPE
6953  036536  104401  046226                  TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
6954  036542  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
6955                                          ;;TYPE PC
6956  036544  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
6957  036546     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
6958  036547     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
6959  036550  032777  001000  142362  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
6960  036556  001403                          BEQ     2$              ;NO, BRANCH
6961  036560  022626                          CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
6962  036562  000177  142320                  JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
6963                                          ;THAT GAVE BAD TIMEOUT
6964  036566  032777  040000  142344  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
6965  036574  001401                          BEQ     3$              ;NO BRANCH
6966  036576  000002                          RTI                     ;YES
6967
6968  036600  000000                  3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
6969                                          ;AS INDICATED. YOU CAN LOOP ON
6970                                          ;ERROR, LOOP ON TEST OR INHIBIT
```

```
                                                                ;ERROR TYPEOUT BY SETTING THOSE
                                                                ;SWITCHES.
                                                                ;
036602  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
036604  000137  031550                  JMP     $EOP1           ;ABORT TESTS

                                .SBTTL  MEMORY CHECK ENABLE TRAP

036610  012737  036624  001176  MEMERR: MOV     #1$,$ESCAPE
036616  011637  001334                  MOV     (SP),TRAPPC     ;STORE PC
036622  104041                          ERROR   41              ;UNEXP MEM PARITY TRAP
036624  005037  001176          1$:     CLR     $ESCAPE
036630  032777  001000  142302          BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
036636  001001                          BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
036640  000002                          RTI                     ;ELSE RETURN

036642  012706  001100          2$:     MOV     #STACK,SP       ;INIT STACK
036646  000177  142236                  JMP     @$LPERR         ;LOOP ON ERROR

                                .SBTTL  RK06 INTERRUPT HANDLER

036652  000240                  INTER:  NOP
036654  000240                          NOP
036656  000240                          NOP
036660  011600                          MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
036662  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE.
036664  104401  044473                  TYPE    ,MSG6           ;INT AT PC=
036670  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE PC
036672  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
036674     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
036675     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
036676  000000                          HALT
036700  000240                          NOP
036702  000240                          NOP
036704  000002                          RTI

                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;POWER DOWN ROUTINE

036706  012737  036720  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
036714  000000                          HALT
036716  000776                          BR      .-2             ;HANG UP.

                                ;POWER UP ROUTINE

036720  005037  036772          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
036724  005237  036772          1$:     INC     $PWRCT          ;WAIT FOR THE INCR
036730  001375                          BNE     1$              ;OF WORD
036732  012737  036706  000024          MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
036740  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;PRIORITY 7
036746  012737  000340  000036          MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
036754  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
036760  104401  044663                  TYPE    ,MSG11          ;REPORT POWER FAIL
```

```
7027  036764  000005                  RESET
7028  036766  000137  015122          JMP     PFSRT
7029
7030  036772  000000                  $PWRCT: 0                       ;WAIT COUNT FOR TTY
7031
7032                                  ;
7033                                  ;DIVISION UTILITY ROUTINE
7034                                  ;
7035                                  ;R0-R1-R2-R3=DIVIDEND
7036                                  ;R4-R5=DIVISOR
7037                                  ;R0-R1=REMAINDER AFTER DIVISION
7038                                  ;R2-R3=QUOTIENT AFTER DIVISION
7039                                  ;ENTER WITH JSR PC,M.DPID
7040                                  ;
7041  036774  012746  000040          M.DPID: MOV     #40,-(SP)       ;COUNTER FOR DIVISION CYCLES
7042  037000  010446                          MOV     R4,-(SP)        ;HI ORDER
7043  037002  010546                          MOV     R5,-(SP)        ;LO ORDER TO THE STACK
7044  037004  005466  000002                  NEG     2(SP)           ;FORM NEGATIVE
7045  037010  005416                          NEG     @SP             ;VERSION OF DIVISOR
7046  037012  005666  000002                  SBC     2(SP)
7047  037016  061601                          ADD     @SP,R1
7048  037020  005500                          ADC     R0              ;PERFORM INIT SUBT.
7049  037022  066600  000002                  ADD     2(SP),R0
7050  037026  103445                          BCS     M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
7051  037030  005046                          CLR     -(SP)           ;THIS IS A LONGER LASTING CARRY BIT
7052  037032  006103                  M.DP40: ROL     R3
7053  037034  006102                          ROL     R2
7054  037036  006101                          ROL     R1
7055  037040  006100                          ROL     R0
7056  037042  005716                          TST     @SP             ;TEST CARRY INDICATOR
7057  037044  001410                          BEQ     M.DP41          ;IF TO CARRY THEN ADD, ELSE SUBT.
7058  037046  005016                          CLR     @SP             ;CLEAR UP FOR NEXT TIME
7059  037050  066601  000002                  ADD     2(SP),R1
7060  037054  005500                          ADC     R0              ;ADD -(DIVISOR)
7061  037056  005516                          ADC     @SP             ;SET CARRY
7062  037060  066600  000004                  ADD     4(SP),R0
7063  037064  000404                          BR      M.DP42
7064
7065  037066  060501                  M.DP41: ADD     R5,R1
7066  037070  005500                          ADC     R0              ;ADD +(DIVISOR)
7067  037072  005516                          ADC     @SP             ;SET CARRY
7068  037074  060400                          ADD     R4,R0
7069  037076  005516                  M.DP42: ADC     @SP             ;SET CARRY
7070  037100  005716                          TST     @SP             ;TEST THE UPDATE INDICATOR
7071  037102  001401                          BEQ     .+4             ;IF 0,FORGET IT
7072  037104  005203                          INC     R3              ;NO CARRY POSSIBLE HERE
7073  037106  005366  000006                  DEC     6(SP)           ;DECREMENT CTR
7074  037112  003347                          BGT     M.DP40          ;BR IF MORE TO DO
7075  037114  006003                          ROR     R3
7076  037116  103404                          BCS     M.DP44
7077  037120  060501                          ADD     R5,R1
7078  037122  005500                          ADC     R0
7079  037124  060400                          ADD     R4,R0
7080  037126  000241                          CLC
7081
7082  037130  006103                  M.DP44: ROL     R3
```

```
 7083  037132  062706  000010          ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
 7084  037136  000242                  CLV
 7085  037140  000207                  RTS     PC
 7086
 7087  037142  062706  000006  M.DP50: ADD     #6,SP
 7088  037146  000262                  SEV
 7089  037150  000207                  RTS     PC
 7090
```

```
                                .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

037152                          $SCOPE:
037152  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
037154  032777  040000  141756  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
037162  001114                         BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
037164  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
037166  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
037172  012737  037212  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
037200  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
037204  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
037210  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
037212  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
037214  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
037220  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
037222                          6$:;#####END OF CODE FOR THE XOR TESTER#####
037222  032777  000400  141710          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
037230  001404                         BEQ     2$              ;;BR IF NO
037232  127737  141702  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
037240  001465                         BEQ     $OVER           ;;BR IF YES
037242  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
037246  001421                         BEQ     3$              ;;BR IF NO
037250  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
037256  101015                         BHI     3$              ;;BR IF NO
037260  032777  001000  141652          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
037266  001404                         BEQ     4$              ;;BR IF NO
037270  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
037276  000446                         BR      $OVER
037300  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
037304  005037  001174                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
037310  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
037312  032777  004000  141620  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
037320  001011                         BNE     1$              ;;BR IF YES
037322  005737  001216                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
037326  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
037330  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
037334  023737  001174  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
037342  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
037344  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
037352  013737  037430  001174          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
037360  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
037364  113737  001102  001214          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 7147  037372  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 7148  037376  011637  001110                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 7149  037402  005037  001176                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 7150  037406  112737  000001  001115         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 7151  037414  013777  001102  141520 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 7152  037422  013716  001106                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 7153  037426  000002                         RTI                     ;;FIXES PS
 7154  037430  003720                 $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 7155                                  .SBTTL  ERROR HANDLER ROUTINE
 7156
 7157                                  ;;**************************************************************
 7158                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 7159                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 7160                                  ;*AND GO TO TYPERR ON ERROR
 7161                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 7162                                  ;*SW15=1        HALT ON ERROR
 7163                                  ;*SW13=1        INHIBIT ERROR TYPEOUTS
 7164                                  ;*SW10=1        BELL ON ERROR
 7165                                  ;*SW09=1        LOOP ON ERROR
 7166                                  ;*CALL
 7167                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 7168
 7169  037432                         $ERROR:
 7170  037432  104407                         CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 7171  037434  105237  001103         7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 7172  037440  001775                         BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 7173  037442  013777  001102  141472         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 7174  037450  032777  002000  141462         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 7175  037456  001402                         BEQ     1$              ;;NO - SKIP
 7176  037460  104401  001200                 TYPE    ,$BELL          ;;RING BELL
 7177  037464  005237  001112         1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 7178  037470  011637  001116                 MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 7179  037474  162737  000002  001116         SUB     #2,$ERRPC
 7180  037502  117737  141410  001114         MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 7181  037510  032777  020000  141422         BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 7182  037516  001004                         BNE     20$             ;;SKIP TYPEOUTS
 7183  037520  004737  055062                 JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 7184  037524  104401  001205                 TYPE    ,$CRLF
 7185  037530                         20$:
 7186  037530  122737  000001  001230         CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 7187  037536  001007                         BNE     2$              ;;NO,SKIP APT ERROR REPORT
 7188  037540  113737  001114  037552         MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 7189  037546  004737  040356                 JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 7190  037552     000         21$:    .BYTE   0
 7191  037553     000                 .BYTE   0
 7192  037554  000777         22$:    BR      22$             ;;APT ERROR LOOP
 7193  037556  005777  141356  2$:     TST     @SWR            ;;HALT ON ERROR
 7194  037562  100002                 BPL     3$              ;;SKIP IF CONTINUE
 7195  037564  000000                 HALT                    ;;HALT ON ERROR!
 7196  037566  104407                 CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 7197  037570  032777  001000  141342 3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 7198  037576  001402                 BEQ     4$              ;;BR IF NO
 7199  037600  013716  001110                 MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 7200  037604  005737  001176         4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 7201  037610  001402                 BEQ     5$              ;;BR IF NONE
 7202  037612  013716  001176                 MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
7203  037616                                5$:
7204  037616  022737  031636  000042              CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
7205  037624  001001                              BNE     6$              ;;BRANCH IF NO
7206  037626  000000                              HALT                    ;;YES
7207  037630                                6$:
7208  037630  000002                              RTI                     ;;RETURN
7209                                .SBTTL  TYPE ROUTINE
7210
7211                                ;;*****************************************************************
7212                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
7213                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
7214                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
7215                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
7216                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
7217                                ;*
7218                                ;*CALL:
7219                                ;*1) USING A TRAP INSTRUCTION
7220                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
7221                                ;*OR
7222                                ;*      TYPE
7223                                ;*      MESADR
7224                                ;*
7225
7226  037632  105737  001157       $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
7227  037636  100002                        BPL     1$              ;;BR IF YES
7228  037640  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
7229  037642  000430                        BR      3$              ;;LEAVE
7230  037644  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
7231  037646  017600  000002              MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
7232  037652  122737  000001  001230        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
7233  037660  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
7234  037662  132737  000100  001231        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
7235  037670  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
7236  037672  010037  037702                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
7237  037676  004737  040346                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
7238  037702  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
7239  037704  132737  000040  001231  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
7240  037712  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
7241  037714  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
7242  037716  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
7243  037720  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
7244  037722  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
7245  037724  062716  000002          3$:   ADD     #2,(SP)         ;;ADJUST RETURN PC
7246  037730  000002                        RTI                     ;;RETURN
7247  037732  122716  000011          4$:   CMPB    #HT,(SP)        ;;BRANCH IF <HT>
7248  037736  001430                        BEQ     8$
7249  037740  122716  000200              CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
7250  037744  001006                        BNE     5$
7251  037746  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
7252  037750  104401                        TYPE                    ;;TYPE A CR AND LF
7253  037752  001205                        $CRLF
7254  037754  105037  040110              CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
7255  037760  000755                        BR      2$              ;;GET NEXT CHARACTER
7256  037762  004737  040044          5$:   JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
7257  037766  123726  001156          6$:   CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
7258  037772  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
7259  037774  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
7260                                                                    ;;AND THE NULL CHAR.
7261  040000  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
7262  040004  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
7263  040006  004737  040044                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
7264  040012  105337  040110                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
7265  040016  000770                          BR      7$              ;;LOOP
7266
7267                                  ;HORIZONTAL TAB PROCESSOR
7268
7269  040020  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
7270  040024  004737  040044          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
7271  040030  132737  000007  040110          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
7272  040036  001372                          BNE     9$              ;;TAB STOP
7273  040040  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
7274  040042  000724                          BR      2$              ;;GET NEXT CHARACTER
7275  040044  105777  141100          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
7276  040050  100375                          BPL     $TYPEC
7277  040052  116677  000002  141072          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
7278  040060  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
7279  040066  001003                          BNE     1$              ;;BRANCH IF NO
7280  040070  105037  040110                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
7281  040074  000406                          BR      $TYPEX          ;;EXIT
7282  040076  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
7283  040104  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
7284  040106  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
7285  040110  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
7286  040112  000207          $TYPEX: RTS     PC
7287
7288                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7289
7290                                  ;;******************************************************************
7291                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7292                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7293                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7294                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
7295                                  ;*REPLACED WITH SPACES.
7296                                  ;*CALL:
7297                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
7298                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
7299
7300  040114                          $TYPDS:
7301  040114  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7302  040116  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7303  040120  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7304  040122  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
7305  040124  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
7306  040126  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
7307  040132  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
7308  040136  100004                          BPL     1$              ;;BR IF INPUT IS POS.
7309  040140  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
7310  040142  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
7311  040150  005000          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
7312  040152  012703  040330                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
7313  040156  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
7314  040162  005002          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
```

```
7315  040164  016001  040320          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
7316  040170  160105                  SUB     R1,R5           ;;FORM THIS BCD DIGIT
7317  040172  002402                  BLT     4$              ;;BR IF DONE
7318  040174  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
7319  040176  000774                  BR      3$
7320  040200  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
7321  040202  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
7322  040204  001002                  BNE     5$              ;;FALL THROUGH IF 0
7323  040206  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
7324  040210  100407                  BMI     7$              ;;BR IF YES
7325  040212  106316          5$:     ASLB    (SP)            ;;MSD?
7326  040214  103003                  BCC     6$              ;;BR IF NO
7327  040216  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
7328  040224  052702  000060  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
7329  040230  052702  000040  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
7330  040234  110223                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
7331  040236  005720                  TST     (R0)+           ;;JUST INCREMENTING
7332  040240  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
7333  040244  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
7334  040246  003002                  BGT     8$              ;;GO TO EXIT
7335  040250  010502                  MOV     R5,R2           ;;GET THE LSD
7336  040252  000764                  BR      6$              ;;GO CHANGE TO ASCII
7337  040254  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
7338  040256  100003                  BPL     9$              ;;BR IF NO
7339  040260  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
7340  040266  105013          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
7341  040270  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
7342  040272  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
7343  040274  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
7344  040276  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
7345  040300  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
7346  040302  104401  040330          TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
7347  040306  016666  000002  000004  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
7348  040314  012616                  MOV     (SP)+,(SP)
7349  040316  000002                  RTI                     ;;RETURN TO USER
7350  040320  023420          $DTBL:  10000.
7351  040322  001750                  1000.
7352  040324  000144                  100.
7353  040326  000012                  10.
7354  040330  000004          $DBLK:  .BLKW   4
7355                           .SBTTL  APT COMMUNICATIONS ROUTINE
7356
7357                           ;;**********************************************************************
7358  040340  112737  000001  040604  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
7359  040346  112737  000001  040602  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
7360  040354  000403                  BR      $ATYC
7361  040356  112737  000001  040604  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
7362  040364                  $ATYC:
7363  040364  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7364  040366  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7365  040370  105737  040602          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
7366  040374  001450                  BEQ     5$              ;;IF NOT:  BR
7367  040376  122737  000001  001230  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
7368  040404  001031                  BNE     3$              ;;IF NOT:  BR
7369  040406  132737  000100  001231  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
7370  040414  001425                  BEQ     3$              ;;IF NOT:  BR
```

```
7371  040416  017600  000004            MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
7372  040422  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
7373  040430  005737  001210        1$: TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
7374  040434  001375                    BNE     1$              ;;IF NOT:  WAIT
7375  040436  010037  001224            MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
7376  040442  105720                2$: TSTB    (R0)+           ;;FIND END OF MESSAGE
7377  040444  00[illegible]              BNE     2$
7378  040446  16[illegible]  001224      SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
7379  040452  006200                    ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
7380  040454  010037  001226            MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
7381  040460  012737  000004  001210    MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
7382  040466  000413                    BR      5$
7383  040470  017637  000004  040514 3$: MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
7384  040476  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
7385  040504  013746  177776            MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
7386  040510  004737  037632            JSR     PC,$TYPE        ;;CALL TYPE MACRO
7387  040514  000000              4$:   .WORD   0
7388  040516                      5$:
7389  040516  105737  040604       10$: TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
7390  040522  001416                    BEQ     12$             ;;IF NOT:  BR
7391  040524  005737  001230            TST     $ENV            ;;RUNNING UNDER APT?
7392  040530  001413                    BEQ     12$             ;;IF NOT:  BR
7393  040532  005737  001210       11$: TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
7394  040536  001375                    BNE     11$             ;;IF NOT:  WAIT
7395  040540  017637  000004  001212    MOV     @4(SP),$FATAL   ;;GET ERROR #
7396  040546  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
7397  040554  005237  001210            INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
7398  040560  105037  040604       12$: CLRB    $FFLG           ;;CLEAR FATAL FLAG
7399  040564  105037  040603            CLRB    $LFLG           ;;CLEAR LOG FLAG
7400  040570  105037  040602            CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
7401  040574  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
7402  040576  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
7403  040600  000207                    RTS     PC              ;;RETURN
7404  040602     000          $MFLG:   .BYTE   0               ;;MESSG. FLAG
7405  040603     000          $LFLG:   .BYTE   0               ;;LOG FLAG
7406  040604     000          $FFLG:   .BYTE   0               ;;FATAL FLAG
7407          040606                   .EVEN
7408          000200          APTSIZE=200
7409          000001          APTENV=001
7410          000100          APTSPOOL=100
7411          000040          APTCSUP=040
7412                          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
7413
7414                          ;;**************************************************************
7415                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7416                          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
7417                          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7418                          ;*CALL:
7419                          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7420                          ;*      TYPOS                   ;;CALL FOR TYPEOUT
7421                          ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7422                          ;*      .BYTE   M               ;;M=1 OR 0
7423                          ;*                                      ;;1=TYPE LEADING ZEROS
7424                          ;*                                      ;;0=SUPPRESS LEADING ZEROS
7425                          ;*
7426                          ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
                                        
040606  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
040612  116637  000001  041031          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
040620  112637  041033                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
040624  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
040630  000406                          BR      $TYPON
040632  112737  000001  041031  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
040640  112737  000006  041033          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
040646  112737  000005  041030  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
040654  010346                          MOV     R3,-(SP)        ;;SAVE R3
040656  010446                          MOV     R4,-(SP)        ;;SAVE R4
040660  010546                          MOV     R5,-(SP)        ;;SAVE R5
040662  113704  041033                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
040666  005404                          NEG     R4
040670  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
040674  110437  041032                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
040700  113704  041031                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
040704  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
040710  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
040712  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
040714  000404                          BR      3$              ;;GO DO MSB
040716  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
040720  006105                          ROL     R5
040722  006105                          ROL     R5
040724  010503                          MOV     R5,R3
040726  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
040730  105337  041032                  DECB    $OMODE          ;;TYPE THIS DIGIT?
040734  100016                          BPL     7$              ;;BR IF NO
040736  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
040742  001002                          BNE     4$              ;;TEST FOR 0
040744  005704                          TST     R4              ;;SUPPRESS THIS 0?
040746  001403                          BEQ     5$              ;;BR IF YES
040750  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
040752  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
040756  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
040762  110337  041026                  MOVB    R3,8$           ;;SAVE FOR TYPING
040766  104401  041026                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
040772  105337  041030          7$:     DECB    $OCNT           ;;COUNT BY 1
040776  003347                          BGT     2$              ;;BR IF MORE TO DO
041000  002402                          BLT     6$              ;;BR IF DONE
041002  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
041004  000744                          BR      2$              ;;GO DO THE LAST DIGIT
041006  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
041010  012604                          MOV     (SP)+,R4        ;;RESTORE R4
041012  012603                          MOV     (SP)+,R3        ;;RESTORE R3
041014  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
041022  012616                          MOV     (SP)+,(SP)
```

```
 7483  041024  000002                        RTI                     ;;RETURN
 7484  041026     000  8$:      .BYTE   0      ;;STORAGE FOR ASCII DIGIT
 7485  041027     000           .BYTE   0      ;;TERMINATOR FOR TYPE ROUTINE
 7486  041030     000  $OCNT:   .BYTE   0      ;;OCTAL DIGIT COUNTER
 7487  041031     000  $OFILL:  .BYTE   0      ;;ZERO FILL SWITCH
 7488  041032  000000  $OMODE:  .WORD   0      ;;NUMBER OF DIGITS TO TYPE
 7489                  .SBTTL  TTY INPUT ROUTINE
 7490
 7491                  ;;*****************************************************************
 7492                  .ENABL  LSB
 7493  041034  000000  $TKCNT:  .WORD   0      ;;NUMBER OF ITEMS IN QUEUE
 7494  041036  000000  $TKQIN:  .WORD   0      ;;INPUT POINTER
 7495  041040  000000  $TKQOUT: .WORD   0      ;;OUTPUT POINTER
 7496  041042  000001  $TKQSRT: .BLKB   1      ;;TTY KEYBOARD QUEUE
 7497          041043  $TKQEND=.
 7498          041044  .EVEN
 7499
 7500                  ;*TK INITIALIZE ROUTINE
 7501                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
 7502                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
 7503                  ;
 7504                  ;*CALL:
 7505                  ;*      JSR     PC,$TKINT
 7506                  ;*      RETURN
 7507                  ;
 7508  041044  005037  041034  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
 7509  041050  012737  041042  041036  MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
 7510  041056  013737  041036  041040  MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
 7511  041064  012737  041114  000060  MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
 7512  041072  012737  000200  000062  MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
 7513  041100  005777  140042          TST     @$TKB           ;;CLEAR DONE FLAG
 7514  041104  012777  000100  140032  MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
 7515  041112  000207                  RTS     PC              ;;RETURN TO CALLER
 7516
 7517                  ;*TK SERVICE ROUTINE
 7518                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
 7519                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
 7520                  ;*IT IN THE QUEUE.
 7521                  ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
 7522                  ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
 7523                  ;
 7524  041114  117746  140026  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
 7525  041120  042716  177600          BIC     #↑C177,(SP)     ;;STRIP THE JUNK
 7526  041124  021627  000003          CMP     (SP),#3         ;;IS IT A CONTROL C?
 7527  041130  001007                  BNE     1$              ;;BRANCH IF NO
 7528  041132  104401  042242          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
 7529  041136  004737  041044          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
 7530  041142  005726                  TST     (SP)+           ;;CLEAN UP STACK
 7531  041144  000137  036172          JMP     STOP            ;;CONTROL C RESTART
 7532  041150  021627  000007  1$:      CMP     (SP),#7         ;;IS IT A CONTROL G?
 7533  041154  001004                  BNE     2$              ;;BRANCH IF NO
 7534  041156  022737  000176  001140  CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
 7535  041164  001500                  BEQ     6$              ;;GO TO SWR CHANGE
 7536
 7537  041166                  2$:
 7538  041166  022737  000001  041034  CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
```

```
7539  041174  001004                    BNE     3$              ;;BRANCH IF NO
7540  041176  104401  001200            TYPE    ,$BELL          ;;RING THE TTY BELL
7541  041202  005726                    TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
7542  041204  000451                    BR      5$              ;;EXIT
7543  041206  021627  000023    3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
7544  041212  001021                    BNE     32$             ;;BRANCH IF NO
7545  041214  005077  137724            CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
7546  041220  005726                    TST     (SP)+           ;;CLEAN CHAR OFF STACK
7547  041222  105777  137716    31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
7548  041226  100375                    BPL     31$             ;;LOOP UNTIL ITS THERE
7549  041230  117746  137712            MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
7550  041234  042716  177600            BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
7551  041240  022627  000021            CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
7552  041244  001366                    BNE     31$             ;;BRANCH IF NO
7553  041246  012777  000100  137670    MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
7554  041254  000002                    RTI                     ;;RETURN
7555  041256  005237  041034    32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
7556  041262  021627  000140            CMP     (SP),#140       ;;IS IT UPPER CASE?
7557  041266  002405                    BLT     4$              ;;BRANCH IF YES
7558  041270  021627  000175            CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
7559  041274  003002                    BGT     4$              ;;BRANCH IF YES
7560  041276  042716  000040            BIC     #40,(SP)        ;;MAKE IT UPPER CASE
7561  041302  112677  177530    4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
7562  041306  005237  041036            INC     $TKQIN          ;;UPDATE THE POINTER
7563  041312  023727  041036  041043    CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
7564  041320  001003                    BNE     5$              ;;BRANCH IF NO
7565  041322  012737  041042  041036    MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
7566  041330  000002            5$:     RTI                     ;;RETURN
7567
7568                            ;;**************************************************************
7569                            ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
7570                            ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
7571                            ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
7572                            ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
7573  041332  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
7574  041340  001124                    BNE     15$             ;;EXIT IF NOT
7575  041342  105777  137576            TSTB    @$TKS           ;;IS A CHAR WAITING?
7576  041346  100121                    BPL     15$             ;;IF NOT, EXIT
7577  041350  117746  137572            MOVB    @$TKB,-(SP)     ;;YES
7578  041354  042716  177600            BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
7579  041360  021627  000007            CMP     (SP),#7         ;;IS IT A CONTROL-G?
7580  041364  001300                    BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
7581                                                            ;;AND EXIT
7582
7583                            ;;**************************************************************
7584                            ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
7585                            ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
7586                            ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
7587  041366  123727  001134  000001 6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
7588  041374  001674                    BEQ     2$              ;;BRANCH IF YES
7589  041376  005726                    TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
7590  041400  004737  041044            JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
7591  041404  005077  137534            CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
7592  041410  112737  000001  001135    MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
7593
7594  041416  104401  042254            TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
```

```
7595  041422  104401  042261          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
7596  041426  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
7597  041432  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7598  041434  104401  042272                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
7599  041440  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
7600  041442  005046                          CLR     -(SP)           ;;THE NEW SWR
7601  041444  105777  137474          7$:     TSTB    @$TKS           ;;CHAR THERE?
7602  041450  100375                          BPL     7$              ;;IF NOT TRY AGAIN
7603
7604  041452  117746  137470                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
7605  041456  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
7606
7607  041462  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
7608  041466  001015                          BNE     9$              ;;BRANCH IF NOT
7609  041470  104401  042242                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
7610  041474  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
7611  041500  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
7612  041506  001003                          BNE     8$              ;;BRANCH IF NO
7613  041510  012777  000100  137426          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
7614  041516  000137  036172          8$:     JMP     STOP            ;;CONTROL-C RESTART
7615
7616
7617  041522  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
7618  041526  001005                          BNE     10$             ;;BRANCH IF NOT
7619  041530  104401  042247                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
7620  041534  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
7621  041540  000737                          BR      19$             ;;LET'S TRY IT AGAIN
7622
7623
7624  041542  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
7625  041546  001022                          BNE     16$             ;;BRANCH IF NO
7626  041550  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
7627  041554  001403                          BEQ     11$             ;;BRANCH IF YES
7628  041556  016677  000002  137354          MOV     2(SP),@SWR      ;;SAVE NEW SWR
7629  041564  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
7630  041570  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
7631  041574  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
7632  041602  001003                          BNE     15$             ;;BRANCH IF NOT
7633  041604  012777  000100  137332          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
7634  041612  000002                  15$:    RTI                     ;;RETURN
7635  041614  004737  040044          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
7636  041620  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
7637  041624  002420                          BLT     18$             ;;BRANCH IF YES
7638  041626  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
7639  041632  003015                          BGT     18$             ;;BRANCH IF YES
7640  041634  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
7641  041640  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
7642  041644  001403                          BEQ     17$             ;;BRANCH IF YES
7643  041646  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
7644  041650  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
7645  041652  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
7646  041654  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
7647  041660  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
7648  041664  000667                          BR      7$              ;;GET THE NEXT ONE
7649  041666  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
7650  041672  000720                          BR      20$             ;;SIMULATE CONTROL-U
```

```
                                .DSABL  LSB


                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
041674  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
041676  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
041704  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
041710  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
041712  012746  041720                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
041716  000002                          RTI                     ;;POP NEW PC AND PS
041720                          64$:
041720  005737  041034          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
041724  001775                          BEQ     1$
041726  005337  041034                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
041732  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
041740  005237  041040                  INC     $TKQOUT         ;;UPDATE THE POINTER
041744  023727  041040  041043          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
041752  001003                          BNE     2$              ;;BRANCH IF NO
041754  012737  041042  041040          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
041762  000002                  2$:     RTI                     ;;RETURN
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

041764  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
041766  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
041770  012703  042220          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
041774  022703  042242          2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
042000  101456                          BLOS    4$              ;;BR IF YES
042002  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
042004  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
042006  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
042012  001022                          BNE     5$              ;;BR IF NO
042014  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
042016  001007                          BNE     6$              ;;BR IF NO
042020  112737  000134  042216          MOVB    #'\,9$          ;;TYPE A BACK SLASH
042026  104401  042216                  TYPE    ,9$
042032  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
042036  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
042040  020327  042220                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
042044  103434                          BLO     4$              ;;BR IF YES
042046  111337  042216                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
042052  104401  042216                  TYPE    ,9$             ;;GO TYPE
042056  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
042060  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
042062  001406                          BEQ     7$              ;;BR IF NO
```

```
7707  042064  112737  000134  042216          MOVB    #'\,9$          ;;TYPE A BACK SLASH
7708  042072  104401  042216                  TYPE    9$
7709  042076  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
7710  042100  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
7711  042104  001003                          BNE     8$              ;;BR IF NO
7712  042106  104401  042247                  TYPE    $CNTLU          ;;TYPE A CONTROL "U"
7713  042112  000726                          BR      1$              ;;GO START OVER
7714  042114  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
7715  042120  001011                          BNE     3$              ;;BRANCH IF NO
7716  042122  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
7717  042124  104401  001205                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
7718  042130  104401  042220                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
7719  042134  000717                          BR      2$              ;;GO PICKUP ANOTHER CHARACTER
7720  042136  104401  001204          4$:     TYPE    $QUES           ;;TYPE A '?'
7721  042142  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
7722  042144  111337  042216          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
7723  042150  104401  042216                  TYPE    9$
7724  042154  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
7725  042160  001305                          BNE     2$              ;;LOOP IF NOT RETURN
7726  042162  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
7727  042166  104401  001206                  TYPE    $LF             ;;TYPE A LINE FEED
7728  042172  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
7729  042174  012603                          MOV     (SP)+,R3        ;;RESTORE R3
7730  042176  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
7731  042200  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
7732  042206  012766  042220  000004          MOV     #$TTYIN,4(SP)
7733  042214  000002                          RTI                     ;;RETURN
7734  042216     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
7735  042217     000                  .BYTE   0                       ;;TERMINATOR
7736  042220  000022          $TTYIN: .BLKB   22                      ;;RESERVE 22 BYTES FOR TTY INPUT
7737  042242  041536  005015     000  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
7738  042247     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
7739  042254  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
7740  042261     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
7741  042266  036440  000040
7742  042272  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
7743  042300  020075     000
7744          042304          .EVEN
7745                          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
7746
7747                  ;;******************************************************************
7748                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
7749                  ;*CHANGE IT TO BINARY.
7750                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
7751                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
7752                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
7753                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
7754                  ;*CALL:
7755                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
7756                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
7757                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
7758
7759  042304  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
7760  042306  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
7761  042314  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7762  042316  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
7763  042320  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7764  042322  104411         1$:     RDLIN                   ;;READ AN ASCIZ LINE
7765  042324  012600                 MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
7766  042326  010037  042432         MOV     R0,5$           ;;AND SAVE IT
7767  042332  005001                 CLR     R1              ;;CLEAR DATA WORD
7768  042334  005002                 CLR     R2
7769  042336  112046         2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
7770  042340  001420                 BEQ     3$              ;;IF ZERO GET OUT
7771  042342  122716  000060         CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
7772  042346  003026                 BGT     4$              ;;IS AN OCTAL DIGIT
7773  042350  122716  000067         CMPB    #'7,(SP)
7774  042354  002423                 BLT     4$
7775  042356  006301                 ASL     R1              ;;*2
7776  042360  006102                 ROL     R2
7777  042362  006301                 ASL     R1              ;;*4
7778  042364  006102                 ROL     R2
7779  042366  006301                 ASL     R1              ;;*8
7780  042370  006102                 ROL     R2
7781  042372  042716  177770         BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
7782  042376  062601                 ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
7783  042400  000756                 BR      2$              ;;LOOP
7784  042402  005726         3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
7785  042404  010166  000012         MOV     R1,12(SP)       ;;SAVE THE RESULT
7786  042410  010237  042442         MOV     R2,$HIOCT
7787  042414  012602                 MOV     (SP)+,R2        ;;POP STACK INTO R2
7788  042416  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
7789  042420  012600                 MOV     (SP)+,R0        ;;POP STACK INTO R0
7790  042422  000002                 RTI                     ;;RETURN
7791  042424  005726         4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
7792  042426  105010                 CLRB    (R0)            ;;SET A TERMINATOR
7793  042430  104401                 TYPE                    ;;TYPE UP THRU THE BAD CHAR.
7794  042432  000000         5$:     .WORD   0
7795  042434  104401  001204         TYPE    ,$QUES          ;;"?" "CR" & "LF"
7796  042440  000730                 BR      1$              ;;TRY AGAIN
7797  042442  000000         $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
7798                         .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
7799
7800                         ;;*****************************************************************
7801                         ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
7802                         ;*UNSIGNED OCTAL ASCIZ NUMBER.
7803                         ;*CALL
7804                         ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
7805                         ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
7806                         ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
7807
7808
7809  042444  104413         $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
7810  042446  016601  000002         MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
7811  042452  012705  042563         MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
7812  042456  012704  000014         MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
7813  042462  012703  177770         MOV     #^C7,R3         ;;MASK
7814  042466  012100                 MOV     (R1)+,R0        ;;LOWER WORD
7815  042470  012101                 MOV     (R1)+,R1        ;;HIGH WORD
7816  042472  005002                 CLR     R2              ;;TERMINATOR
7817  042474  110245         1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
7818  042476  010002                 MOV     R0,R2           ;;GET THIS DIGIT
```

```
7819  042500  005304                          DEC     R4              ;;COUNT THIS CHARACTER
7820  042502  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
7821  042504  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
7822  042506  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
7823  042510  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
7824  042514  104414                          RESREG                  ;;RESTORE ALL REGISTERS
7825  042516  000207                          RTS     PC              ;;RETURN TO USER
7826  042520  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
7827  042522  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
7828  042524  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
7829  042526  006001                          ROR     R1
7830  042530  006000                          ROR     R0
7831  042532  006001                          ROR     R1
7832  042534  006000                          ROR     R0
7833  042536  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
7834  042540  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
7835  042544  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
7836  042546  000016                  $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
7837                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
7838
7839                                  ;;**********************************************************************
7840                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
7841                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
7842                                  ;*POSITIVE.
7843                                  ;*CALL
7844                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
7845                                  ;*      JSR     PC,@#$DB2D
7846                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
7847                                                                  ;;IS ON THE STACK
7848
7849
7850  042564  104413                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
7851  042566  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
7852  042572  012700  042744                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
7853  042576  010066  000002                  MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
7854  042602  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
7855  042604  012202                          MOV     (R2)+,R2
7856  042606  012737  000012  042662          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
7857  042614  012704  042674                  MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
7858  042620  012705  042676                  MOV     #$TNPWR+2,R5
7859  042624  005003                  1$:     CLR     R3              ;;CLEAR PARTIAL
7860  042626  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
7861  042630  005602                          SBC     R2
7862  042632  161502                          SUB     (R5),R2
7863  042634  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
7864  042636  005203                          INC     R3              ;;ADD 1 TO PARTIAL
7865  042640  000772                          BR      2$              ;;LOOP
7866  042642  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
7867  042644  005502                          ADC     R2
7868  042646  062402                          ADD     (R4)+,R2
7869  042650  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
7870  042652  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
7871  042656  110320                          MOVB    R3,(R0)+        ;;SAVE IT
7872  042660  005327                          DEC     (PC)+           ;;DONE?
7873  042662  000000                  4$:     .WORD   0
7874  042664  001357                          BNE     1$              ;;BR IF NO
```

```
                                        CLRB    (R0)+           ;;TERMINATOR
                                        RESREG                  ;;RESTORE REGISTERS
                                        RTS     PC              ;;RETURN
                                $TNPWR: 145000                  ;;1.0E09
                                        35632
                                        160400                  ;;1.0E08
                                        2765
                                        113200                  ;;1.0E07
                                        230
                                        041100                  ;;1.0E06
                                        17
                                        103240                  ;;1.0E05
                                        1
                                        23420                   ;;1.0E04
                                        0
                                        1750                    ;;1.0E03
                                        0
                                        144                     ;;1.0E02
                                        0
                                        12                      ;;1.0E01
                                        0
                                        1                       ;;1.0E00
                                        0
                                $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
                                .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE

                                ;;******************************************************************
                                ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
                                ;*UNSIGNED DECIMAL ASCIZ NUMBER.
                                ;*CALL
                                ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
                                ;*      JSR     PC,@#$SB2D      ;;CALL
                                ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK

 042760  016637  000002  043010  $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
 042766  012746  043010          MOV     #1$,-(SP)       ;;SET POINTER
 042772  004737  042564          JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
 042776  062716  000005          ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
 043002  012666  000002          MOV     (SP)+,2(SP)     ;;PICKUP POINTER
 043006  000207                  RTS     PC              ;;RETURN
 043010  000000  000000  1$:     .WORD   0,0
                                .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS

                                ;;******************************************************************
                                ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
                                ;*LEADING NUMBERS.
                                ;*CALL
                                ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
                                ;*      JSR     PC,@#$SUPRS

 043014  010046          $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
 043016  016600  000004          MOV     4(SP),R0        ;;PICKUP THE POINTER
 043022  105710          1$:     TSTB    (R0)            ;;TERMINATEOR?
 043024  001403                  BEQ     2$              ;;BR IF YES
```

```
7931  043026  122720  000060          CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
7932  043032  001773                  BEQ     1$              ;;BR IF YES
7933  043034  005300                  DEC     R0              ;;BACKUP BY "1"
7934  043036  010037  043044          MOV     R0,3$           ;;SAVE FOR TYPING
7935  043042  104401                  TYPE                    ;;GO TYPE
7936  043044  000000          3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
7937  043046  012600                  MOV     (SP)+,R0        ;;RESTORE R0
7938  043050  012616                  MOV     (SP)+,(SP)      ;;RESTORE THE STACK
7939  043052  000207                  RTS     PC              ;;RETURN
7940                          .SBTTL  INTEGER MULTIPLY ROUTINE
7941
7942                          ;;**************************************************************
7943                          ;*CALL
7944                          ;*      MOV     MULTIPLER,-(SP)
7945                          ;*      MOV     MULTIPLICAND,-(SP)
7946                          ;*      JSR     PC,@#$MULT
7947                          ;*      RETURN  ;;PRODUCT IS ON THE STACK
7948                          ;*
7949                          ;*      STACK   PRODUCT
7950                          ;*      -----   -------
7951                          ;*      TOP     LSB'S
7952                          ;*      +2      MSB'S
7953
7954  043054          $MULT:
7955  043054  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7956  043056  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7957  043060  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7958  043062  005046                  CLR     -(SP)           ;;CLEAR THE SIGN KEY
7959  043064  016601  000012          MOV     12(SP),R1       ;;GET THE MULTIPLICAND
7960  043070  100002                  BPL     1$              ;;BR IF PLUS
7961  043072  005216                  INC     (SP)            ;;SET THE SIGN KEY
7962  043074  005401                  NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
7963  043076  016602  000014  1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
7964  043102  100002                  BPL     2$              ;;BR IF PLUS
7965  043104  005316                  DEC     (SP)            ;;UPDATE THE SIGN KEY
7966  043106  005402                  NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
7967  043110  012746  000021  2$:     MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
7968  043114  005000                  CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
7969  043116  103001          3$:     BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
7970  043120  060200                  ADD     R2,R0
7971  043122  006000          4$:     ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
7972  043124  006001                  ROR     R1              ;;THE MULTIPLICAND
7973  043126  005316                  DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
7974  043130  001372                  BNE     3$              ;;BR IF NO
7975  043132  022616                  CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
7976  043134  001403                  BEQ     5$              ;;GO TO EXIT IF NO
7977  043136  005400                  NEG     R0              ;;YES--SO MAKE IT SO
7978  043140  005401                  NEG     R1
7979  043142  005600                  SBC     R0
7980  043144  005726          5$:     TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
7981  043146  010066  000012          MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
7982  043152  010166  000010          MOV     R1,10(SP)       ;;LSB'S
7983  043156  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
7984  043160  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
7985  043162  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
7986  043164  000207                  RTS     PC
```

```
                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

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

043166                          $SAVREG:
043166  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043170  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043172  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043174  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
043176  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
043200  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
043202  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
043206  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
043212  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
043216  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
043222  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
043224                          $RESREG:
043224  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
043230  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
043234  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
043240  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
043244  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
043246  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
043250  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
043252  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
043254  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
043256  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
043260  000002                          RTI
                                .SBTTL  TRAP DECODER

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

043262  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
043264  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
043270  005740                          TST     -(R0)           ;;BACKUP BY 2
```

```
 8043  043272  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 8044  043274  006300                  ASL     R0              ;;POSITION FOR INDEXING
 8045  043276  016000  043316          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 8046  043302  000200                  RTS     R0              ;;GO TO ROUTINE
 8047
 8048
 8049                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 8050
 8051  043304  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 8052  043306  016666  000004  000002  MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 8053  043314  000002                  RTI                     ;;RESTORE THE PSW
 8054
 8055                                  .SBTTL  TRAP TABLE
 8056
 8057                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 8058                                  ;*BY THE "TRAP" INSTRUCTION.
 8059
 8060                                  ;       ROUTINE
 8061                                  ;       -------
 8062  043316  043304          $TRPAD: .WORD   $TRAP2
 8063  043320  037632                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 8064  043322  040632                  $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 8065  043324  040606                  $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 8066  043326  040646                  $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 8067  043330  040114                  $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 8068
 8069  043332  041422                  $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 8070
 8071  043334  041332                  $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 8072  043336  041674                  $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 8073  043340  041764                  $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 8074  043342  042304                  $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 8075  043344  043166                  $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
 8076  043346  043224                  $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
 8077  043350  036140                  SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
 8078
```

```
                                        .SBTTL  SERVICE MESSAGES

043352  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
043360  051525  051040  030113
043366  020066  051104  053111
043374  020105  044504  043501
043402  047516  052123  041511
043410  005015  040515  047111          .ASCII  <CR><LF>/MAINDEC-11-DZR6I-D-PB/<CR><LF>
043416  042504  026503  030461
043424  042055  051132  044466
043432  042055  050055  006502
043440      012
043441      015  004412  025052          .ASCII  <CR><LF>/       ***** CAUTION *****/<CR><LF>
043446  025052  020052  040503
043454  052125  047511  020116
043462  025052  025052  006452
043470      012
043471      015  052012  044510          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY BY TYPING CONTROL-C/
043476  020123  051120  043517
043504  040522  020115  044123
043512  052517  042114  020040
043520  042502  044040  046101
043526  042524  020104  047117
043534  054514  041040  020131
043542  054524  044520  043516
043550  041440  047117  051124
043556  046117  041455
043562  005015  052117  042510          .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND, OR THE DRIVE/
043570  053522  051511  026105
043576  041440  051101  051124
043604  042111  042507  043040
043612  051117  040515  052124
043620  047111  020107  047101
043626  026104  047440  020122
043634  044124  020105  051104
043642  053111      105
043645      015  046412  054501          .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/
043652  041040  020105  042514
043660  052106  044440  020116
043666  047101  052440  042116
043674  052105  051105  044515
043702  042516  020104  052123
043710  052101      105
043713      015  044412  044516          .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
043720  044524  046101  054514
043726  020054  051104  053111
043734  051505  052040  020117
043742  042502  052040  051505
043750  042524  020104  044123
043756  052517  042114  044040
043764  053101  035105  005015
043772  005015  027101  044040          .ASCII  <CR><LF>/A. HEADS MANUALLY LOADED/
044000  040505  051504  046440
044006  047101  040525  046114
```

```
 8135  044014  020131  047514  042101
 8136  044022  042105
 8137  044024  005015  027102  041440          .ASCII  <CR><LF>/B. CORRECT PORT SELECTED/
 8138  044032  051117  042522  052103
 8139  044040  050040  051117  020124
 8140  044046  042523  042514  052103
 8141  044054  042105
 8142  044056  005015  027103  053440          .ASCII  <CR><LF>/C. WRITE LOCK DISABLED/
 8143  044064  044522  042524  046040
 8144  044072  041517  020113  044504
 8145  044100  040523  046102  042105
 8146  044106  005015  027104  042040          .ASCII  <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
 8147  044114  044522  042526  051040
 8148  044122  040505  054504  044440
 8149  044130  042116  041511  052101
 8150  044136  051117  047440  006516
 8151  044144      012
 8152  044145      015  042012  044522          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
 8153  044152  042526  020123  047516
 8154  044160  020124  047524  041040
 8155  044166  020105  042524  052123
 8156  044174  042105  046440  051525
 8157  044202  020124  040510  042526
 8158  044210  005015  047502  044124          .ASCIZ  <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
 8159  044216  050040  051117  051524
 8160  044224  042040  051505  046105
 8161  044232  041505  042524  006504
 8162  044240  000012
 8163
 8164  044242  005015  042502  051440  MSG2:   .ASCIZ  <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
 8165  044250  051125  020105  047524
 8166  044256  050040  052125  051440
 8167  044264  051103  052101  044103
 8168  044272  050040  041501  020113
 8169  044300  047111  042040  044522
 8170  044306  042526  030040      000
 8171  044313      015  042012  044522  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
 8172  044320  042526  051450  020051
 8173  044326  047524  041040  020105
 8174  044334  042524  052123  042105
 8175  044342  020072      000
 8176  044345      015  052012  050131  MSG4:   .ASCIZ  <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440:  /
 8177  044352  020105  052502  051523
 8178  044360  040440  042104  042522
 8179  044366  051523  044440  020106
 8180  044374  047516  020124  033461
 8181  044402  032067  030064  020072
 8182  044410  000040
 8183  044412  005015  054524  042520  MSG5:   .ASCIZ  <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210:  /
 8184  044420  041440  047117  051124
 8185  044426  046117  042514  020122
 8186  044434  047111  042524  051122
 8187  044442  050125  020124  042526
 8188  044450  052103  051117  044440
 8189  044456  020106  047516  020124
 8190  044464  030462  035060  020040
```

```
8191  044472  000
8192  044473  015  044412  052116  MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
8193  044500  051105  052522  052120
8194  044506  047440  041503  051125
8195  044514  042522  020104  052101
8196  044522  050040  036503  000
8197  044527  015  042012  044522  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
8198  044534  042526  030040  053440
8199  044542  046111  020114  047516
8200  044550  020124  042502  052040
8201  044556  051505  042524  000104
8202  044564  005015  042522  042101  MSG8:   .ASCIZ  <CR><LF>/READ DATA WITH OFFSET TEST/<CR><LF>
8203  044572  042040  052101  020101
8204  044600  044527  044124  047440
8205  044606  043106  042523  020124
8206  044614  042524  052123  005015
8207  044622  000
8208  044623  015  044012  040505  MSG9:   .ASCIZ  <CR><LF>/HEAD NO./
8209  044630  020104  047516  000056
8210  044636  005015  053412  046111  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
8211  044644  020114  042524  052123
8212  044652  042040  044522  042526
8213  044660  035123  000
8214  044663  015  005012  047520  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
8215  044670  042527  020122  050125
8216  044676  051040  051505  040524
8217  044704  052122  052040  020117
8218  044712  042524  052123  030440
8219  044720  005015  000
8220  044723  015  005012  044124  MSG12:  .ASCII  <CR><LF><LF>/THE ABOVE OFFSET FAILURES ARE NOT ERRORS/
8221  044730  020105  041101  053117
8222  044736  020105  043117  051506
8223  044744  052105  043040  044501
8224  044752  052514  042522  020123
8225  044760  051101  020105  047516
8226  044766  020124  051105  047522
8227  044774  051522
8228  044776  005015  052502  020124          .ASCIZ  <CR><LF>/BUT INDICATORS OF SURFACE,HEAD, & ELECTRONICS QUALITY/<CR><LF>
8229  045004  047111  044504  040503
8230  045012  047524  051522  047440
8231  045020  020106  052523  043122
8232  045026  041501  026105  042510
8233  045034  042101  020054  020046
8234  045042  046105  041505  051124
8235  045050  047117  041511  020123
8236  045056  052521  046101  052111
8237  045064  006531  000012
8238  045070  005015  047516  046040  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
8239  045076  047440  020122  020120
8240  045104  046103  041517  051513
8241  045112  050040  042522  042523
8242  045120  052116
8243  045122  005015  042510  042101          .ASCIZ  <CR><LF>/HEAD SWITCHING TIME TEST BYPASSED/
8244  045130  051440  044527  041524
8245  045136  044510  043516  052040
8246  045144  046511  020105  042524
```

```
8247  045152  052   041040  050131
8248  045160  061.  042523  000104
8249  045166  005015  054502  040520  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
8250  045174  051523  047111  020107
8251  045202  051104  053111  020105
8252  045210      000
8253  045211      015  005012  051104  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
8254  045216  053111  020105      000
8255  045223      015  042012  044522  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL NO. /
8256  045230  042526  051440  051105
8257  045236  040511  020114  047516
8258  045244  020056      000
8259  045247      015  041412  051101  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
8260  045254  051124  042111  042507
8261  045262  051440  051105  040511
8262  045270  020114  047516  020056
8263  045276      000
8264  045277      015  005012  041101  MSG26:  .ASCIZ  <CR><LF><LF>/ABORTING BALANCE OF TESTS ON THIS DRIVE/<CR><LF><LF>
8265  045304  051117  044524  043516
8266  045312  041040  046101  047101
8267  045320  042503  047440  020106
8268  045326  042524  052123  020123
8269  045334  047117  052040  044510
8270  045342  020123  051104  053111
8271  045350  006505  005012      000
8272  045355      015  005012  046101  MSG27:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
8273  045362  020114  051104  053111
8274  045370  051505  052040  051505
8275  045376  042524  006504  005012
8276  045404      000
8277  045405      015  047012  020117  MSG37:  .ASCIZ  <CR><LF>/NO WRITE CHECK ERROR AT MAX POSITIVE OFFSET/
8278  045412  051127  052111  020105
8279  045420  044103  041505  020113
8280  045426  051105  047522  020122
8281  045434  052101  046440  054101
8282  045442  050040  051517  052111
8283  045450  053111  020105  043117
8284  045456  051506  052105      000
8285  045463      015  047012  020117  MSG38:  .ASCIZ  <CR><LF>/NO WRITE CHECK ERROR AT MAX NEGATIVE OFFSET/<CR><LF>
8286  045470  051127  052111  020105
8287  045476  044103  041505  020113
8288  045504  051105  047522  020122
8289  045512  052101  046440  054101
8290  045520  047040  043505  052101
8291  045526  053111  020105  043117
8292  045534  051506  052105  005015
8293  045542      000
8294  045543      015  053412  044522  MSG39:  .ASCIZ  <CR><LF>/WRITE CHECK FAILURE AT OFFSET =/
8295  045550  042524  041440  042510
8296  045556  045503  043040  044501
8297  045564  052514  042522  040440
8298  045572  020124  043117  051506
8299  045600  052105  036440      000
8300  045605      015  041412  052517  MSG40:  .ASCII  <CR><LF>/COULD NOT READ BAD SECTOR INFO ON CYL 410/
8301  045612  042114  047040  052117
8302  045620  051040  040505  020104
```

```
 8303  045626  040502  020104  042523
 8304  045634  052103  051117  044440
 8305  045642  043116  020117  047117
 8306  045650  041440  046131  032040
 8307  045656  030061
 8308  045660  005015  051117  040440          .ASCIZ  <CR><LF>/OR ALIGNMENT CARTRIDGE USED/<CR><LF>
 8309  045666  044514  047107  042515
 8310  045674  052116  041440  051101
 8311  045702  051124  042111  042507
 8312  045710  052440  042523  006504
 8313  045716  000012
 8314  045720  005015  051120  043517  MSG74:  .ASCIZ  <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
 8315  045726  040522  020115  041101
 8316  045734  051117  020124  042520
 8317  045742  042116  047111  027107
 8318  045750  027056  046120  040505
 8319  045756  042523  053440  044501
 8320  045764  000124
 8321  045766  005015  040510  052114  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING...PLEASE WAIT/
 8322  045774  050040  047105  044504
 8323  046002  043516  027056  050056
 8324  046010  042514  051501  020105
 8325  046016  040527  052111  000000
 8326  046023  000015  050012  047522  MSG76:  .ASCIZ  <CR><LF>/PROGRAM ABORTED/
 8327  046030  051107  046501  040440
 8328  046036  047502  052122  042105
 8329  046044  000000
 8330  046045  000015  041412  052520  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
 8331  046052  044040  046101  042524
 8332  046060  000104
 8333
 8334                                  .SBTTL  ERROR MESSAGES
 8335
 8336  046062  005015  051105  047522  EM1:    .ASCIZ  <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
 8337  046070  026122  047440  046116
 8338  046076  020131  020060  044124
 8339  046104  052522  033440  040440
 8340  046112  046114  053517  042105
 8341  046120  020054  051124  020131
 8342  046126  043501  044501  006516
 8343  046134  000012
 8344  046136  051104  053111  020105  EM2:    .ASCIZ  /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
 8345  046144  020043  047111  051040
 8346  046152  041513  031123  041440
 8347  046160  047101  047516  020124
 8348  046166  042502  051040  040505
 8349  046174  020104  040502  045503
 8350  046202  041440  051117  042522
 8351  046210  052103  054514  044440
 8352  046216  020116  045522  051115
 8353  046224  000062
 8354  046226  005015  041101  051117  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
 8355  046234  020124  042524  052123
 8356  046242  027123  027056  047125
 8357  046250  054105  042520  052103
 8358  046256  042105  052040  046511
```

```
 8359  046264  020105  052517  020124
 8360  046272  052101  050040  036503
 8361  046300     000
 8362  046301     015  040412  047502  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
 8363  046306  052122  052040  051505
 8364  046314  051524  027056  052456
 8365  046322  042516  050130  041505
 8366  046330  042524  020104  047111
 8367  046336  042524  051122  050125
 8368  046344  020124  052101  050040
 8369  046352  036503     000
 8370  046355     115  051504  051440  EM5:    .ASCIZ  /MDS SET IN RKCS2/
 8371  046362  052105  044440  020116
 8372  046370  045522  051503  000062
 8373  046376  043125  020105  042523  EM6:    .ASCIZ  /UFE SET IN RKCS2/
 8374  046404  020124  047111  051040
 8375  046412  041513  031123     000
 8376  046417     104  040522  044440  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
 8377  046424  020116  045522  051504
 8378  046432  023040  047040  042105
 8379  046440  044440  020116  045522
 8380  046446  051503  020062  042522
 8381  046454  042523  035524  053440
 8382  046462  047522  043516  050040
 8383  046470  051117  020124  042523
 8384  046476  042514  052103  042105
 8385  046504  000077
 8386  046506  051104  053111  020105  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
 8387  046514  051120  051505  047105
 8388  046522  020124  052502  020124
 8389  046530  047516  020124  050123
 8390  046536  041505  043111  042511
 8391  046544  020104  054502  047440
 8392  046552  042520  040522  047524
 8393  046560  000122
 8394  046562  051104  053111  020105  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
 8395  046570  047516  020124  051120
 8396  046576  051505  047105  020124
 8397  046604  052502  020124  050123
 8398  046612  041505  043111  042511
 8399  046620  020104  054502  047440
 8400  046626  042520  040522  047524
 8401  046634  000122
 8402  046636  041101  051117  020124  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
 8403  046644  042524  052123  027123
 8404  046652  027056  040503  047116
 8405  046660  052117  051040  043105
 8406  046666  051105  047105  042503
 8407  046674  041440  047117  051124
 8408  046702  046117  042514  020122
 8409  046710  042522  044507  052123
 8410  046716  051105     000
 8411  046721     104  040522  044440  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
 8412  046726  020116  045522  051504
 8413  046734  023040  047040  042105
 8414  046742  044440  020116  045522
```

```
8415  046750  051503  020062  047502
8416  046756  044124  051440  052105
8417  046764     000
8418  046765     103  047117  051124  EM12:   .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
8419  046772  046117  042514  020122
8420  047000  047516  020124  042522
8421  047006  042101  020131  047111
8422  047014  051040  041513  030523
8423  047022     000
8424  047023     116  020117  052101  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
8425  047030  047124  044440  020116
8426  047036  045522  051501  043117
8427  047044     000
8428  047045     125  042516  050130  EM14:   .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
8429  047052  041505  042524  020104
8430  047060  042515  047515  054522
8431  047066  050040  051101  052111
8432  047074  020131  051124  050101
8433  047102     000
8434  047103     122  042113  020103  EM15:   .ASCII  /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
8435  047110  020046  045522  040504
8436  047116  044440  042116  041511
8437  047124  052101  020105  044124
8438  047132  052101  053440  042503
8439  047140  047440  041503  051125
8440  047146  042522  020104  052101
8441  047154  005015  054503  020114          .ASCIZ  <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
8442  047162  030464  026061  052040
8443  047170  040522  045503  031040
8444  047176  020054  042523  052103
8445  047204  051117  031040  000061
8446  047212  040503  047116  052117  EM16:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
8447  047220  051040  040505  020104
8448  047226  040502  020104  042523
8449  047234  052103  051117  044440
8450  047242  043116  051117  040515
8451  047250  044524  047117     000
8452  047255     115  051505  040523  EM17:   .ASCIZ  /MESSAGE A0 ERROR/
8453  047262  042507  040440  020060
8454  047270  051105  047522  000122
8455  047276  042515  051523  043501  EM18:   .ASCIZ  /MESSAGE B0 ERROR/
8456  047304  020105  030102  042440
8457  047312  051122  051117     000
8458  047317     115  051505  040523  EM19:   .ASCIZ  /MESSAGE A1 ERROR/
8459  047324  042507  040440  020061
8460  047332  051105  047522  000122
8461  047340  042515  051523  043501  EM20:   .ASCIZ  /MESSAGE B1 ERROR/
8462  047346  020105  030502  042440
8463  047354  051122  051117     000
8464  047361     103  051105  020122  EM21:   .ASCIZ  /CERR SET IN RKCS1/
8465  047366  042523  020124  047111
8466  047374  051040  041513  030523
8467  047402     000
8468  047403     116  020117  051104  EM22:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/<CR><LF>
8469  047410  053111  051505  043040
8470  047416  052517  042116  044440
```

```
8471   047424  020116  042504  044526
8472   047432  042503  046440  050101
8473   047440  024040  042044  053105
8474   047446  024515  005015
8475   047452  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY AND RESTART/<CR><LF>
8476   047460  047503  051122  041505
8477   047466  046124  020131  047101
8478   047474  020104  042522  052123
8479   047502  051101  006524  000012
8480   047510  047516  042040  044522  EM23:   .ASCII  /NO DRIVES FOUND ON BUSS/<CR><LF>
8481   047516  042526  020123  047506
8482   047524  047125  020104  047117
8483   047532  041040  051525  006523
8484   047540     012
8485   047541     123  052105  050125          .ASCIZ  /SETUP CORRECTLY AND PRESS 'CONTINUE'/<CR><LF>
8486   047546  041440  051117  042522
8487   047554  052103  054514  040440
8488   047562  042116  050040  042522
8489   047570  051523  023440  047503
8490   047576  052116  047111  042525
8491   047604  006447  000012
8492   047610  047526  020114  040526  EM24:   .ASCIZ  /VOL VALID NOT SET IN RKMR2/
8493   047616  044514  020104  047516
8494   047624  020124  042523  020124
8495   047632  047111  051040  046513
8496   047640  031122     000
8497   047643     015  042012  052105  EM25:   .ASCIZ  <CR><LF>/DETECTED 10 BAD SECTORS...ABORTING TEST/
8498   047650  041505  042524  020104
8499   047656  030061  041040  042101
8500   047664  051440  041505  047524
8501   047672  051522  027056  040456
8502   047700  047502  052122  047111
8503   047706  020107  042524  052123
8504   047714     000
8505   047715     104  052105  041505  EM26:   .ASCIZ  /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
8506   047722  042524  020104  051502
8507   047730  020105  052502  020124
8508   047736  047516  020124  044514
8509   047744  052123  042105  044440
8510   047752  020116  040502  020104
8511   047760  042523  052103  051117
8512   047766  043040  046111  000105
8513   047774  042504  042524  052103  EM27:   .ASCII  /DETECTED BSE IN READ COMMAND/
8514   050002  042105  041040  042523
8515   050010  044440  020116  042522
8516   050016  042101  041440  046517
8517   050024  040515  042116
8518   050030  005015  052502  020124          .ASCIZ  <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
8519   050036  047516  020124  047111
8520   050044  050040  042522  044526
8521   050052  052517  020123  051127
8522   050060  052111  020105  047503
8523   050066  046515  047101  020104
8524   050074  047524  051440  046501
8525   050102  020105  042523  052103
8526   050110  051117     000
```

```
 8527  050113    103  046131  040440  EM36:   .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
 8528  050120  042104  020122  047111
 8529  050126  051040  046513  031522
 8530  050134  047040  052117  051440
 8531  050142  046501  020105  051501
 8532  050150  051040  042113  000103
 8533  050156  054503  020114  044504  EM39:   .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
 8534  050164  043106  023040  047440
 8535  050172  043106  042523  020124
 8536  050200  047111  051040  046513
 8537  050206  031122  047040  052117
 8538  050214  041440  042514  051101
 8539  050222  042105     000
 8540  050225    103  046131  040440  EM40:   .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
 8541  050232  042104  020122  047111
 8542  050240  051040  046513  031522
 8543  050246  047040  052117  041440
 8544  050254  042514  051101  042105
 8545  050262     000
 8546  050263    103  046131  040440  EM41:   .ASCIZ  /CYL ADDR IN B2 DID NOT REMAIN CLEARED/
 8547  050270  042104  020122  047111
 8548  050276  041040  020062  044504
 8549  050304  020104  047516  020124
 8550  050312  042522  040515  047111
 8551  050320  041440  042514  051101
 8552  050326  042105     000
 8553  050331    101  052124  020116  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
 8554  050336  047516  020124  046103
 8555  050344  040505  042522  020104
 8556  050352  047111  051040  040513
 8557  050360  047523  000106
 8558  050364  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
 8559  050372  020124  047111  051040
 8560  050400  041513  031123     000
 8561  050405    122  040505  020104  EM65:   .ASCIZ  /READ HEADER ERROR/
 8562  050412  042510  042101  051105
 8563  050420  042440  051122  051117
 8564  050426     000
 8565  050427    101  044514  047107  EM69:   .ASCIZ  /ALIGNMENT CARTRIDGE USED/
 8566  050434  042515  052116  041440
 8567  050442  051101  051124  042111
 8568  050450  042507  052440  042523
 8569  050456  000104
 8570  050460  052103  020117  042523  EM73:   .ASCIZ  /CTO SET IN RKCS1/
 8571  050466  020124  047111  051040
 8572  050474  041513  030523     000
 8573  050501    122  055124  047040  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
 8574  050506  052117  051440  052105
 8575  050514  044440  020116  045522
 8576  050522  051115  000062
 8577  050526  042516  020104  042523  EM79:   .ASCIZ  /NED SET IN RKCS2/
 8578  050534  020124  047111  051040
 8579  050542  041513  031123     000
 8580  050547    127  044522  042524  EM80:   .ASCIZ  /WRITE CHECK ERROR SET IN RKCS2/
 8581  050554  041440  042510  045503
 8582  050562  042440  051122  051117
```

```
8583  050570  051440  052105  044440
8584  050576  020116  045522  051503
8585  050604  000062
8586  050606  051127  052111  020105  EM81:   .ASCIZ  /WRITE CHECK COMMAND NOT FUNCTIONING/
8587  050614  044103  041505  020113
8588  050622  047503  046515  047101
8589  050630  020104  047516  020124
8590  050636  052506  041516  044524
8591  050644  047117  047111  000107
8592  050652  042522  042101  042040  EM82:   .ASCIZ  /READ DATA DID NOT COMPARE WITH WRITE DATA/
8593  050660  052101  020101  044504
8594  050666  020104  047516  020124
8595  050674  047503  050115  051101
8596  050702  020105  044527  044124
8597  050710  053440  044522  042524
8598  050716  042040  052101  000101
8599  050724  040504  040524  041440  EM83:   .ASCIZ  /DATA CHECK ERROR SET IN RKER/
8600  050732  042510  045503  042440
8601  050740  051122  051117  051440
8602  050746  052105  044440  020116
8603  050754  045522  051105     000  EM84:   .ASCIZ  /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
8604  050761     127  044510  042514
8605  050766  053440  044501  044524
8606  050774  043516  043040  051117
8607  051002  041440  047117  051124
8608  051010  051040  040505  054504
8609  051016  047440  020122  043101
8610  051024  042524  020122  047503
8611  051032  052116  020122  042522
8612  051040  042101  020131  042522
8613  051046  023503  000104
8614  051052  043117  051506  052105  EM85:   .ASCIZ  /OFFSET STATUS BIT IN RKMR2 CLEARED/
8615  051060  051440  040524  052524
8616  051066  020123  044502  020124
8617  051074  047111  051040  046513
8618  051102  031122  041440  042514
8619  051110  051101  042105     000  EM86:   .ASCIZ  /OFFSET REG IN A2 NOT = RKASOF/
8620  051115     117  043106  042523
8621  051122  020124  042522  020107
8622  051130  047111  040440  020062
8623  051136  047516  020124  020075
8624  051144  045522  051501  043117
8625  051152     000
8626  051153     104  042111  047040  EM88:   .ASCIZ  /DID NOT FIND SECTOR 0 FROM INDEX/
8627  051160  052117  043040  047111
8628  051166  020104  042523  052103
8629  051174  051117  030040  043040
8630  051202  047522  020115  047111
8631  051210  042504  000130
8632  051214  042522  042101  047111  EM93:   .ASCIZ  /READING WRONG CYLINDER # IN HEADER...MISPOSITION/
8633  051222  020107  051127  047117
8634  051230  020107  054503  044514
8635  051236  042116  051105  021440
8636  051244  044440  020116  042510
8637  051252  042101  051105  027056
8638  051260  046456  051511  047520
```

```
8639  051266  044523  044524  047117
8640  051274     000
8641  051275     117  043106  042523  EM94:   .ASCIZ  /OFFSET IT IN A2 NOT CLEARED/
8642  051302  020124  052111  044440
8643  051310  020116  031101  047040
8644  051316  052117  041440  042514
8645  051324  051101  042105     000
8646  051331     106  051117  040515  EM95:   .ASCIZ  /FORMAT BIT NOT SET IN RKMR2/
8647  051336  020124  044502  020124
8648  051344  047516  020124  042523
8649  051352  020124  047111  051040
8650  051360  046513  031122     000
8651  051365     103  047101  047516  EM96:   .ASCIZ  /CANNOT FIND SECTOR 23(8)/
8652  051372  020124  044506  042116
8653  051400  051440  041505  047524
8654  051406  020122  031462  034050
8655  051414  000051
8656  051416  042510  042101  051440  EM97:   .ASCIZ  /HEAD SWITCHING REQ'D ANOTHER FULL REVOLUTION OF DISK/
8657  051424  044527  041524  044510
8658  051432  043516  051040  050505
8659  051440  042047  040440  047516
8660  051446  044124  051105  043040
8661  051454  046125  020114  042522
8662  051462  047526  052514  044524
8663  051470  047117  047440  020106
8664  051476  044504  045523     000
8665  051503     104  044522  042526  EM100:  .ASCIZ  /DRIVE OFF TRACK SET IN RKMR3/
8666  051510  047440  043106  052040
8667  051516  040522  045503  051440
8668  051524  052105  044440  020116
8669  051532  045522  051115  000063
8670  051540  044504  020104  047516  EM101:  .ASCIZ  /DID NOT GO TO CYLINDER 10/
8671  051546  020124  047507  052040
8672  051554  020117  054503  044514
8673  051562  042116  051105  030440
8674  051570  000060
8675
8676                                  .SBTTL  DATA HEADERS
8677
8678  051572  042524  052123  047040  DH1:    .ASCIZ  /TEST NO.  PC/
8679  051600  027117  020040  041520
8680  051606     000
8681  051607     122  046513  030522  DH2:    .ASCIZ  /RKMR1  RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2/
8682  051614  051011  046513  031122
8683  051622  051011  046513  031522
8684  051630  051011  042513  004522
8685  051636  045522  051504  051011
8686  051644  041513  030523  051011
8687  051652  041513  031123     000
8688  051657     122  053513  004503  DH3:    .ASCIZ  /RKWC   RKBA    RKDA    RKASOF  RKDC    RKECPS  RKECPT/
8689  051664  045522  040502  051011
8690  051672  042113  004501  045522
8691  051700  051501  043117  051011
8692  051706  042113  004503  045522
8693  051714  041505  051520  051011
8694  051722  042513  050103  000124
```

```
 8695  051730  051106  046517  041440  DH6:    .ASCIZ  /FROM CYL  TO CYL  CYL DIFF/
 8696  051736  046131  020040  047524
 8697  051744  041440  046131  020040
 8698  051752  054503  020114  044504
 8699  051760  043106     000
 8700  051763     124  051505  020124  DH8:    .ASCIZ  /TEST NO.       TRAP PC/
 8701  051770  047516  004456  051124
 8702  051776  050101  050040  000103
 8703  052004  043101  042524  020122  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
 8704  052012  052123  051101  020124
 8705  052020  050123  047111  046104
 8706  052026  020105  047503  046515
 8707  052034  047101  020104  042522
 8708  052042  023503  020104  054502
 8709  052050  042040  044522  042526
 8710  052056     000
 8711  052057     101  020124  047105  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
 8712  052064  020104  043117  044040
 8713  052072  040505  020104  047514
 8714  052100  042101  047111  000107
 8715  052106  054105  042520  052103  DH11:   .ASCIZ  /EXPECTED       WAS/
 8716  052114  042105  053411  051501
 8717  052122     000
 8718  052123     117  020116  042523  DH13:   .ASCIZ  /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
 8719  052130  052103  051117  020123
 8720  052136  030061  020054  031061
 8721  052144  020054  032061  020054
 8722  052152  033061  020054  034061
 8723  052160  047440  020122  030062
 8724  052166  041440  046131  032040
 8725  052174  030061  052040  040522
 8726  052202  045503  031040     000
 8727  052207     117  020116  042523  DH14:   .ASCIZ  /ON SECTORS 11, 13, 15, 17, 19 OR 21 CYL 410 TRACK 2/
 8728  052214  052103  051117  020123
 8729  052222  030461  020054  031461
 8730  052230  020054  032461  020054
 8731  052236  033461  020054  034461
 8732  052244  047440  020122  030462
 8733  052252  041440  046131  032040
 8734  052260  030061  052040  040522
 8735  052266  045503  031040     000
 8736  052273     101  052106  051105  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
 8737  052300  051040  041505  046101
 8738  052306  041440  046517  040515
 8739  052314  042116     000
 8740  052317     101  052106  051105  DH19:   .ASCIZ  /AFTER PACK COMMAND/
 8741  052324  050040  041501  020113
 8742  052332  047503  046515  047101
 8743  052340  000104
 8744  052342  043101  042524  020122  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
 8745  052350  042523  042514  052103
 8746  052356  042040  044522  042526
 8747  052364  041440  046517  040515
 8748  052372  042116     000
 8749  052375     101  052106  051105  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
 8750  052402  051440  041125  054523
```

```
 8751  052410  052123  046505  041440
 8752  052416  042514  051101     000
 8753  052423     101  052106  051105  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
 8754  052430  042040  044522  042526
 8755  052436  041440  042514  051101
 8756  052444  041440  046517  040515
 8757  052452  042116     000
 8758  052455     101  052106  051105  DH24:   .ASCIZ  /AFTER OFFSET COMMAND/
 8759  052462  047440  043106  042523
 8760  052470  020124  047503  046515
 8761  052476  047101  000104
 8762  052502  043101  042524  020122  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
 8763  052510  042523  045505  041440
 8764  052516  046517  040515  042116
 8765  052524     000
 8766  052525     101  052106  051105  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
 8767  052532  051040  040505  020104
 8768  052540  040504  040524  041440
 8769  052546  046517  040515  042116
 8770  052554     000
 8771  052555     101  052106  051105  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
 8772  052562  053440  044522  042524
 8773  052570  042040  052101  020101
 8774  052576  047503  046515  047101
 8775  052604  000104
 8776  052606  042411  050130  041505  DH28:   .ASCIZ  /       EXPECT/
 8777  052614  000124
 8778  052616  040411  052103  040525  DH29:   .ASCIZ  /       ACTUAL/
 8779  052624  000114
 8780  052626  043101  042524  020122  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
 8781  052634  042522  042101  044040
 8782  052642  040505  042504  020122
 8783  052650  047503  046515  047101
 8784  052656  000104
 8785  052660  030101  041011  004460  DH31:   .ASCIZ  /A0     B0      A1      B1      A2      B2      B3/
 8786  052666  030501  041011  004461
 8787  052674  031101  041011  004462
 8788  052702  031502     000
 8789  052705     101  052106  051105  DH32:   .ASCIZ  /AFTER WRITE CHECK COMMAND/
 8790  052712  053440  044522  042524
 8791  052720  041440  042510  045503
 8792  052726  041440  046517  040515
 8793  052734  042116     000
 8794  052737     101  052106  051105  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
 8795  052744  053440  044522  042524
 8796  052752  044040  040505  042504
 8797  052760  020122  047503  046515
 8798  052766  047101  000104
 8799  052772  047527  042122  004443  DH40:   .ASCIZ  /WORD#  HEADER WAS  SHOULD BE/
 8800  053000  042510  042101  051105
 8801  053006  053440  051501  020040
 8802  053014  044123  052517  042114
 8803  053022  041040  000105
 8804  053026  052504  044522  043516  DH41:   .ASCIZ  /DURING RECAL COMMAND/
 8805  053034  051040  041505  046101
 8806  053042  041440  046517  040515
```

```
8807   053050  042116     000
8808   053053     117  020116  042523  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
8809   053060  052103  051117  020123
8810   053066  026060  026062  026064
8811   053074  020066  051117  034040
8812   053102  020040  054503  020114
8813   053110  030464  020060  051124
8814   053116  041501  020113  000062
8815   053124  047117  051440  041505  DH43:   .ASCIZ  /ON SECTORS 1,3,5,7 OR 9  CYL 410 TRACK 2/
8816   053132  047524  051522  030440
8817   053140  031454  032454  033454
8818   053146  047440  020122  020071
8819   053154  041440  046131  032040
8820   053162  030061  052040  040522
8821   053170  045503  031040     000
8822   053175     106  051117  040515  DH44:   .ASCIZ  /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8823   053202  020124  020046  046101
8824   053210  020114  042522  042101
8825   053216  053455  044522  042524
8826   053224  052040  051505  051524
8827   053232  053440  046111  020114
8828   053240  042502  041040  050131
8829   053246  051501  042523  000104
8830   053254  043101  042524  020122  DH47:   .ASCIZ  /AFTER READ HEADER COMMAND WITH MOVEMENT/
8831   053262  042522  042101  044040
8832   053270  040505  042504  020122
8833   053276  047503  046515  047101
8834   053304  020104  044527  044124
8835   053312  046440  053117  046505
8836   053320  047105  000124
8837   053324  051515  020107  020101  DH49:   .ASCIZ  /MSG A & B IN RKMR2 & RKMR3 RESP. ARE INVALID/
8838   053332  020046  020102  047111
8839   053340  051040  046513  031122
8840   053346  023040  051040  046513
8841   053354  031522  051040  051505
8842   053362  027120  040440  042522
8843   053370  044440  053116  046101
8844   053376  042111     000
8845   053401     101  052106  051105  DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
8846   053406  051440  042505  020113
8847   053414  047524  051440  046105
8848   053422  020106  047503  046515
8849   053430  047101  000104
8850   053434  044527  044124  044440  DH52:   .ASCIZ  /WITH INTENTIONAL MISCOMPARE/
8851   053442  052116  047105  044524
8852   053450  047117  046101  046440
8853   053456  051511  047503  050115
8854   053464  051101  000105
8855   053470  052504  044522  043516  DH53:   .ASCIZ  /DURING OFFSET COMMAND/
8856   053476  047440  043106  042523
8857   053504  020124  047503  046515
8858   053512  047101  000104
8859   053516  043101  042524  020122  DH54:   .ASCIZ  /AFTER FORMAT CHANGE AND CONTR READY REC'D/
8860   053524  047506  046522  052101
8861   053532  041440  040510  043516
8862   053540  020105  047101  020104
```

```
 8863  053546  047503  052116  020122
 8864  053554  042522  042101  020131
 8865  053562  042522  023503  000104
 8866  053570  054503  020114  004443  DH56:   .ASCIZ  /CYL #  HEADER WORD 0/
 8867  053576  042510  042101  051105
 8868  053604  053440  051117  020104
 8869  053612  000060
 8870  053614  043101  042524  020122  DH57:   .ASCIZ  /AFTER WRITE COMMAND WITH OFFSET/
 8871  053622  051127  052111  020105
 8872  053630  047503  046515  047101
 8873  053636  020104  044527  044124
 8874  053644  047440  043106  042523
 8875  053652  000124
 8876
 8877                                  .SBTTL  ERROR OUTPUT DATA
 8878
 8879                                          .EVEN
 8880  053654  001214  001116          DT1:    $TESTN,$ERRPC
 8881  053660  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8882  053666  007354  007352  007340
 8883  053674  007342
 8884  053676  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8885  053704  007356  007360  007372
 8886  053712  007374
 8887  053714  001214  001334          DT3:    $TESTN,TRAPPC
 8888  053720  001214  001116  001344  DT4:    $TESTN,$ERRPC,FRCYL,TOCYL,CALDIF
 8889  053726  001346  001354
 8890  053732  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8891  053740  007354  007352  007340
 8892  053746  007342
 8893  053750  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8894  053756  007356  007360  007372
 8895  053764  007374
 8896  053766  001214  001116  001440  DT6:    $TESTN,$ERRPC,WD2,WD1
 8897  053774  001436
 8898  053776  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8899  054004  007354  007352  007340
 8900  054012  007342
 8901  054014  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8902  054022  007356  007360  007372
 8903  054030  007374
 8904  054032  001214  001116  001456  DT7:    $TESTN,$ERRPC,WDCNT,HDWD,TEMP1
 8905  054040  001474  007376
 8906  054044  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8907  054052  007354  007352  007340
 8908  054060  007342
 8909  054062  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8910  054070  007356  007360  007372
 8911  054076  007374
 8912  054100  001214  001116  001346  DT8:    $TESTN,$ERRPC,TOCYL,FRCYL,CALDIF
 8913  054106  001344  001354
 8914  054112  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8915  054120  007354  007352  007340
 8916  054126  007342
 8917  054130  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8918  054136  007356  007360  007372
```

```
 8919  054144  007374
 8920  054146  001214  001116  001346  DT9:    $TESTN,$ERRPC,TOCYL,RHTAB
 8921  054154  001712
 8922  054156  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8923  054164  007354  007352  007340
 8924  054172  007342
 8925  054174  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8926  054202  007356  007360  007372
 8927  054210  007374
 8928  054212  001214  001116  007430  DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
 8929  054220  007432  007434  007436
 8930  054226  007410  007412  007414
 8931  054234  007416
 8932  054236  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8933  054244  007354  007352  007340
 8934  054252  007342
 8935  054254  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8936  054262  007356  007360  007372
 8937  054270  007374
 8938  054272  001214  001116  007430  DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
 8939  054300  007432  007434  007436
 8940  054306  007440  007442
 8941  054312  007410  007412  007414          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
 8942  054320  007416  007420  007422
 8943  054326  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8944  054334  007354  007352  007340
 8945  054342  007342
 8946  054344  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8947  054352  007356  007360  007372
 8948  054360  007374
 8949  054362  001214  001116  007430  DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
 8950  054370  007432  007434  007436
 8951  054376  007440  007442  007446
 8952  054404  007410  007412  007414          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
 8953  054412  007416  007420  007422
 8954  054420  007426
 8955  054422  007364  007366  007370          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8956  054430  007354  007352  007340
 8957  054436  007342
 8958  054440  007344  007346  007350          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8959  054446  007356  007360  007372
 8960  054454  007374
 8961
 8962                                  .SBTTL  ERROR DATA FORMATS
 8963
 8964  054456  000003                  DF1:    3
 8965  054460     002        000               .BYTE 2,0
 8966  054462  051607                        DH2
 8967  054464     007        000               .BYTE 7,0
 8968  054466  051657                        DH3
 8969  054470     007        000               .BYTE   7,0
 8970
 8971  054472  000001                  DF2:    1
 8972  054474     002        000               .BYTE   2,0
 8973
 8974  054476  000005                  DF3:    5
```

```
8975  054500  000      000          .BYTE  0,0
8976  054502  051572               DH1
8977  054504  002      000          .BYTE  2,0
8978  054506  052106               DH11
8979  054510  002      000          .BYTE  2,0
8980  054512  051607               DH2
8981  054514  007      000          .BYTE  7,0
8982  054516  051657               DH3
8983  054520  007      000          .BYTE  7,0
8984
8985  054522  000003          DF4: 3
8986  054524  002      000          .BYTE  2,0
8987  054526  051607               DH2
8988  054530  007      000          .BYTE  7,0
8989  054532  051657               DH3
8990  054534  007      000          .BYTE  7,0
8991
8992  054536  000005          DF5: 5
8993  054540  000      000          .BYTE  0,0
8994  054542  053324               DH49
8995  054544  000      000          .BYTE  0,0
8996  054546  051572               DH1
8997  054550  002      000          .BYTE  2,0
8998  054552  051607               DH2
8999  054554  007      000          .BYTE  7,0
9000  054556  051657               DH3
9001  054560  007      000          .BYTE  7,0
9002
9003  054562  000005          DF6: 5
9004  054564  000      000          .BYTE 0,0
9005  054566  051572               DH1
9006  054570  002      000          .BYTE 2,0
9007  054572  051730               DH6
9008  054574  003      000          .BYTE 3,0
9009  054576  051607               DH2
9010  054600  007      000          .BYTE  7,0
9011  054602  051657               DH3
9012  054604  007      000          .BYTE  7,0
9013
9014
9015  054606  000004          DF10: 4
9016  054610  000      000          .BYTE 0,0
9017  054612  051572               DH1
9018  054614  002      000          .BYTE 2,0
9019  054616  051607               DH2
9020  054620  007      000          .BYTE 7,0
9021  054622  051657               DH3
9022  054624  007      000          .BYTE  7,0
9023
9024  054626  000004          DF14: 4
9025  054630  002      000          .BYTE  2,0
9026  054632  052772               DH40
9027  054634  003      000          .BYTE  3,0
9028  054636  051607               DH2
9029  054640  007      000          .BYTE  7,0
9030  054642  051657               DH3
```

```
 9031  054644     007     000                .BYTE   7,0
 9032
 9033
 9034  054646  000004                 DF15:  4
 9035  054650     000     000                .BYTE   0,0
 9036  054652  051572                        DH1
 9037  054654     002     000                .BYTE   2,0
 9038  054656  051607                        DH2
 9039  054660     007     000                .BYTE   7,0
 9040  054662  051657                        DH3
 9041  054664     007     000                .BYTE   7,0
 9042
 9043  054666  000005                 DF17:  5
 9044  054670     000     000                .BYTE   0,0
 9045  054672  053175                        DH44
 9046  054674     000     000                .BYTE   0,0
 9047  054676  051572                        DH1
 9048  054700     002     000                .BYTE   2,0
 9049  054702  051607                        DH2
 9050  054704     007     000                .BYTE   7,0
 9051  054706  051657                        DH3
 9052  054710     007     000                .BYTE   7,0
 9053  054712  000005                 DF20:  5
 9054  054714     000     000                .BYTE   0,0
 9055  054716  051572                        DH1
 9056  054720     002     000                .BYTE   2,0
 9057  054722  053570                        DH56
 9058  054724     002     000                .BYTE   2,0
 9059  054726  051607                        DH2
 9060  054730     007     000                .BYTE   7,0
 9061  054732  051657                        DH3
 9062  054734     007     000                .BYTE   7,0
 9063
 9064  054736  000007                 DF21:  7
 9065  054740     000     000                .BYTE   0,0
 9066  054742  051572                        DH1
 9067  054744     002     000                .BYTE   2,0
 9068  054746  052606                        DH28
 9069  054750     000     000                .BYTE   0,0
 9070  054752  052660                        DH31
 9071  054754     004     000                .BYTE   4,0
 9072  054756  052616                        DH29
 9073  054760     004     000                .BYTE   4,0
 9074  054762  051607                        DH2
 9075  054764     007     000                .BYTE   7,0
 9076  054766  051657                        DH3
 9077  054770     007     000                .BYTE   7,0
 9078
 9079  054772  000007                 DF22:  7
 9080  054774     000     000                .BYTE   0,0
 9081  054776  051572                        DH1
 9082  055000     002     000                .BYTE   2,0
 9083  055002  052606                        DH28
 9084  055004     000     000                .BYTE   0,0
 9085  055006  052660                        DH31
 9086  055010     006     000                .BYTE   6,0
```

```
 9087  055012  052616                          DH29
 9088  055014     006      000                 .BYTE   6,0
 9089  055016  051607                          DH2
 9090  055020     007      000                 .BYTE   7,0
 9091  055022  051657                          DH3
 9092  055024     007      000                 .BYTE   7,0
 9093
 9094  055026  000007                  DF23:   7
 9095  055030     000      000                 .BYTE   0,0
 9096  055032  051572                          DH1
 9097  055034     002      000                 .BYTE   2,0
 9098  055036  052606                          DH28
 9099  055040     000      000                 .BYTE   0,0
 9100  055042  052660                          DH31
 9101  055044     007      000                 .BYTE   7,0
 9102  055046  052616                          DH29
 9103  055050     007      000                 .BYTE   7,0
 9104  055052  051607                          DH2
 9105  055054     007      000                 .BYTE   7,0
 9106  055056  051657                          DH3
 9107  055060     007      000                 .BYTE   7,0
 9108                                  .EVEN
 9109                                  ;;*****************************************************************
 9110                                  .SBTTL  TYPE ERROR ROUTINE
 9111                                  ;*ENTRY JSR PC,TYP ERR
 9112                                  ;*RETURN RTS PC
 9113                                  ;*
 9114                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 9115                                  ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
 9116                                  ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
 9117                                  ;*THE ERROR.
 9118                                  ;;*****************************************************************
 9119  055062  104413          TYPERR: SAVREG
 9120  055064  113700  001114          MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
 9121  055070  042700  177400          BIC     #177400,R0      ;CLEAR SIGN EXTENSION
 9122  055074  005300                   DEC     R0              ;FORM INDEX FOR ERROR TABLE
 9123  055076  006300                   ASL     R0
 9124  055100  006300                   ASL     R0
 9125  055102  006300                   ASL     R0
 9126  055104  062700  007512  1$:      ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
 9127  055110  012037  055124          MOV     (R0)+,2$        ;GET EM POINTER
 9128  055114  001404                   BEQ     3$              ;BRANCH IF THERE ISN'T ONE
 9129  055116  104401  001205          TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
 9130  055122  104401                   TYPE                    ;TYPE ERROR MESSAGE (EM)
 9131  055124  000000          2$:     .WORD   0               ;EM POINTER GOES HERE
 9132  055126  012037  055142  3$:      MOV     (R0)+,4$        ;GET DH POINTER
 9133  055132  001404                   BEQ     5$              ;BRANCH IF THERE ISN'T ONE
 9134  055134  104401  001205          TYPE    ,$CRLF          ;TYPE CR-LF
 9135  055140  104401                   TYPE                    ;TYPE DATA HEADER
 9136  055142  000000          4$:     .WORD   0               ;DH POINTER GOES HERE
 9137  055144  012001          5$:     MOV     (R0)+,R1        ;GET DT POINTER
 9138  055146  001455                   BEQ     20$             ;BRANCH IF THERE ARE NONE
 9139  055150  005004                   CLR     R4              ;SET INDENT SWITCH
 9140  055152  012000                   MOV     (R0)+,R0        ;GET DF POINTER
 9141  055154  012002                   MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
 9142  055156  001446                   BEQ     17$             ;DH NUM IS 0-BRANCH
```

```
 9143  055160  005104                    COM     R4              ;NO INDENT
 9144  055162  104401  001205            TYPE    $CRLF
 9145  055166  112003                10$:MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
 9146  055170  105720                    TSTB    (R0)+           ;BUMP PAST FORMAT WORD
 9147  055172  005703                    TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
 9148  055174  001407                    BEQ     14$             ;NO - SKIP DATA PRINT
 9149  055176  013146                11$:MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
 9150  055200  104402                    TYPOC                   ;TYPE IT
 9151  055202  005303                    DEC     R3              ;MORE DATA WORDS
 9152  055204  001403                    BEQ     14$             ;NO-BRANCH
 9153  055206  104401  055336            TYPE    $SPACE2         ;TYPE SEPARATORS
 9154  055212  000771                    BR      11$             ;LOOP
 9155  055214  005302                14$:DEC     R2              ;MORE DH'S?
 9156  055216  003431                    BLE     20$             ;NO-BRANCH
 9157  055220  104401  001205            TYPE    $CRLF
 9158  055224  005760  000002            TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
 9159  055230  001404                    BEQ     15$             ;YES-BRANCH BYPASS INDENT
 9160  055232  005104                    COM     R4              ;INDENT?
 9161  055234  001002                    BNE     15$             ;NO-BRANCH
 9162  055236  104401  055336            TYPE    $SPACE2         ;YES-TYPE SPACES
 9163  055242  012037  055250        15$:MOV     (R0)+,16$       ;GET NEXT DH POINTER
 9164  055246  104401                    TYPE                    ;TYPE DH
 9165  055250  000000                16$:.WORD   0               ;DH POINTER GOES HERE
 9166  055252  105710                    TSTB    (R0)            ;TYPE A DT?
 9167  055254  001003                    BNE     21$             ;YES-BRANCH
 9168  055256  062700  000002            ADD     #2,R0           ;INCREMENT DF POINTER
 9169  055262  000754                    BR      14$             ;SEE IF END OF DF BLOCK
 9170  055264  104401  001205        21$:TYPE    $CRLF
 9171  055270  005704                    TST     R4              ;INDENT?
 9172  055272  001335                    BNE     10$             ;NO-BRANCH
 9173  055274  104401  055336        17$:TYPE    $SPACE2         ;YES-TYPE SPACES
 9174  055300  000732                    BR      10$             ;LOOP
 9175  055302  104414                20$:RESREG
 9176
 9177  055304  032777  010000  123626    BIT     #SW12,@SWR      ;SEE IF ABORT DRV AFTER 20 ERRORS
 9178  055312  001410                    BEQ     25$             ;BR IF NO
 9179  055314  023727  001103  000024    CMP     $ERFLG,#20.     ;ELSE SEE IF HAVE 20 ERRORS
 9180  055322  001004                    BNE     25$             ;BR IF NO
 9181  055324  012706  001100            MOV     #STACK,SP       ;ELSE RESTORE STACK PTR
 9182  055330  000137  031514            JMP     $EOP            ;AND GO TO NEXT DRV
 9183
 9184  055334  000207                25$:RTS     PC
 9185  055336  020040     000    SPACE2: .ASCIZ/  /              ;2 SPACES
 9186                              ;  ODT-11 -- V005A
 9187
 9188                              ; DEC-11-UODPA-A-LA
 9189
 9190                              ; COPYRIGHT 1969,1970,1972
 9191                              ; DIGITAL EQUIPMENT CORPORATION
 9192                              ; MAYNARD, MASSACHUSETTS 01754
 9193                                      .ENABL  ABS,AMA
 9194          055342                      .EVEN
 9195          055422                      .=.+60
 9196          000000              R0      =       %0      ; REGISTER
 9197          000001              R1      =       %1      ;  NAMING
 9198          000002              R2      =       %2      ;   CONVENTIONS
```

```
 9199            000003                  R3      =       %3
 9200            000004                  R4      =       %4
 9201            000005                  R5      =       %5
 9202            000006                  SP      =       %6
 9203            000007                  PC      =       %7
 9204            177776                  ST      =       177776          ;STATUS REGISTER
 9205                                    ;
 9206            000014                  O.TVEC  =       14              ;TRT VECTOR LOCATION
 9207            000340                  O.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
 9208            000020                  O.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
 9209            000003                  TRT     =       000003          ;TRT INSTRUCTION
 9210            000006                  RTT     =       000006          ;RTT INSTRUCTION
 9211                                    ;
 9212                                    ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
 9213                                    ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
 9214                                    ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
 9215                                    ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
 9216                                    ;
 9217            177562                  O.RDB   =       177562  ;R DATA BUFFER
 9218            177560                  O.RCSR  =       177560  ;R C/SR
 9219            177566                  O.TDB   =       177566  ;T DATA BUFFER
 9220            177564                  O.TCSR  =       177564  ;T C/SR
 9221                                    ;
 9222                                    ;
 9223                                    ; INITIALIZE ODT
 9224                                    ;  USE 0.ODT FOR A NORMAL ENTRY
 9225                                    ;  USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
 9226                                    ;  USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
 9227                                    ;
 9228  055422  000413                    O.ODT:  BR      O.STRT          ;NORMAL ENTRY
 9229  055424  000417                            BR      O.RST           ;RESTART
 9230  055426  013737  177776  055402    O.ENTR: MOV     ST,O.UST        ;RE-ENTER  --  SAVE STATUS
 9231  055434  013737  000016  177776            MOV     O.TVEC+2,ST     ;SET UP LOCAL STATUS
Z9232  055442  010737  055400                    MOV     PC,O.UPC        ;FAKE THE PC
 9233  055446  000137  056600                    JMP     O.BK1
 9234                                    ;
 9235  055452  012706  055362            O.STRT: MOV     #O.URO,SP       ;SET UP STACK
 9236  055456  010637  055376                    MOV     SP,O.USP        ;FAKE THE SAVED STACK
 9237  055462  000414                            BR      O.RST1          ;CLEAR BREAKPOINT TABLES
 9238  055464  004037  057006            O.RST:  JSR     0,O.SVR         ;SAVE REGISTERS
 9239  055470  013777  055420  177716            MOV     O.UIN,@O.ADR1   ;REMOVE THE BREAKPOINT
 9240  055476  113704  055404                    MOVB    O.PRI,R4        ;GET ODT PRIORITY
 9241  055502  106004                            RORB    R4              ;SHIFT
 9242  055504  106004                            RORB    R4              ; INTO
 9243  055506  106004                            RORB    R4              ;  POSITION
 9244  055510  110437  177776                    MOVB    R4,ST           ;STORE IN STATUS
Z9245  055514  000127                    O.RST1: JMP     (PC)+
 9246  055516  000403                            BR      O.45
 9247  055520  012737  000002  056510            MOV     #RTI,O.RTIT     ;SET TO RTI IF 11/20 OR /05
 9248  055526  105037  057427            O.45:   CLRB    O.P             ;DISALLOW PROCEED
 9249  055532  012737  000340  000016            MOV     #O.STM,O.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
 9250  055540  012737  056570  000014            MOV     #O.BRK,O.TVEC   ;PC TO TRT VECTOR
 9251  055546  000447                            BR      O.RALL          ;CLEAR BREAKPOINT TABLES
 9252                                    ;
 9253                                    ; SPECIAL NAME HANDLER
 9254                                    ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URO
```

```
                                        ;
055550  004537  057230    O.REGT: JSR     5,O.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
055554  012704  057453            MOV     #O.TL,R4        ;TABLE START ADDRESS
055560  120024            O.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
055562  001413                    BEQ     O.SP            ;JUMP IF YES
055564  022704  057461            CMP     #O.TL+O.LG,R4   ;IS THE SEARCH DONE?
055570  101373                    BHI     O.RSP           ;BRANCH IF NOT
055572  042700  177770            BIC     #177770,R0      ;MASK OFF OCTAL
055576  010004                    MOV     R0,R4
055600  006304            O.SP1:  ASL     R4
055602  062704  055362            ADD     #O.UR0,R4       ;GENERATE ADDRESS
055606  005202                    INC     R2              ;SET FOUND FLAG
055610  000444                    BR      O.SCAN          ;GO FIND NEXT CHARACTER
055612  162704  057444    O.SP:   SUB     #O.TL-7,R4      ;CORRECT CONSTANT
055616  000770                    BR      O.SP1
                                        ;
                                        ; ^ HANDLER - OPEN INDEXED ON THE PC
                                        ;
055620  004737  057354    O.ORPC: JSR     PC,O.TCLS
055624  010502                    MOV     R5,R2           ;CURRENT ADDRESS IN R2
055626  061202                    ADD     @R2,R2          ;COMPUTE
055630  006202                    ASR     R2              ;MOVE ONE BIT TO CARRY
055632  103421                    BCS     O.ERR           ;ERROR IF ODD NUMBER
055634  006302                    ASL     R2              ;RESTORE WORD
055636  005722                    TST     (R2)+           ; AND INCREMENT BY TWO
055640  010205                    MOV     R2,R5           ;UPDATE CAD
055642  000137  056114            JMP     O.OP2           ;GO FINISH UP
                                        ;
                                        ; B HANDLER - SET AND REMOVE BREAKPOINTS
                                        ;
055646  005702            O.BKPT: TST     R2              ;IF NO NUMBER TYPED
055650  001406                    BEQ     O.RALL          ;  REMOVE BREAKPOINT
055652  006204                    ASR     R4              ;CHECK IF ODD
055654  103410                    BCS     O.ERR           ;JUMP IF ODD
055656  006304                    ASL     R4              ;RESTORE ONE BIT
055660  010437  055414            MOV     R4,O.ADR1       ;SET A BREAKPOINT
055664  000412                    BR      O.DCD
055666  012737  057470  055414 O.RALL: MOV  #O.TRTC,O.ADR1 ;CLEAR BREAKPOINT
055674  000406                    BR      O.DCD
                                        ;
                                        ; COMMAND DECODER - ODT11
                                        ;
                                        ;   REGISTERS R0-R4 MAY BE USED,
                                        ;    REGISTER R5 WILL BE CONSIDERED SAFE
                                        ;
055676  052705  000001    O.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
055702  012700  000077            MOV     #'?,R0          ;  ? TO BE TYPED
055706  004537  057306            JSR     5,O.FTYP        ; OUTPUT ?
055712  004537  057406    O.DCD:  JSR     5,O.CRLS        ;TYPE  <CR><LF>*
055716  005004    O.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
055720  005002                    CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
055722  004537  057230    O.SCAN: JSR     5,O.GET         ;GET A CHAR, RETURN IN R0
055726  022700  000060            CMP     #'0,R0          ;COMPARE WITH ASCII 0
055732  101013                    BHI     O.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
055734  022700  000067            CMP     #'7,R0          ;COMPARE WITH ASCII 7
055740  103410                    BLO     O.CLGL          ;CHECK LEGALITY IF NOT OCTAL
```

```
 9311  055742  042700  177770          BIC     #177770,R0      ;CONVERT TO BCD
 9312  055746  006304                  ASL     R4              ; MAKE ROOM
 9313  055750  006304                  ASL     R4              ;   IN
 9314  055752  006304                  ASL     R4              ;     R4
 9315  055754  060004                  ADD     R0,R4           ;PACK THREE BITS IN R4
 9316  055756  005202                  INC     R2              ;R2 HAS NUMERIC FLAG
 9317  055760  000760                  BR      O.SCAN          ;   AND TRY AGAIN
 9318  055762  005001          O.CLGL: CLR     R1              ;CLEAR INDEX
 9319  055764  120061  057437  O.LGL1: CMPB    R0,O.LGCH(R1)   ;DO THE CODES MATCH?
 9320  055770  001405                  BEQ     O.LGL2          ;JUMP IF YES
 9321  055772  005201                  INC     R1              ; SET INDEX FOR NEXT SEARCH
 9322  055774  020127  000014          CMP     R1,#O.CLGT      ;IS THE SEARCH DONE?
 9323  056000  103336                  BHIS    O.ERR           ;   OOPS!
 9324  056002  000770                  BR      O.LGL1          ;RE-LOOP
 9325  056004  006301          O.LGL2: ASL     R1              ;MULTIPLY BY TWO
 9326  056006  000171  056012          JMP     @O.LGDR(R1)     ;GO TO PROPER ROUTINE
 9327                                  ;
 9328  056012  056042          O.LGDR: O.WRD   ;  /    OPEN WORD
 9329  056014  056074                  O.CRET  ;  CARRIAGE RETURN    CLOSE
 9330  056016  055550                  O.REGT  ;  $  REGISTER OPS
 9331  056020  056404                  O.GO    ;  G  GO TO ADDRESS K
 9332  056022  056106                  O.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
 9333  056024  055620                  O.ORPC  ;  ←  OPEN RELATED, INDEX - PC
 9334  056026  056140                  O.BACK  ;  ↑  OPEN PREVIOUS
 9335  056030  056150                  O.OFST  ;  O  OFFSET
 9336  056032  056226                  O.WSCH  ;  W  SEARCH WORD
 9337  056034  056222                  O.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
 9338  056036  055646                  O.BKPT  ;  B  BREAKPOINTS
 9339  056040  056512                  O.PROC  ;  P  PROCEED
 9340          000030          O.LGL   =       .-O.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS
 9341                                  ;
 9342                                  ; PROCESS / - OPEN WORD
 9343                                  ;
 9344  056042  005702          O.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
 9345  056044  001410                  BEQ     O.WRDA          ;SKIP OTHERWISE
 9346  056046  010405                  MOV     R4,R5           ;   PUT VALUE IN CAD
 9347  056050  006205          O.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
 9348  056052  103711          O.ERR2: BCS     O.ERR           ;JUMP IF ODD ADDRESS
 9349  056054  006305                  ASL     R5              ;RESTORE THE CARRY BIT
 9350  056056  011500                  MOV     @R5,R0          ;GET CONTENTS OF WORD
 9351  056060  004537  057144          JSR     5,O.CADV        ;GO GET AND TYPE OUT @CAD
 9352  056064  000714                  BR      O.DCD1          ;GO BACK TO DECODER
 9353  056066  042705  000001  O.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
 9354  056072  000766                  BR      O.WRD1          ;GO BACK TO MAIN-LINE
 9355                                  ;
 9356                                  ; PROCESS CARRIAGE RETURN
 9357                                  ;
 9358  056074  004737  057354  O.CRET: JSR     PC,O.TCLS       ;CLOSE LOCATION
 9359  056100  052705  000001          BIS     #1,R5           ;CLOSE EVERYTHING
 9360  056104  000702                  BR      O.DCD           ;RETURN TO DECODER
 9361                                  ;
 9362                                  ; PROCESS <LF>, OPEN NEXT WORD
 9363                                  ;
 9364  056106  004737  057354  O.OP1:  JSR     PC,O.TCLS       ;CLOSE PRESENT CELL
 9365  056112  005725                  TST     (R5)+           ;GENERATE NEW ADDRESS
 9366  056114  004537  057400  O.OP2:  JSR     5,O.CRLF        ;<CR><LF>
```

```
 9367  056120  010500                        MOV     R5,R0           ;NUMBER TO TYPE
 9368  056122  004537  057144                JSR     5,0.CADV        ; TYPE OUT ADDRESS
 9369  056126  012700  000057                MOV     #'/,R0          ;TYPE A /
 9370  056132  004537  057306                JSR     5,0.FTYP
 9371  056136  000744                        BR      0.WRD1          ;GO PROCESS IT
 9372                                  ;
 9373                                  ; PROCESS ↑, OPEN PREVIOUS WORD
 9374                                  ;
 9375  056140  004737  057354        0.BACK: JSR     PC,0.TCLS
 9376  056144  005745                        TST     -(R5)           ;GENERATE NEW ADDRESS
 9377  056146  000762                        BR      0.OP2           ;GO DO THE REST
 9378                                  ;
 9379                                  ; PROCESS 0, COMPUTE OFFSET
 9380                                  ;
 9381  056150  006205                0.OFST: ASR     R5              ;GET LOW ORDER BIT
 9382  056152  103737                        BCS     0.ERR2          ;ERROR IF CLOSED
 9383  056154  006305                        ASL     R5              ;RESTORE WORD
 9384  056156  012700  000040                MOV     #' ,R0          ;TYPE ONE BLANK
 9385  056162  004537  057306                JSR     5,0.FTYP        ; AS A SEPARATOR
 9386  056166  160504                        SUB     R5,R4           ;COMPUTE
 9387  056170  005304                        DEC     R4
 9388  056172  005304                        DEC     R4              ;     16 BIT OFFSET
 9389  056174  010400                        MOV     R4,R0           ;TYPE A
 9390  056176  010402                        MOV     R4,R2           ;SAVE R4
 9391  056200  004537  057144                JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
 9392  056204  010200                        MOV     R2,R0
 9393  056206  006200                        ASR     R0              ;DIVIDE BY TWO
 9394  056210  103402                        BCS     0.OF1           ;BRANCH IF ODD
 9395  056212  004537  057144                JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
 9396  056216  000137  055716        0.OF1:  JMP     0.DCD1          ;ALL DONE
 9397                                  ;
 9398                                  ; SEARCHES - $MSK   HAS THE MASK
 9399                                  ;            $MSK+2 HAS THE FWA
 9400                                  ;            $MSK+4 HAS THE LWA
```

```
 9401                                   ;
 9402  056222  005201                   O.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
 9403  056224  000401                           BR      O.WDS
 9404  056226  005001                   O.WSCH: CLR     R1              ;SET WORD SEARCH
 9405  056230  005702                   O.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
 9406  056232  001621                   O.ERR1: BEQ     O.ERR           ;ERROR IF NO OBJECT
 9407  056234  013702  055410                   MOV     O.MSK+2,R2      ;SET ORIGIN
 9408  056240  013705  055406                   MOV     O.MSK,R5        ;SET MASK
 9409  056244  005105                           COM     R5              ;AND COMPLEMENT IT
 9410  056246  020237  055412           O.WDS2: CMP     R2,O.MSK+4      ; IS THE SEARCH ALL DONE?
 9411  056252  101217                           BHI     O.DCD           ;  YES
 9412  056254  011200                           MOV     @R2,R0          ; GET OBJECT
 9413  056256  005701                           TST     R1              ;NO
 9414  056260  001027                           BNE     O.EFF1          ;BRANCH IF EFFECTIVE SEARCH
 9415  056262  010046                           MOV     R0,-(SP)
 9416  056264  010403                           MOV     R4,R3           ;EXCLUSIVE OR
 9417  056266  040400                           BIC     R4,R0           ; IS DONE
 9418  056270  042603                           BIC     (SP)+,R3        ;  IN A VERY
 9419  056272  050003                           BIS     R0,R3           ;     FANCY MANNER HERE
 9420  056274  040503                           BIC     R5,R3           ;AND RESULT WITH MASK
 9421  056276  001016                   O.WDS3: BNE     O.WDS4          ;RE-LOOP IF NO MATCH
 9422  056300  010446                           MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
 9423  056302  004537  057400                   JSR     5,O.CRLF        ;TYPE <CR,LF>
 9424  056306  010200                           MOV     R2,R0           ;GET READY TO TYPE
 9425  056310  004537  057144                   JSR     5,O.CADV        ;  TYPE ADDRESS
 9426  056314  012700  000057                   MOV     #'/,R0          ;SLASH TO R0
 9427  056320  004537  057306                   JSR     5,O.FTYP        ;TYPE IT
 9428  056324  011200                           MOV     @R2,R0          ;GET CONTENTS
 9429  056326  004537  057144                   JSR     5,O.CADV        ;TYPE CONTENTS
 9430  056332  012604                           MOV     (SP)+,R4        ; RESTORE R4
 9431  056334  005722                   O.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
 9432  056336  000743                           BR      O.WDS2          ;     RETURN
 9433  056340  020004                   O.EFF1: CMP     R0,R4           ; IS (X)=K?
 9434  056342  001755                           BEQ     O.WDS3          ;TYPE IF EQUAL
 9435  056344  010003                           MOV     R0,R3           ;(X) TO R3
 9436  056346  060203                           ADD     R2,R3           ;(X)+X
 9437  056350  005203                           INC     R3
 9438  056352  005203                           INC     R3              ;(X)+X+2
 9439  056354  020304                           CMP     R3,R4           ;IS (X)+X+2=K?
 9440  056356  001747                           BEQ     O.WDS3          ;BRANCH IF EQUAL
 9441  056360  042700  177400                   BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
 9442  056364  110000                           MOVB    R0,R0           ;EXTEND SIGN
 9443  056366  000257                           CCC
 9444  056370  006300                           ASL     R0              ;MULTIPLY BY TWO
 9445  056372  005200                           INC     R0              ;ADD TWO
 9446  056374  005200                           INC     R0
 9447  056376  060200                           ADD     R2,R0           ;ADD PC
 9448  056400  020004                           CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
 9449  056402  000735                           BR      O.WDS3
 9450                                  ;
 9451                                  ; PROCESS G - GO
 9452                                  ;
 9453  056404  105037  057427          O.GO:   CLRB    O.P             ;DISALLOW PROCEED
 9454  056410  006204                           ASR     R4              ;CHECK LOW ORDER BIT
 9455  056412  103617                           BCS     O.ERR2          ;ERROR IF ODD NUMBER
 9456  056414  006304                           ASL     R4              ;RESTORE WORD
```

```
 9457  056416  010437  055400                 MOV     R4,0.UPC        ;SET UP NEW PC
 9458  056422  112737  000340  177776         MOVB    #0.STM,ST       ;SET HIGH PRIORITY
 9459  056430  004537  057076                 JSR     5,0.RSTT        ;RESTORE TELETYPE
 9460  056434  105037  057426         0.TBIT: CLRB    0.T             ;CLEAR BOTH
 9461  056440  042737  000020  055402         BIC     #0.TBT,0.UST    ;  T-BIT FLAGS
 9462  056446  017737  176742  055420         MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
 9463  056454  013777  057470  176732         MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
 9464  056462  012600                 0.G02:  MOV     (SP)+,R0        ;RESTORE
 9465  056464  012601                         MOV     (SP)+,R1        ;  R0
 9466  056466  012602                         MOV     (SP)+,R2        ;    THRU
 9467  056470  012603                         MOV     (SP)+,R3
 9468  056472  012604                         MOV     (SP)+,R4        ;
 9469  056474  012605                         MOV     (SP)+,R5        ;     R5
 9470  056476  012606                         MOV     (SP)+,SP        ;  AND SP
 9471  056500  013746  055402                 MOV     0.UST,-(SP)     ;     AND STATUS
 9472  056504  013746  055400                 MOV     0.UPC,-(SP)     ;     AND PC
 9473  056510  000006                 0.RTIT: RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
 9474                                 ;
 9475                                 ; PROCESS P - PROCEED
 9476                                 ;   ONLY ALLOWED AFTER A BREAKPOINT
 9477                                 ;
 9478  056512  105737  057427         0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
 9479  056516  001645                         BEQ     0.ERR1          ;NOT LEGAL
 9480  056520  105037  057427                 CLRB    0.P             ;CLEAR PROCEED FLAG
 9481  056524  005702                         TST     R2              ;WAS COUNT SPECIFIED?
 9482  056526  001402                         BEQ     0.PR1           ;NO
 9483  056530  010437  055416                 MOV     R4,0.CT         ;YES, PUT AWAY COUNT
 9484  056534  112737  000340  177776 0.PR1:  MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
 9485  056542  004537  057076                 JSR     5,0.RSTT        ;RESTORE TTY
 9486  056546  112737  000340  177776 0.C1:   MOVB    #0.STM,ST       ;SET HIGH PRIORITY
 9487  056554  105237  057426                 INCB    0.T             ;SET T-BIT FLAG
 9488  056560  052737  000020  055402         BIS     #0.TBT,0.UST    ;SET T-BIT
 9489  056566  000735                         BR      0.G02
 9490                                 ;
 9491                                 ; BREAKPOINT HANDLER
 9492                                 ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
 9493                                 ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
 9494                                 ;  AND GIVES CONTROL TO THE COMMAND DECODER
 9495                                 ;
 9496  056570  012637  055400         0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
 9497  056574  012637  055402                 MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
 9498  056600  004037  057006         0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
 9499  056604  105737  057426                 TSTB    0.T             ;CHECK FOR T-BIT SET
 9500  056610  001311                         BNE     0.TBIT          ;JUMP IF SET
 9501  056612  013777  055420  176574         MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
 9502  056620  105737  055404                 TSTB    0.PRI           ;CHECK IF PRIORITY
 9503  056624  100003                         BPL     0.BK2           ;  IS AS SAME AS USER PGM
 9504  056626  113705  055402                 MOVB    0.UST,R5        ;PICK UP USER UST IF SO
 9505  056632  000407                         BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
 9506  056634  113705  055404         0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
 9507  056640  000257                         CCC                     ;CLEAR CARRY
 9508  056642  106005                         RORB    R5              ;SHIFT LOW ORDER BITS
 9509  056644  106005                         RORB    R5              ;  INTO
 9510  056646  106005                         RORB    R5              ;    HIGH ORDER
 9511  056650  106005                         RORB    R5              ;      POSITION
 9512  056652  110537  177776         0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
```

```
 9513  056656  013705  055400                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
 9514  056662  005745                          TST     -(R5)           ;SUBTRACT TWO
 9515  056664  010537  055400                  MOV     R5,0.UPC        ;FROM THE USER'S PC
 9516  056670  020537  055414                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
 9517  056674  001417                          BEQ     0.B2            ;JUMP IF FOUND
 9518  056676  004537  057044                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
 9519  056702  004537  057400                  JSR     5,0.CRLF
 9520  056706  012704  057432                  MOV     #0.BD,R4        ;ERROR, NOTHING FOUND
 9521  056712  012703  057433                  MOV     #0.BD+1,R3
 9522  056716  004537  057272                  JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
 9523  056722  010500                          MOV     R5,R0
 9524  056724  042737  000020  055402          BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
 9525  056732  000420                          BR      0.B3            ; AND CONTINUE
 9526  056734  005337  055416          0.B2:   DEC     0.CT
 9527  056740  003302                          BGT     0.C1            ;JUMP IF REPEAT
 9528  056742  012737  000001  055416          MOV     #1,0.CT         ;RESET COUNT TO 1
 9529  056750  105237  057427                  INCB    0.P             ;ALLOW PROCEED
 9530  056754  004537  057044                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
 9531  056760  012700  000102                  MOV     #'B,R0
 9532  056764  004537  057306                  JSR     5,0.FTYP        ;TYPE "B"
 9533  056770  013700  055414                  MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
 9534  056774  004537  057144          0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
 9535  057000  005005                          CLR     R5              ;CLEAR CAD
 9536  057002  000137  055712                  JMP     0.DCD           ;GO TO DECODER
 9537                                  ;
 9538                                  ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
 9539                                  ;
 9540  057006  012637  057424          0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
 9541  057012  010637  055376                  MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
 9542  057016  012706  055376                  MOV     #0.USP,SP       ;SET TO INTERNAL STACK
 9543  057022  010546                          MOV     R5,-(SP)        ;SAVE
 9544  057024  010446                          MOV     R4,-(SP)        ; REGISTERS
 9545  057026  010346                          MOV     R3,-(SP)        ;1
 9546  057030  010246                          MOV     R2,-(SP)        ; THRU
 9547  057032  010146                          MOV     R1,-(SP)        ;    5
 9548  057034  013746  057424                  MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
 9549  057040  005746                          TST     -(SP)
 9550  057042  000200                          RTS     R0
 9551                                  ;
 9552                                  ; SAVE TELETYPE STATUS
 9553                                  ;
 9554  057044  113737  177560  057430  0.SVTT: MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
 9555  057052  113737  177564  057431          MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
 9556  057060  105037  177560                  CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
 9557  057064  105037  177564                  CLRB    0.TCSR          ;  BITS IN BOTH C/SR
 9558  057070  004537  057400                  JSR     5,0.CRLF        ;TYPE <CR,LF>
 9559  057074  000205                          RTS     R5
 9560                                  ;
 9561                                  ; RESTORE TELETYPE STATUS
 9562                                  ;
 9563  057076  004537  057400          0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
 9564  057102  105737  177564                  TSTB    0.TCSR          ;WAIT READY ON PRINTER
 9565  057106  100375                          BPL     .-4
 9566  057110  032737  004000  177560          BIT     #4000,0.RCSR    ;CHECK BUSY FLAG ON READER
 9567  057116  001403                          BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
 9568  057120  105737  177560                  TSTB    0.RCSR          ;WAIT READY
```

```
 9569  057124  100375                    BPL     .-4             ;    ON READER
 9570  057126  113737  057430  177560    0.RSE1: MOVB    @0.CSR1,0.RCSR  ;RESTORE
 9571  057134  113737  057431  177564            MOVB    @0.CSR2,0.TCSR  ;  THE STATUS REGISTERS
 9572  057142  000205                      RTS     R5
 9573                                    ;
 9574                                    ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
 9575                                    ; WORD IS IN R0
 9576                                    ;
 9577  057144  010246                    0.CADV: MOV     R2,-(SP)        ;SAVE R2
 9578  057146  012704  057467                    MOV     #0.BUF+6,R4     ;BUFFER START ADDRESS
 9579  057152  012746  000060                    MOV     #'0,-(SP)               ;CONSTANT ASCII 0
 9580  057156  010002                    0.SPC:  MOV     R0,R2           ; GET
 9581  057160  042702  177770                    BIC     #177770,R2      ;    OCTAL CHARACTER
 9582  057164  061602                            ADD     @SP,R2          ;CONVERT TO ASCII
 9583  057166  110244                            MOVB    R2,-(R4)        ;STORE IN BUFFER
 9584  057170  006200                            ASR     R0              ;SHIFT THIS MESS
 9585  057172  006200                            ASR     R0              ; RIGHT
 9586  057174  006200                            ASR     R0              ;   THREE WHOLE PLACES
 9587  057176  020427  057462                    CMP     R4,#0.BUF+1     ;DONE?
 9588  057202  101365                            BHI     0.SPC           ;    NO
 9589  057204  042700  177776                    BIC     #177776,R0      ;GET LAST BIT
 9590  057210  062600                            ADD     (SP)+,R0                ;CONVERT TO ASCII
 9591  057212  110044                            MOVB    R0,-(R4)        ;AND PUT IT AWAY
 9592  057214  012703  057467                    MOV     #0.BUF+6,R3     ;LWA
 9593  057220  004537  057272                    JSR     5,0.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
 9594  057224  012602                            MOV     (SP)+,R2        ;RESTORE R2
 9595  057226  000205                            RTS     R5
 9596                                    ;
 9597                                    ; GENERAL CHARACTER INPUT ROUTINE
 9598                                    ; CHARACTER INPUT GOES TO R0
 9599                                    ;
 9600  057230  105737  177560            0.GET:  TSTB    @0.RCSR         ;WAIT FOR
 9601  057234  100375                            BPL     .-4             ;    INPUT FROM KEYBOARD
 9602  057236  113700  177562                    MOVB    @0.RDB,R0       ;GET A CHARACTER
 9603  057242  004537  057306                    JSR     5,0.FTYP        ;ECHO CHARACTER
 9604  057246  042700  177600                    BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
 9605  057252  001766                            BEQ     0.GET           ;IGNORE NULLS
 9606  057254  122700  000040                    CMPB    #40,R0          ;CHECK FOR SPACES
 9607  057260  001763                            BEQ     0.GET           ;IGNORE NULLS
 9608  057262  122700  000073                    CMPB    #';,R0          ;CHECK FOR SEMI-COLON
 9609  057266  001760                            BEQ     0.GET           ;IGNORE THEM IF FOUND
 9610  057270  000205                            RTS     R5
 9611                                    ;
 9612                                    ; GENERAL CHARACTER OUTPUT ROUTINE
 9613                                    ;  ADDRESS OF FIRST BYTE IN R4,
 9614                                    ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
 9615                                    ;
 9616  057272  020304                    0.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
 9617  057274  103426                            BLO     0.TYP1          ; EXIT WHEN DONE
 9618  057276  112400                            MOVB    (R4)+,R0        ;GET A CHARACTER
 9619  057300  004537  057306                    JSR     5,0.FTYP        ;TYPE ONE CHARACTER
 9620  057304  000772                            BR      0.TYPE          ;LOOP UNTIL DONE
 9621                                    ;
 9622                                    ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
 9623                                    ;
 9624  057306  105737  177564            0.FTYP: TSTB    @0.TCSR         ;CHECK STATUS
```

```
 9625  057312  100375                 BPL     .-4             ;WAIT UNTIL READY
 9626  057314  110037  177566         MOVB    R0,@.TDB        ;TYPE ONE CHARACTER
 9627  057320  120037  000045         CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
 9628  057324  001012                 BNE     0.TYP1          ;NO
 9629  057326  113746  000044         MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
 9630  057332  105737  177564 0.TYP2: TSTB    @.TCSR
 9631  057336  100375                 BPL     0.TYP2
 9632  057340  105037  177566         CLRB    @.TDB           ;GENERATE NULL FILLER
 9633  057344  105316                 DECB    @SP
 9634  057346  003371                 BGT     0.TYP2
 9635  057350  005726                 TST     (SP)+           ;POP STACK
 9636  057352  000205 0.TYP1: RTS     R5
 9637                                 ;
 9638                                 ; CLOSE WORD OR BYTE AND EXIT,
 9639                                 ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
 9640                                 ;
 9641  057354  006205 0.TCLS: ASR     R5              ;GET LOW ORDER BIT
 9642  057356  103405                 BCS     0.TC            ;JUMP IF ALREADY CLOSED
 9643  057360  006305                 ASL     R5
 9644  057362  005702                 TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
 9645  057364  001401                 BEQ     0.CLS1          ;NO CHANGE TO THE OPEN CELL
 9646  057366  010415                 MOV     R4,@R5          ;STORE WORD
 9647  057370  000207 0.CLS1: RTS     PC
 9648  057372  005746 0.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
 9649  057374  000137  055676         JMP     0.ERR           ;AND SCREAM BLOODY MURDER
 9650                                 ;
 9651                                 ; 0.CRLF - TYPE <CR,LF>
 9652                                 ; 0.CRLS - TYPE <CR,LF>*
 9653                                 ;
 9654  057400  012703  057435 0.CRLF: MOV     #0.CR+1,R3      ;LWA <CR,LF>
 9655  057404  000402                 BR      0.CRS
 9656  057406  012703  057436 0.CRLS: MOV     #0.CR+2,R3      ;LWA <CR,LF>*
 9657  057412  012704  057434 0.CRS:  MOV     #0.CR,R4        ;FWA
 9658  057416  004537  057272         JSR     5,0.TYPE        ;TYPE SOMETHING
 9659  057422  000205                 RTS     R5
 9660                                 ;
 9661  057424  000000 0.XXX:  .WORD   0               ;TEMPORARY STORAGE
 9662  057426     000 0.T:    .BYTE   0               ;  T-BIT FLAG
 9663  057427     000 0.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
 9664                                 ;                       = 1 IF PROCEED ALLOWED
 9665  057430     000 0.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
 9666  057431     000 0.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
 9667                                 ;
 9668                                 .EVEN
 9669  057432  042502 0.BD:   .WORD   "BE
 9670                                 ;
 9671  057434     015 0.CR:   .BYTE   015     ;  <CR>
 9672  057435     012                 .BYTE   012     ;  <LF>
 9673  057436     052                 .BYTE   '*      ;  *
 9674                                 ;
 9675  057437     057 0.LGCH: .BYTE   '/      ;  /
 9676  057440     015                 .BYTE   015     ;  CARRIAGE RETURN
 9677  057441     044                 .BYTE   '$      ;  $
 9678  057442     107                 .BYTE   'G      ;  G
 9679  057443     012                 .BYTE   012     ;  <LF>
 9680  057444     137                 .BYTE   '_      ;  ←
```

```
 9681  057445      136                    .BYTE   '↑      ;  ↑
 9682  057446      117                    .BYTE   'O      ;  O
 9683  057447      127                    .BYTE   'W      ;  W
 9684  057450      105                    .BYTE   'E      ;  E
 9685  057451      102                    .BYTE   'B      ;  B
 9686  057452      120                    .BYTE   'P      ;  P
 9687              000014          O.CLGT  =       .-O.LGCH                ;TABLE LENGTH
 9688                              ;
 9689  057453      123             O.TL:   .BYTE   'S      ;DO        1
 9690  057454      120                     .BYTE   'P      ;NOT       2
 9691  057455      115                     .BYTE   'M      ;CHANGE    3
 9692  057456      000                     .BYTE   0       ;THE       4
 9693  057457      000                     .BYTE   0       ;ORDER     5
 9694  057460      102                     .BYTE   'B      ;HERE      6
 9695              000006          O.LG    =       .-O.TL
 9696                              ;
 9697  057461                      O.BUF:                                  ;6 CHAR. BUFFER WITH
 9698              057467          .       =       .+6
 9699  057467      040                     .BYTE   ' '                     ;TRAILING BLANK
 9700                                      .EVEN
 9701                              ;
 9702  057470  000003              O.TRTC: TRT                             ;TRACE TRAP PROTOTYPE
 9703                              ;
 9704                              ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
 9705                              ;
 9706              055362          .       =       O.ODT-40
 9707  055362      000000          O.URO:  0       ;USER R0
 9708  055364      000000                  0       ;       R1
 9709  055366      000000                  0       ;       R2
 9710  055370      000000                  0       ;       R3
 9711  055372      000000                  0       ;       R4
 9712  055374      000000                  0       ;       R5
 9713  055376      000000          O.USP:  0       ;USER SP
 9714  055400      000000          O.UPC:  0       ;USER PC
 9715  055402      000000          O.UST:  0       ;USER ST
 9716  055404      000007          O.PRI:  7       ;ODT PRIORITY
 9717  055406      000000          O.MSK:  0       ;MASK
 9718  055410      000000                  0       ;LOW LIMIT
 9719  055412      000000                  0       ;HIGH LIMIT
 9720                              ;
 9721                              ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
 9722                              ;   UIN = CONTENTS
 9723                              ;
 9724  055414      000000          O.ADR1: 0
 9725  055416      000000          O.CT:   0
 9726  055420      000000          O.UIN:  0
 9727              000001                  .END
```

```
ABASE = 177440          1900    1941    1955#
ACDW1 = 000000          1900    1943
ACDW2 = 000000          1900    1944
ACLO  = 000010          1240#
ACPUOP= 000000          1900    1915
ACT11   007454          2126#   3301#
ADDW0 = 000000          1900    1945
ADDW1 = 000000          1900    1946
ADDW10= 000000          1900
ADDW11= 000000          1900
ADDW12= 000000          1900
ADDW13= 000000          1900
ADDW14= 000000          1900
ADDW15= 000000          1900
ADDW2 = 000000          1900    1947
ADDW3 = 000000          1900    1948
ADDW4 = 000000          1900    1949
ADDW5 = 000000          1900    1950
ADDW6 = 000000          1900    1951
ADDW7 = 000000          1900    1952
ADDW8 = 000000          1900
ADDW9 = 000000          1900
ADEVCT= 000000          1900    1906
ADEVM = 000000          1900    1942
AENV  = 000000          1900    1911
AENVM = 000000          1900    1912
AFATAL= 000000          1900    1903
AMADR1= 000000          1900    1928
AMADR2= 000000          1900    1932
AMADR3= 000000          1900    1935
AMADR4= 000000          1900    1938
AMAMS1= 000000          1900    1922
AMAMS2= 000000          1900    1930
AMAMS3= 000000          1900    1933
AMAMS4= 000000          1900    1936
AMSGAD= 000000          1900    1908
AMSGLG= 000000          1900    1909
AMSGTY= 000000          1900    1902
AMTYP1= 000000          1900    1923
AMTYP2= 000000          1900    1931
AMTYP3= 000000          1900    1934
AMTYP4= 000000          1900    1937
APASS = 000000          1900    1905
APRIOR= 000000          1900
APTCSU= 000040          7239    7411#
APTENV= 000001          7186    7232    7367    7409#
APTSIZ= 000200          3230    7408#
APTSPO= 000100          7234    7369    7410#
ASWREG= 000000          1900    1913
ATESTN= 000000          1900    1904
ATTN    007330          2059#   6141    6160    6186
AUNIT = 000000          1900    1907
AUSWR = 000000          1900    1914
AVECT1= 000000          1900    1939
AVECT2= 000000          1900    1940
BADHDR  007324          2049#   3309#   3978#   4112#   4383#   4582#   5692#   5912#   5915#   6891
```

```
BADTMO  036522          3382    6949#
BAI   = 000020          1203#   4123    4274    4416    4489    4898    4959    5087    5256    5331    5451    5505    5630
                        5762    5808
BA16  = 000400          1189#
BA17  = 001000          1190#
BIT0  = 000001          1132#   1186    1218    1237    2116    3421    6835
BIT00 = 000001          1122#   1132
BIT01 = 000002          1121#   1131
BIT02 = 000004          1120#   1130
BIT03 = 000010          1119#   1129
BIT04 = 000020          1118#   1128
BIT05 = 000040          1117#   1127
BIT06 = 000100          1116#   1126
BIT07 = 000200          1115#   1125
BIT08 = 000400          1114#   1124    7121
BIT09 = 001000          1113#   1123    7129    7197
BIT1  = 000002          1131#   1219    2117
BIT10 = 002000          1112#   1191    1209    1228    1260    1274    1288    1301    1315    7174
BIT11 = 004000          1111#   1192    1210    1229    1246    1261    1275    1289    1302    1316    7136
BIT12 = 010000          1110#   1193    1211    1230    1262    1276    1290    1303    1317
BIT13 = 020000          1109#   1194    1212    1231    1247    1263    1277    1291    1304    1318    7181
BIT14 = 040000          1108#   1195    1213    1232    1248    1264    1278    1292    1305    1319    1332    6637    6648
                        7107
BIT15 = 100000          1107#   1196    1197    1214    1233    1249    1265    1335    5694    5857    6633    6644
BIT2  = 000004          1130#   1220    1239    2118
BIT3  = 000010          1129#   1202    1221    1240
BIT4  = 000020          1128#   1203    1222    1241    1254    1282    1309
BIT5  = 000040          1127#   1204    1223    1242    1255    1269    1283    1296    1310
BIT6  = 000100          1126#   1187    1205    1224    1243    1256    1270    1284    1297    1311
BIT7  = 000200          1125#   1188    1206    1225    1244    1257    1271    1285    1298    1312    3411    4654    4698
                        4750    4849    5112
BIT8  = 000400          1124#   1189    1207    1226    1245    1258    1272    1286    1299    1313
BIT9  = 001000          1123#   1190    1208    1227    1259    1273    1287    1300    1314
BPTVEC= 000014          1139#
BSE   = 000200          1225#   4025    4134    4199    4431    4911    5267    5462
BSERR   001476          2032#   3886*   3891*   3969
BSE20H  002322          2041#   3907    6642    6747
BSE20S  004322          2043#   3916    6646    6751
BSE22H  003322          2042#   3826    3888    3925    6631    6736
BSE22S  005322          2044#   3898    6635    6740
BYP     032736          3491    3514    3518    3533    3585    3593    3611    3615    3619    3623    6213#
BYPCER  001502          2034#   3408*   3637*   6327
CALADD  001362          1982#   4094*   4386*   4564*   5866*   6604    6610    6612    6666
CALDIF  001354          1979#   5755*   8888    8912
CCLR  = 100000          1197#   3787    3805    4035    4144    4441    4668    4679    4921    5075    5145    5163    5241
                        5277    5392    5410    5563    5581    6368    6378    6388    6398    6895    6913
CCYL    001350          1977#
CDT   = 002000          1191#
CERR  = 100000          1196#   3471    3571    3838    3944    3997    4022    4105    4131    4197    4284    4400    4428
                        4500    4575    4721    4778    4842    4908    4968    5264    5340    5459    5513    5645    5718
                        5741    5775    5821    5877    5902    6329    6424
CFMT  = 010000          1193#   4392    4396    4420    4424    4463    4492    4496    4529    6733
CHKFLG  001504          2035#   6223*   6305    6308    6312
CHKMSG  032752          3850    4547    4066    4156    4175    4223    4240    4303    4322    4357    4453    4481    4519
                        4547    4619    4640    4690    4742    4805    4933    4952    4987    5006    5046    5067    5212
                        5233    5289    5308    5359    5378    5477    5496    5532    5551    5657    5789    5835    6223#
```

```
CKCERR  033504          6091  6095  6106  6111  6327#
CKSWR = 104407          7106  7170  7196  8071#
CLEAR = 000005          1174#  3807  4670  5077  5165  5243  5412  5583  6915
CLKOF   036044          6806#
CLKON   035750          6784#
CLOCK   036010          3331  6795#
CLRFLG  031672          3251  5968#
COE   = 001000          1227#
COUNT   001366          1986#  6796*  6798*
CR    = 000015          1047#  7278  7288  8083  8088  8093  8098  8108  8117  8124  8132  8137  8142
                        8146  8152  8158  8164  8171  8176  8183  8192  8197  8202  8208  8210  8214
                        8220  8228  8238  8243  8249  8253  8255  8259  8264  8272  8277  8285  8294
                        8300  8308  8314  8321  8326  8330  8336  8354  8362  8441  8468  8475  8480
                        8485  8497  8518
CRLF  = 000200          1048#  7249  7288
CTO   = 004000          1192#  6333
CYL     001422          2003#  5444
CYLADD  001360          1981#  4647  4822  5318  5795  5841  6480*  6481*  6482*  6483*  6484*  6485*  6503*
                        6504*  6505*  6506*  6507*  6508*
CYLDIF  001356          1980#  4657  4662  4701  4706  4753  4758  4826  5315  6447*  6448*  6449*  6450*
                        6451*  6452*  6453  6455*  6471*  6472*  6490*  6491*  6492*  6493*  6494*  6495*  6496
                        6498*
DATAO   001460          2023#  4121  4122  4258  4335  5255  5330  5345
DATAO1  001462          2024#  5629
DATA1   001464          2025#  4013  4262  4263  4275  4289  4415  4488  4505  4897  4958  4973  5086
                        5761  5807
DCK   = 100000          1233#  4206
DCLO  = 000020          1241#
DCPAR = 020000          1194#
DDISP = 177570          1054#  1875  3218
DDPCH   007452          2125#  3297*  3479  3666
DDT   = 000400          1245#
DDUMP   007450          2124#  3264*  5968
DF1     054456          2166  2172  2178  2184  2189  2195  2201  2207  2433  2473  2478  2518  8964#
DF10    054606          2212  2217  2222  2227  2232  2242  2252  2257  2267  2283  2303  2308  2343
                        2353  2358  2393  2423  2428  2468  2498  2503  2558  2563  2568  2573  2588
                        2613  2618  2643  2693  2708  2794  2814  2819  2869  2939  2974  3010  9015#
DF14    054626          2839  9024#
DF15    054646          2523  2804  2809  2829  2834  2944  9034#
DF17    054666          2949  2954  2964  2969  9043#
DF2     054472          2333  8971#
DF20    054712          2373  9053#
DF21    054736          2262  2272  2277  2288  2293  2298  2313  2318  2323  2328  2378  2383  2388
                        2398  2403  2408  2413  2418  2648  2653  2658  2663  2914  2919  3070  3075
                        3095  3100  3105  3110  3115  3120  3125  3130  3135  3140  3145  3150  3155
                        3160  9064#
DF22    054772          2338  2363  2368  2493  2508  2513  2548  2553  2578  2583  9079#
DF23    055026          9094#
DF3     054476          2237  2247  8974#
DF4     054522          2348  8985#
DF5     054536          2593  2598  2603  8992#
DF6     054562          2538  2543  2864  3005  9003#
DH1     051572          2164  2170  2176  2182  2187  2193  2199  2205  2346  2431  2471  2476  2516
                        2837  8678#  8976  8996  9005  9017  9036  9047  9055  9066  9081  9096
DH10    052057          2466  2812  2817  8711#
DH11    052106          8715#  8978
```

```
DH13    052123          2947    8718#
DH14    052207          2952    8727#
DH17    052273          2361    2366    2391    2586    2912    2917    3133    3138    8736#
DH19    052317          2281    2556    8740#
DH2     051607          8681#   8966    8980    8987    8998    9009    9019    9028    9038    9049    9059    9074    9089
                        9104
DH20    052342          2561    8744#
DH21    052375          2265    2566    8749#
DH22    052423          2341    2551    2691    2706    3093    3098    3123    3128    8753#
DH24    052455          2301    2306    2321    2326    2336    2546    3068    3073    8758#
DH25    052502          2371    2611    2616    2641    2862    2867    3003    8762#
DH26    052525          2220    2225    2245    2250    2275    2286    2386    2396    2937    2942    8766#
DH27    052555          2210    2215    2260    2270    2356    2376    2381    2421    2496    2521    2536    2541    8771#
DH28    052606          8776#   9068    9083    9098
DH29    052616          8778#   9072    9087    9102
DH3     051657          8688#   8968    8982    8989    9000    9011    9021    9030    9040    9051    9061    9076    9091
                        9106
DH30    052626          2576    2792    2802    2807    3113    3118    3153    3158    8780#
DH31    052660          8785#   9070    9085    9100
DH32    052705          2230    2235    2255    2291    2296    2401    2406    2501    8789#
DH39    052737          2827    2832    3103    3108    3143    3148    8794#
DH40    052772          8799#   9026
DH41    053026          3008    8804#
DH42    053053          2962    8808#
DH43    053124          2967    8815#
DH44    053175          2972    8822#   9045
DH47    053254          2491    8830#
DH49    053324          8837#   8994
DH51    053401          2351    2581    2646    2651    2656    2661    8845#
DH52    053434          2240    8850#
DH53    053470          2311    2316    2411    2416    8855#
DH54    053516          2426    8859#
DH56    053570          8866#   9057
DH57    053614          2506    2511    8870#
DH6     051730          8695#   9007
DH8     051763          2331    8700#
DH9     052004          2571    8703#
DI      =   040000      1195#
DISPLA  001142          1875#   3218*   3226*   7151*   7173*
DISPRE  000174          1344#   3226
DLT     =   100000      1214#   4789
DLY     032720          5636    6204#   6207
DMD     =   000040      1255#
DOTIM   007506          2144#   3330*   3339*   5617
DPAT1   001466          2026#   5450    5504    5518
DPAT2   001470          2027#
DRA     =   000001      1237#   3508    3601
DRDY    =   000200      1244#
DRIVS   007460          2128#   3416*   3423*   3440    3455*   3481*   3498    3589*   3655    5931    6023*
DRIVO   007462          2133#   3312    3418    3457    3557    6006
DRIV1   007464          2134#
DRIV2   007466          2135#
DRIV3   007470          2136#
DRIV4   007472          2137#
DRIV5   007474          2138#
DRIV6   007476          2139#
```

```
DRIV7   007500          2140#
DROT  = 000040          1242#
DRPAR = 000010          1221#
DRVMSK= 000007          1201#   3474    3574
DRVPTR  001342          1973#   3312*   3660    3670*
DSC   = 040000          1248#
DSWR  = 177570          1053#   1874    3217
DTE   = 010000          1230#
DTYE  = 000040          1223#
DT1     053654          2165    2171    2177    2183    2188    2194    2200    2206    2211    2216    2221    2226    2231
                        2241    2251    2256    2266    2282    2302    2307    2342    2347    2352    2357    2392    2422
                        2427    2432    2467    2472    2477    2497    2502    2517    2522    2557    2562    2567    2572
                        2587    2592    2597    2602    2612    2617    2642    2692    2707    2793    2803    2808    2813
                        2819    2828    2833    2868    2938    2943    2948    2953    2963    2968    2973    3009    8880#
DT13    054212          2261    2271    2276    2287    2292    2297    2312    2317    2322    2327    2377    2382    2387
                        2397    2402    2407    2412    2417    2647    2652    2657    2662    2913    2918    3069    3074
                        3094    3099    3104    3109    3114    3119    3124    3129    3134    3139    3144    3149    3154
                        3159    8928#
DT14    054272          2337    2362    2367    2492    2507    2512    2547    2552    2577    2582    8938#
DT15    054362          8949#
DT3     053714          2332    8887#
DT4     053720          2537    2542    2863    8888#
DT6     053766          2236    2246    8896#
DT7     054032          2838    8904#
DT8     054100          3004    8912#
DT9     054146          2372    8920#
D.ACLO= 000100          1297#
D.BRHM= 000100          1284#   3844    4041    4060    4150    4169    4217    4234    4297    4316    4351    4447    4475
                        4513    4541    4616    4634    4687    4736    4799    4927    4946    4981    5000    5043    5061
                        5209    5227    5283    5302    5353    5372    5471    5490    5526    5545    5651    5783    5829
D.CART= 000400          1286#   3844    4041    4060    4150    4169    4217    4234    4297    4316    4351    4447    4475
                        4513    4541    4616    4634    4687    4736    4799    4927    4946    4981    5000    5043    5061
                        5209    5227    5283    5302    5353    5372    5471    5490    5526    5545    5651    5783    5829
D.DDT = 000400          1272#
D.DOOR= 000200          1285#   3844    4041    4060    4150    4169    4217    4234    4297    4316    4351    4447    4475
                        4513    4541    4616    4634    4687    4736    4799    4927    4946    4981    5000    5043    5061
                        5209    5227    5283    5302    5353    5372    5471    5490    5526    5545    5651    5783    5829
D.DRA = 000040          1269#   3842    4039    4058    4148    4167    4215    4232    4295    4314    4349    4445    4473
                        4511    4539    4614    4632    4685    4734    4797    4925    4944    4979    4998    5041    5059
                        5207    5225    5281    5300    5351    5370    5469    5488    5524    5543    5649    5781    5827
D.DRDY= 000200          1271#   3842    4039    4058    4148    4167    4215    4232    4295    4314    4349    4445    4473
                        4511    4539    4632    4685    4734    4797    4925    4944    4979    4998    5059    5225    5281
                        5300    5351    5370    5469    5488    5524    5543    5649    5781    5827
D.DROT= 020000          1304#   5771    5817
D.DSC = 040000          1278#   4632    4734    5059    5225
D.FLT = 000200          1298#
D.FORM= 001000          1273#   4467    4533
D.FWD = 002000          1288#
D.HDFL= 000200          1312#
D.HDHM= 000040          1283#   6560
D.IDAE= 000040          1296#
D.ILF = 000400          1299#
D.LIMD= 020000          1318#
D.LOAD= 010000          1290#   6577
D.MHD = 000400          1313#
D.NMOV= 010000          1317#
```

```
D.OFF = 002000          1274#  4614   4632   4681   4685   4728   4734   4818   5041   5059   5207   5225
D.PAR = 001000          1300#
D.PIP = 020000          1277#  4614   5041   5207
D.PLO = 004000          1316#
D.REV = 004000          1289#
D.RTZ = 020000          1291#  3798   5156   5403   5574   6906
D.SECT= 000020          1309#
D.SKI = 002000          1301#
D.SPIN= 010000          1276#  3842   4039   4058   4148   4167   4215   4232   4295   4314   4349   4445   4473
                        4511   4539   4614   4632   4685   4734   4797   4925   4944   4979   4998   5041   5059
                        5207   5225   5281   5300   5351   5370   5469   5488   5524   5543   5649   5781   5827
D.SPLS= 010000          1303#
D.SPOK= 001000          1287#  3844   4041   4060   4150   4169   4217   4234   4297   4316   4351   4447   4475
                        4513   4541   4616   4634   4687   4736   4799   4927   4946   4981   5000   5043   5061
                        5209   5227   5283   5302   5353   5372   5471   5490   5526   5545   5651   5783   5829
D.SSP = 000020          1282#  3844   4041   4060   4150   4169   4217   4234   4297   4316   4351   4447   4475
                        4513   4541   4616   4634   4687   4736   4799   4927   4946   4981   5000   5043   5061
                        5209   5227   5283   5302   5353   5372   5471   5490   5526   5545   5651   5783   5829
D.SUNS= 040000          1319#
D.TIB = 002000          1315#
D.UNLD= 040000          1292#
D.UNS = 040000          1305#
D.VV  = 000100          1270#  3736   3750   3842   4039   4058   4148   4167   4215   4232   4295   4314   4349
                        4445   4473   4511   4539   4614   4632   4685   4734   4797   4925   4944   4979   4998
                        5041   5059   5207   5225   5281   5300   5351   5370   5469   5488   5524   5543   5649
                        5781   5827
D.WCUR= 000040          1310#
D.WGAT= 000100          1311#
D.WLE = 004000          1302#
D.WRL = 004000          1275#
D.XERR= 001000          1314#
ECCW  = 020000          1263#
ECH   = 000100          1224#
EMTVEC= 000030          1142#  3201*  3202*
EM1     046062          6032   8336#
EM10    046636          2198   8402#
EM100   051503          2535   8665#
EM101   051540          2575   8670#
EM11    046721          2204   8411#
EM12    046765          2209   2219   2229   2300   2555   2560   2565   2570   2585   2610   2690   2791   2836
                        2936   8418#
EM13    047023          2305   2390   2465   2615   8424#
EM14    047045          2330   8428#
EM15    047103          2345   8434#
EM16    047212          2946   2951   2961   2966   8446#
EM17    047255          2310   2375   2385   2400   2645   2911   3067   3122   3142   3152   8452#
EM18    047276          2259   2274   2290   2410   2650   3072   3092   3102   3112   3132   8455#
EM19    047317          2315   2320   2380   2395   2405   2655   2916   3127   3147   3157   8458#
EM2     046136          2163   8344#
EM20    047340          2269   2285   2295   2325   2415   2660   3097   3107   3117   3137   8461#
EM21    047361          2214   2224   2254   2264   2806   2831   2866   2941   8464#
EM22    047403          2470   8468#
EM23    047510          2475   8480#
EM24    047610          2280   8492#
EM25    047643          2355   8497#
EM26    047715          2420   8505#
```

```
EM27    047774          2430    8513#
EM3     046226          6953    8354#
EM36    050113          2540    2861    3002    8527#
EM39    050156          2360    2505    2640    2811    8533#
EM4     046301          8362#
EM40    050225          2365    2510    2816    8540#
EM41    050263          2335    8546#
EM5     046355          2169    2600    8370#
EM55    050331          2705    8553#
EM6     046376          2175    8373#
EM63    050364          2801    8558#
EM65    050405          2836    8561#
EM69    050427          2971    8565#
EM7     046417          2181    8376#
EM73    050460          2590    8570#
EM74    050501          3007    8573#
EM79    050526          2595    8577#
EM8     046506          2186    8386#
EM80    050547          2234    8580#
EM81    050606          2239    8586#
EM82    050652          2244    8592#
EM83    050724          2249    8599#
EM84    050761          2591    2596    2601    8604#
EM85    051052          2340    2350    8614#
EM86    051115          2545    2550    2580    8620#
EM88    051153          2425    8626#
EM9     046562          2192    8394#
EM93    051214          2370    8632#
EM94    051275          2490    8641#
EM95    051331          2495    2500    8646#
EM96    051365          2515    8651#
EM97    051416          2520    8656#
ERRVEC= 000004          1135#   3215    3216*   3227*   3234*   3235*   3248*   3249*   3315*   3319*   3326*   3364*   3382*
                        7112    7113*   7115*   7118*
ESEC    001400          1992#
E.A0    007430          2105#   3842*   4039*   4058*   4148*   4167*   4215*   4232*   4295*   4314*   4349*   4445*   4473*
                        4511*   4539*   4614*   4632*   4685*   4734*   4797*   4925*   4944*   4979*   4998*   5041*   5059*
                        5207*   5225*   5281*   5300*   5351*   5370*   5469*   5488*   5524*   5543*   5649*   5781*   5827*
                        6227*   6234    6236*   6265    8928    8938    8949
E.A1    007434          2107#   3844*   4041*   4060*   4150*   4169*   4217*   4234*   4297*   4316*   4351*   4447*   4475*
                        4513*   4541*   4616*   4634*   4687*   4736*   4799*   4927*   4946*   4981*   5000*   5043*   5061*
                        5209*   5227*   5283*   5302*   5353*   5372*   5471*   5490*   5526*   5545*   5651*   5783*   5829*
                        6228*   6238    6240*   6285    8928    8938    8949
E.A2    007440          2109#   3846*   4043*   4062*   4152*   4171*   4219*   4236*   4299*   4318*   4353*   4449*   4477*
                        4515*   4543*   4636*   4738*   4801*   4929*   4948*   4983*   5002*   5063*   5229*   5285*   5304*
                        5355*   5374*   5473*   5492*   5528*   5547*   5653*   5785*   5831*   6229*   6242    6244*   8938
                        8949
E.A3    007444          2111#   6230*
E.B0    007432          2106#   3843*   4040*   4059*   4149*   4168*   4216*   4233*   4296*   4315*   4350*   4446*   4474*
                        4512*   4540*   4615*   4633*   4686*   4735*   4798*   4926*   4945*   4980*   4999*   5042*   5060*
                        5208*   5226*   5282*   5301*   5352*   5371*   5470*   5489*   5525*   5544*   5650*   5782*   5828*
                        6246    6248*   6275    8928    8938    8949
E.B1    007436          2108#   3845*   4042*   4061*   4151*   4170*   4218*   4235*   4298*   4317*   4352*   4448*   4476*
                        4514*   4542*   4617*   4635*   4688*   4737*   4800*   4928*   4947*   4982*   5001*   5044*   5062*
                        5210*   5228*   5284*   5303*   5354*   5373*   5472*   5491*   5527*   5546*   5652*   5784*   5830*
                        6250    6252*   6295    8928    8938    8949
```

```
E.B2    007442          2110#  3847#  4044#  4063#  4153#  4172#  4220#  4237#  4300#  4319#  4354#  4450#  4478#
                        4516#  4544#  4637#  4739#  4802#  4930#  4949#  4984#  5003#  5064#  5230#  5286#  5305#
                        5356#  5375#  5474#  5493#  5529#  5548#  5654#  5786#  5832#  6254   6256#  8938   8949
E.B3    007446          2112#  3848#  4045#  4064#  4154#  4173#  4221#  4238#  4301#  4320#  4355#  4451#  4479#
                        4517#  4545#  4638#  4740#  4803#  4931#  4950#  4985#  5004#  5065#  5231#  5287#  5306#
                        5357#  5376#  5475#  5494#  5530#  5549#  5655#  5787#  5833#  6258   6260#  8949
FATT1   032544          3802   5160   5407   5578   6157#  6885   6910
FATT2   032640          3941   4628   4718   4839   5055   5221   5738   5899   6184#
FHDHM   035000          6557#  6565
FHDTAB  035122          4097   4389   4567   5869   6590#
FLGTST  035402          6632   6636   6643   6647   6659#
FLOAD   035054          6574#
FMTE  = 000020          1222#
FMT1    001454          2020#  6600#  6601#  6602#  6607
FORM    031226          5747   5853   5856#  6925
FORMAT  001452          2019#  4096#  4388#  4566#  5868#  6600   6629
FRCYL   001344          1975#  5750#  8888   8912
FRDY    032230          3450   3464   3469   3564   3569   3747   3793   3809   3835   3937   3994   4019   4102
                        4128   4194   4281   4340   4394   4398   4422   4426   4465   4494   4498   4531   4572
                        4611   4672   4714   4776   4835   4905   4965   5038   5079   5092   5151   5167   5204
                        5245   5261   5337   5398   5414   5456   5510   5569   5585   5640   5715   5734   5767
                        5813   5874   5895   6086#  6089   6358   6419   6881   6901   6917
FRDY1   032276          6100#  6103   6373   6383   6393   6402
FSEC23  034710          5625   6533#
FTITLE  001340          1971#  5979   5981*
GBA     032132          3283   6046#
GDRVS   031772          3281   6001#
GINT    032160          3285   6059#
GNS   = ****** U        1343   8063   8064   8065   8066   8067   8069   8071   8072   8073   8074   8075   8076
                        8077
GO    = 000001          1186#
GSTAT   033566          3700   3797   3837   3996   4021   4104   4130   4196   4283   4342   4574   4680   4791
                        4907   4967   5094   5155   5263   5339   5402   5458   5512   5573   5642   5717   5770
                        5816   5876   6167   6171   6191   6195   6354#  6423   6434   6446   6479   6559   6576
                        6905
GSTAT1  033630          6225   6366#
GTSWR = 104406          5990   8069#
HASOF   007356          2076#  6125*  6141   8884   8893   8901   8909   8917   8925   8935   8946   8958
HBA     007346          2072#  6121*  8884   8893   8901   8909   8917   8925   8935   8946   8958
HCS1    007340          2069#  3471   3571   3838   3944   3997   4022   4105   4131   4197   4284   4400   4428
                        4500   4575   4721   4778   4842   4908   4968   5264   5340   5459   5513   5645   5718
                        5741   5775   5821   5877   5902   6118#  6329   6333   6424   6733   8881   8890   8898
                        8906   8914   8922   8932   8943   8955
HCS2    007342          2070#  3504   3506   3510   3522   3597   3599   3603   3606   4286   4343   4502   4970
                        5095   5342   5515   6119#  6338   6342   8881   8890   8898   8906   8914   8922   8932
                        8943   8955
HDA     007350          2073#  6122*  6776   8884   8893   8901   8909   8917   8925   8935   8946   8958
HDB     007362          2079#
HDC     007360          2077#  6126*  6768   8884   8893   8901   8909   8917   8925   8935   8946   8958
HDS     007352          2074#  3508   3601   6123*  8881   8890   8898   8906   8914   8922   8932   8943   8955
HDTAB   001506          2037#  3980   3986   3989   4090   4380   4559   5699   5706   5709   5862   6592
HDWD    001474          2030#  8904
HDI     001450          2018#  6594#  6595#  6596#  6597#  6598#  6599#  6606
HEAD    001444          2016#  4095#  4387#  4565#  5867#  6594   6671
HEADA   001446          2017#  6523#  6524#  6525#  6526#
HER     007354          2075#  4025   4134   4199   4206   4431   4911   5267   5462   6124#  8881   8890   8898
```

```
                        8906    8914    8922    8932    8943    8955
HMR1    007364          2079#   6127#   8881    8890    8898    8906    8914    8922    8932    8943    8955
HMR2    007366          2080#   3473    3573    3701    3736    3750    3798    4467    4533    4681    4728    4818    5156
                        5403    5574    6104#   6109#   6128#   6375    6385    6395    6404    6447    6560    6577    6906
                        8881    8890    8898    8906    8914    8922    8932    8943    8955
HMR3    007370          2081#   5771    5817    6105#   6110#   6129#   6376    6386    6396    6405    6435    6480    8881
                        8890    8898    8906    8914    8922    8932    8943    8955
HOLD    032364          6090    6094    6118#   6367
HPAT    007374          2083#   6131#   8884    8893    8901    8909    8917    8925    8935    8946    8958
HPEND   007326          2051#   3310#   5909    5911#   6893#
HPOS    007372          2082#   6130#   8884    8893    8901    8909    8917    8925    8935    8946    8958
HT    = 000011          1045#   7247    7288
HVRC  = 000400          1226#
HWC     007344          2071#   6120#   8884    8893    8901    8909    8917    8925    8935    8946    8958
HZ      001364          1984#   6798
H.A0    007410          2094#   6265    6404#   8928    8941    8952
H.A1    007414          2096#   6285    6395#   8928    8941    8952
H.A2    007420          2098#   6385#   6490    8941    8952
H.A3    007424          2100#   6375#
H.B0    007412          2095#   6275    6405#   8928    8941    8952
H.B1    007416          2097#   6295    6396#   8928    8941    8952
H.B2    007422          2099#   6386#   6503    8941    8952
H.B3    007426          2101#   6376#   6513    6523    8952
IDAE  = 002000          1228#
IE    = 000100          1187#
ILF   = 000001          1218#
INTER   036652          6074    6993#
IOTVEC= 000020          1140#   3199#   3200#
IR    = 000100          1205#
LCLKF   007502          2142#   3317#
LCVEC   001330          1963#   3318
LF    = 000012          1046#   7282    7288    8083    8088    8093    8098    8108    8117    8124    8132    8137    8142
                        8146    8152    8158    8164    8171    8176    8183    8192    8197    8202    8208    8210    8214
                        8220    8228    8238    8243    8249    8253    8255    8259    8264    8272    8277    8285    8294
                        8300    8308    8314    8321    8326    8330    8336    8354    8362    8441    8468    8475    8480
                        8485    8497    8518
LIMERR  001500          2033#
LKS     001326          1961#   3316    6787#   6808#
MCLK  = 000400          1258#
MDS   = 001000          1208#   3504    3597    6342
MEMBAS= 172100          1967#   3237    3246
MEMERR  036610          3240    6979#
MEMPAR  013042          3234#
MEMVEC= 000114          1966#   3240#   3241#
MERD  = 001000          1259#
MESMSK= 000017          1253#
MEWD  = 002000          1260#
MIND  = 000200          1257#
MSG1    043352          5982    8083#
MSG10   044636          3415    3454    8210#
MSG11   044663          7026    8214#
MSG12   044723          5179    8220#
MSG13   045070          3340    8238#
MSG14   045166          6213    8249#
MSG15   045211          3524    3671    8253#
MSG16   045223          3698    8255#
```

```
MSG17   045247            3924    8259#
MSG2    044242            3265    8164#
MSG26   045277            3972    4000    4053    4108    4162    4203    4309    4403    4459    4525    4578    4781    4793
                          4939    4993    5295    5365    5483    5538    5721    5880    8264#
MSG27   045355            3657    8272#
MSG3    044313            3280    8171#
MSG37   045405            5127    8277#
MSG38   045463            5140    8285#
MSG39   045543            5103    8294#
MSG4    044345            3282    8176#
MSG40   045605            3971    8300#
MSG5    044412            3284    8183#
MSG6    044473            6998    8192#
MSG7    044527            3299    8197#
MSG74   045720            6871    8314#
MSG75   045766            6873    8321#
MSG76   046023            6929    8326#
MSG77   046045            6934    8330#
MSG8    044564            5013    8202#
MSG9    044623            5017    8208#
MSP   = 000100            1256#
MULT6   035566            6691    6717#
M.ALGN= 040000            1332#
M.CADD= 017760            1331#   6481    6504
M.CDIF= 017760            1324#   6448    6472    6491
M.DPID  036774            7041#
M.DP40  037032            7052#   7074
M.DP41  037066            7057    7065#
M.DP42  037076            7063    7069#
M.DP44  037130            7076    7082#
M.DP50  037142            7050    7087#
M.DRV = 000007            1323#
M.HEAD= 007000            1334#   6524
M.ID  = 000003            1330#
M.OFST= 017760            1325#
M.PAR = 100000            1335#   6837
M.SECT= 000760            1333#   6436    6514
M.SER = 077770            1326#
NED   = 010000            1211#   3510    3522    3603    3606    6338
NEM   = 004000            1210#
NUDRV   014764            3442    3536    3637#   5933
NXF   = 000004            1220#
OFFERR  001442            2013#   5025*   5099*   5101    5117    5125    5134    5138
OFFSET= 000015            1178#   4609    5036    5202
OFST  = 000004            1239#
OPI   = 020000            1231#
OR    = 000200            1206#
O.ADR1  055414            9239*   9290*   9292*   9462    9463*   9501*   9516    9533    9724#
O.BACK  056140            9334    9375#
O.BD    057432            9520    9521    9669#
O.BKPT  055646            9285#   9338
O.BK1   056600            9233    9498#
O.BK2   056634            9503    9506#
O.BK3   056652            9505    9512#
O.BRK   056570            9250    9496#
O.BUF   057461            9578    9587    9592    9697#
```

```
0.B2     056734          9517    9526#
0.B3     056774          9525    9534#
0.CADV   057144          9351    9368    9391    9395    9425    9429    9534    9577#
0.CLGL   055762          9308    9310    9318#
0.CLGT=  000014          9322    9687#
0.CLS1   057370          9645    9647#
0.CR     057434          9654    9656    9657    9671#
0.CRET   056074          9329    9358#
0.CRLF   057400          9366    9423    9519    9558    9563    9654#
0.CRLS   057406          9303    9656#
0.CRS    057412          9655    9657#
0.CSR1   057430          9554*   9570    9665#
0.CSR2   057431          9555*   9571    9666#
0.CT     055416          9483*   9526*   9528*   9725#
0.C1     056546          9486#   9527
0.DCD    055712          9291    9293    9303#   9360    9411    9536
0.DCD1   055716          9304#   9352    9396
0.EFF    056222          9337    9402#
0.EFF1   056340          9414    9433#
0.ENTR   055426          9230#
0.ERR    055676          9277    9288    9300#   9323    9348    9406    9649
0.ERR1   056232          9406#   9479
0.ERR2   056052          9348#   9382    9455
0.FTYP   057306          9302    9370    9385    9427    9532    9603    9619    9624#
0.GET    057230          9256    9306    9600#   9605    9607    9609
0.GO     056404          9331    9453#
0.GO2    056462          9464#   9489
0.LG  =  000006          9260    9695#
0.LGCH   057437          9319    9675#   9687
0.LGDR   056012          9326    9328#   9340
0.LGL =  000030          9340#
0.LGL1   055764          9319#   9324
0.LGL2   056004          9320    9325#
0.MSK    055406          9407    9408    9410    9717#
0.ODT    055422          1352    9228#   9706
0.OFST   056150          9335    9381#
0.OF1    056216          9394    9396#
0.OP1    056106          9332    9364#
0.OP2    056114          9281    9366#   9377
0.ORPC   055620          9273#   9333
0.P      057427          9248*   9453*   9478    9480*   9529*   9663#
0.PRI    055404          9240    9502    9506    9716#
0.PROC   056512          9339    9478#
0.PR1    056534          9482    9484#
0.RALL   055666          9251    9286    9292#
0.RCSR=  177560          9218#   9554    9556*   9566    9568    9570*   9600
0.RDB =  177562          9217#   9602
0.REGT   055550          9256#   9330
0.RSE1   057126          9567    9570#
0.RSP    055560          9258#   9261
0.RST    055464          9229    9238#
0.RSTT   057076          9459    9485    9563#
0.RST1   055514          9237    9245#
0.RTIT   056510          9247*   9473#
0.SCAN   055722          9267    9306#   9317
0.SP     055612          9259    9268#
```

```
0.SPC   057156          9580#   9588
0.SP1   055600          9264#   9269
0.STM = 000340          9207#   9249    9458    9484    9486
0.STRT  055452          9228    9235#
0.SVR   057006          9238    9498    9540#
0.SVTT  057044          9518    9530    9554#
0.T     057426          9460*   9487*   9499    9662#
0.TBIT  056434          9460#   9500
0.TBT = 000020          9208#   9461    9488    9524
0.TC    057372          9642    9648#
0.TCLS  057354          9273    9358    9364    9375    9641#
0.TCSR= 177564          9220#   9555    9557*   9564    9571*   9624    9630
0.TDB = 177566          9219#   9626*   9632*
0.TL    057453          9257    9260    9268    9689#   9695
0.TRTC  057470          9292    9463    9702#
0.TVEC= 000014          9206#   9231    9249*   9250*
0.TYPE  057272          9522    9593    9616#   9620    9658
0.TYP1  057352          9617    9628    9636#
0.TYP2  057332          9630#   9631    9634
0.UIN   055420          9239    9462*   9501    9726#
0.UPC   055400          9232*   9457*   9472    9496*   9513    9515*   9714#
0.URO   055362          9235    9265    9707#
0.USP   055376          9236*   9541*   9542    9713#
0.UST   055402          9230*   9461*   9471    9488*   9497*   9504    9524*   9715#
0.WDS   056230          9403    9405#
0.WDS2  056246          9410#   9432
0.WDS3  056276          9421#   9434    9440    9449
0.WDS4  056334          9421    9431#
0.WRD   056042          9328    9344#
0.WRDA  056066          9345    9353#
0.WRD1  056050          9347#   9354    9371
0.WSCH  056226          9336    9404#
0.XXX   057424          9540*   9548    9661#
0.45    055526          9246    9248#
PACK  = 000003          1173#   3745
PARAM   001336          1970#   3165*   3168*   3277
PARSRT  012532          1349    3165#
PAT   = 000020          1254#
PCA   = 004000          1261#
PCD   = 010000          1262#
PCLKF   007504          2143#   3321*   3328*   6785    6806
PCVEC   001332          1964#   3322    3329
PCYL    001352          1978#
PFSRT   015122          3681#   7028
PGE   = 002000          1209#
PIP   = 020000          1247#
PIRQ  = 177772          1052#
PIRQVE= 000240          1146#
PKRB    001324          1960#
PKS     001320          1958#   3320    3327    6790*   6810*
PKSB    001322          1959#   6789*
PPTP    007456          2127#   3268*
PRGSRT  012546          3166    3169#   6033
PR0   = 000000          1069#   3171    3359
PR1   = 000040          1070#
PR2   = 000100          1071#
```

```
PR3    = 000140          1072#
PR4    = 000200          1073#
PR5    = 000240          1074#  1957
PR6    = 000300          1075#  3332
PR7    = 000340          1076#  3235  3241  7023  7024
PS     = 177776          1049#  1050
PSEC     001376          1991#
PSW    = 177776          1050#
PWRVEC= 000024           1141#  3205*  3206*  7013*  7022*  7023*
QKCYLD   034354          6461#
RCYLA    034566          6310   6503#
RCYLD    034514          6307   6490#
RDCHR  = 104410          7690   8072#
RDCYLA   034444          6478#
RDCYLD   034270          6445#
RDDATA= 000021           1180#  3833  4192
RDGATE= 100000           1265#
RDHEAD= 000025           1182#  4774
RDLIN  = 104411          6001   7764  8073#
RDOCT  = 104412          6046   6059  8074#
RDSEC    034220          6433#  6534  6538  6541  6689
RDTAB    006322          2045#  4188  4247  4255
RDY    = 000200          1188#  5638  6086  6100  6465
RECAL  = 000013          1177#  3789  5147  5394  5565  6897
RESREG= 104414           7824   7876  8076#  9175
RESVEC= 000010           1136#
RHEAD    034662          6315   6523#
RHTAB    001712          2038#  4773  4812  6696  6698  6702  6705  8920
RKASOF= 000016           1159#  3373  4606*  4652*  4696*  4748*  5030*  5199*  6125  6160  6186
RKBA   = 000004          1154#  3369  3827*  3881*  3899*  3908*  3917*  3989*  4013*  4090*  4121*  4188*  4262*
                         4275*  4335*  4380*  4415*  4488*  4559*  4897*  4958*  5086*  5255*  5330*  5450*  5504*
                         5629*  5709*  5761*  5807*  5862*  6121
RKCS1  = 000000          1152#  3366  3467*  3567*  3745*  3787*  3789*  3805*  3807*  3833*  3935*  3992*  4017*
                         4035*  4100*  4126*  4144*  4192*  4279*  4338*  4392*  4396*  4420*  4424*  4441*  4463*
                         4492*  4496*  4529*  4570*  4609*  4668*  4670*  4679*  4712*  4774*  4833*  4903*  4921*
                         4963*  5036*  5075*  5077*  5090*  5145*  5147*  5163*  5165*  5202*  5241*  5243*  5259*
                         5277*  5335*  5392*  5394*  5410*  5412*  5454*  5508*  5563*  5565*  5581*  5583*  5633*
                         5638   5713*  5732*  5765*  5811*  5872*  5893*  6086  6100  6118  6356*  6368*  6371*
                         6378*  6381*  6388*  6391*  6398*  6400*  6463*  6465  6879*  6895*  6897*  6913*  6915*
RKCS2  = 000010          1156#  3367  3448*  3462*  3466*  3562*  3566*  3788*  3806*  4123*  4274*  4416*  4489*
                         4669*  4789  4898*  4959*  5076*  5087*  5146*  5164*  5242*  5256*  5331*  5393*  5411*
                         5451*  5505*  5564*  5582*  5630*  5762*  5808*  6119  6355*  6369*  6379*  6389*  6399*
                         6417*  6421*  6896*  6914*
RKDA   = 000006          1155#  3370  3829*  3883*  3901*  3910*  3919*  4012*  4120*  4187*  4264*  4277*  4337*
                         4414*  4490*  4896*  4961*  5032*  5089*  5254*  5333*  5632*  6122
RKDB   = 000024          1161#  3375  4288  4504  4784  4785  4786  4972  5098  5344  5517
RKDC   = 000020          1160#  3374  3831*  4768*  5449*  5503*  5711*  5730*  5760*  5864*  5890*  6126
RKDS   = 000012          1157#  3371  6123
RKECPS= 000030           1165#  3379  6130
RKECPT= 000032           1166#  3380  6131
RKER   = 000014          1158#  3372  6124
RKMR1  = 000026          1162#  3376  3699*  3796*  5154*  5401*  5572*  6127  6166*  6170*  6190*  6194*  6370*
                         6380*  6390*  6422*  6433*  6445*  6462*  6478*  6558*  6575*  6904*
RKMR2  = 000034          1163#  3377  6104  6109  6128  6471
RKMR3  = 000036          1164#  3378  6105  6110  6129
RKPRI    001316          1957#  6075
```

```
RKVEC   001314          1956#  3304*  6063*  6066*  6073
RKWC  = 000002          1153#  3368   3832*  3990*  4014*  4091*  4124*  4189*  4276*  4336*  4381*  4417*  4487*
                        4560*  4899*  4960*  5088*  5257*  5332*  5452*  5506*  5631*  5710*  5763*  5809*  5863*
                        6120
RLS   = 000010          1202#
RSEC    034624          6314   6513#
RTT   = 000006          9210#
SAVREG= 104413          7809   7850   8075#  9119
SBPAR   036066          6235   6239   6243   6247   6251   6255   6259   6823#
SCLR  = 000040          1204#  3448   3462   3562   6417
SCOP1 = 104415          3459   3559   3739   4005   4113   4181   4268   4329   4407   4553   5323   5752   5800
                        8077#
SCOP1$  036140          6847#  8077
SEC     001370          1988#  6799*
SECFLG  035302          6608   6628#
SECNT   001374          1990#
SECTOR  001402          1993#  4011*  4012   4030*  4031   4119*  4120   4139*  4140   4187   4264   4277   4337
                        4413*  4414   4436*  4437   4490   4895*  4896   4916*  4917   4961   5089   5253*  5254
                        5272*  5273   5333   6435*  6436*  6437*  6438*  6439*  6440*  6513*  6514*  6515*  6516*
                        6517*  6518*  6535   6539   6542   6690*  6694   6695*  6696*  6707   6717*  6718   6719*
                        6720*
SEEK  = 000017          1179#  3935   4712   4833   5732   5893
SELDRV= 000001          1172#  3467   3567   4396   4424   4463   4496   4529   6356   6371   6381   6391   6400
                        6463
SETINT  032212          3305   6064   6073#
SIZFLG  007510          2145#  3306*  3452   6019*  6039*
SKI   = 000002          1219#
SORT    035460          6687#
SPACE2  055336          9153   9162   9173   9185#
SRTSPL= 000011          1176#  6879
SRTTAB  002116          2039#  6699
ST    = 177776          9204#  9230   9231*  9244*  9458*  9484*  9486*  9512*
STACK = 001100          1040#  3170   3197   3357   3406   3460   3555   3560   3651   3691   3731   3740   3781
                        3967   4006   4085   4114   4182   4269   4330   4375   4408   4554   4599   4890   5192
                        5324   5442   5615   5691   5753   5801   5929   6320   6987   7025   9181
START   012542          1347   3168#
START1  013150          3259#
STKLMT= 177774          1051#
STOP    036172          5913   6862#  7531   7614
ST2     013216          3266   3277#
ST3     013262          3267   3295#
ST4     013314          3279   3300   3303#
ST5     013340          3286   3308#  3446   3502   5958   6936
SUBCLR  034140          3695   3733   3742   3783   3818   3857   3931   3975   4008   4087   4116   4184   4271
                        4332   4377   4410   4556   4603   4764   4830   4892   5027   5196   5326   5446   5622
                        5696   5757   5803   5859   6417#  6864   6876
SVAL  = 100000          1249#
SWR     001140          1874#  3195   3217*  3219   3225*  3232*  5988   6271   6281   6291   6301   6847   6951
                        6959   6964   6983   7107   7121   7123   7129   7136   7174   7181   7193   7197   7534
                        7573   7628*  9177
SWREG   000176          1345#  3225   5988   7534   7573   7596
SW0   = 000001          1104#
SW00  = 000001          1094#  1104
SW01  = 000002          1093#  1103
SW02  = 000004          1092#  1102
SW03  = 000010          1091#  1101
```

```
SW04  = 000020          1090#  1100
SW05  = 000040          1089#  1099
SW06  = 000100          1088#  1098
SW07  = 000200          1087#  1097
SW08  = 000400          1086#  1096
SW09  = 001000          1085#  1095
SW1   = 000002          1103#
SW10  = 002000          1084#
SW11  = 004000          1083#
SW12  = 010000          1082#  9177
SW13  = 020000          1081#  6951
SW14  = 040000          1080#  6964
SW15  = 100000          1079#
SW2   = 000004          1102#
SW3   = 000010          1101#
SW4   = 000020          1100#
SW5   = 000040          1099#
SW6   = 000100          1098#
SW7   = 000200          1097#
SW8   = 000400          1096#
SW9   = 001000          1095#  6271   6281   6291   6301   6847   6959   6983
TBITVE= 000014          1137#
TEMP1   007376          2086#  3449*  3463*  3468*  3473*  3474*  3475   3563*  3568*  3573*  3574*  3575   3746*
                        3792*  3808*  3834*  3936*  3940*  3993*  4018*  4101*  4127*  4193*  4280*  4339*  4393*
                        4397*  4421*  4425*  4464*  4493*  4497*  4530*  4571*  4610*  4627*  4671*  4713*  4717*
                        4775*  4834*  4838*  4904*  4964*  5037*  5054*  5078*  5091*  5150*  5166*  5203*  5220*
                        5244*  5260*  5336*  5397*  5413*  5455*  5509*  5568*  5584*  5635*  5714*  5733*  5737*
                        5766*  5812*  5873*  5894*  5898*  6088*  6102*  6158*  6162*  6188*  6204   6206*  6232
                        6234*  6236   6238*  6240   6242*  6244   6246*  6248   6250*  6252   6254*  6256   6258*
                        6260   6262*  6354   6357*  6360*  6366   6372*  6382*  6392*  6401*  6407*  6418*  6461
                        6464*  6467*  6473*  6533*  6545*  6557*  6562*  6574*  6579*  6610*  6613*  6614   6829*
                        6837*  6880*  6900*  6916*  8904
TEMP2   007400          2087#  3801*  3821*  3860*  3861   3897*  3906*  3915*  5159*  5406*  5577*  6164*  6564*
                        6611*  6612*  6614*  6615   6884*  6909*
TEMP3   007402          2088#  3822*  3863   3868   3873   3894*  3895   3904   3913
TEMP4   007404          2089#  3826*  3827   3881   3898*  3899   3907*  3908   3916*  3917
TEMP5   007406          2090#  3828*  3829   3882*  3883   3900*  3901   3909*  3910   3918*  3919   5621*  5632
                        5664   5666*  5694*  5726*  5857*  5885*
TERR1   035710          6737   6741   6748   6752   6766#  6771   6778
TIMUP   001372          1989#  6784*  6795*  6801*
TITLE   031712          3261   3298   5979#
TKVEC = 000060          1144#  7511*  7512*
TOCYL   001346          1976#  4092*  4094   4382*  4386   4561*  4564   4767*  4812   5693*  5701   5704   5711
                        5727*  5728   5749*  5755   5760   5795   5851*  5852   5856*  5864   5866   5886*  5887
                        8888   8912   8920
TPVEC = 000064          1145#
TRAPPC  001334          1968#  6980*  8887
TRAPVE= 000034          1143#  3203*  3204*  7024*
TRT   = 000003          9209#  9702
TRTVEC= 000014          1138#
TRUERR  035610          4027   4136   4433   4913   5269   5464   6731#
TSTATN  032512          3811   4674   5081   5169   5247   5416   5587   6139#  6919
TST1    013524          3333   3355#
TST10   016426          3887   3923   3965#
TST11   017144          4083#
TST12   020752          4373#
```

```
TST13   022262          4597#
TST14   023574          4864    4888#
TST15   025412          5135    5190#
TST16   026760          5440#
TST17   027754          5613#
TST2    013700          3383    3404#
TST20   030240          5619    5627    5665    5689#
TST3    014440          3453    3553#
TST4    014770          3583    3649#
TST5    015122          3689#
TST6    015260          3694    3729#
TST7    015362          3737    3751    3779#
TYPDS = 104405          5947    8067#
TYPE  = 104401          3265    3280    3282    3284    3299    3340    3415    3425    3454    3483    3524    3657    3671
                        3679    3698    3720    3721    3924    3928    3929    3971    3972    4000    4053    4108    4162
                        4203    4309    4403    4459    4525    4578    4781    4793    4939    4993    5013    5017    5023
                        5103    5109    5110    5127    5140    5179    5295    5365    5483    5538    5721    5880    5945
                        5948    5982    6032    6213    6871    6873    6929    6934    6953    6998    7026    7176    7184
                        7252    7346    7472    7528    7540    7594    7595    7598    7609    7619    7630    7649    7697
                        7703    7708    7712    7717    7718    7720    7723    7727    7793    7795    7935    8063#   9129
                        9130    9134    9135    9144    9153    9157    9162    9164    9170    9173
TYPERR  055062          7183    9119#
TYPOC = 104402          7597    8064#   9150
TYPON = 104404          8066#
TYPOS = 104403          3428    3486    3527    3675    5020    5106    6216    6956    7001    8065#
T.A2  = 000001          2116#   4048    4067    4157    4176    4224    4241    4304    4323    4454    4520    4641    4631
                        4743    4806    4934    4953    4988    5007    5068    5234    5290    5309    5360    5379    5478
                        5497    5533    5552    5790    5836    6305
T.B2  = 000002          2117#   4048    4067    4157    4176    4224    4241    4304    4323    4454    4520    4641    4691
                        4743    4806    4934    4953    4988    5007    5068    5234    5290    5309    5360    5379    5478
                        5497    5533    5552    5790    5836    6308
T.B3  = 000004          2118#   6312
T1      001404          1995#
T10     001406          1996#   3449    3463    3468    3563    3568    3792    3801    3808    4397    4425    4464    4497
                        4530    4671    5078    5150    5159    5166    5244    5397    5406    5413    5568    5577    5584
                        6357    6372    6382    6392    6401    6418    6464    6880    6900    6909    6916
T100    001414          1999#   4610    5037    5203    6884
T50     001410          1997#   3936    4713    4834    5733    5894
T500    001412          1998#
T5000   001416          2000#   4627    5054    5220    6533
T50000  001420          2001#   3834    3940    3993    4018    4101    4127    4193    4280    4393    4421    4493    4571
                        4717    4775    4838    4904    4964    5260    5336    5455    5509    5714    5737    5766    5812
                        5873    5898
UFE   = 000400          1207#   3506    3599
UNLD    007322          2047#   3308*   6867    6888*
UNLOAD= 000007          1175#
UNS   = 040000          1232#
UPE   = 020000          1212#
VV    = 000100          1243#
WCE   = 040000          1213#   4286    4343    4502    4970    5095    5342    5515
WDCNT   001456          2021#   8904
WD1     001436          2010#   4248*   4288*   4504*   4972*   5098*   5344*   5517*   8896
WD2     001440          2011#   4249*   4289*   4505*   4973*   5345*   5518*   8896
WLE   = 004000          1229#
WORD    001472          2029#
WRDATA= 000023          1181#   4017    4126    4420    4903    5259    5454    5633    5765    5811
```

```
WRHEAD= 000027          1183#  3992   4100   4392   4570   5713   5872
WRL   = 004000          1246#
WRTCHK= 000031          1184#  4279   4338   4492   4963   5090   5335   5508
WRTGAT= 040000          1264#
$APTHD  001000          1374   1380#
$ASTAT= ****** U        7389   7404
$ATYC   040364          7360   7362#
$ATY1   040340          7358#
$ATY3   040346          7237   7359#
$ATY4   040356          7189   7361#
$AUTOB  001134          1871#  5992*  7587   7744
$BASE   001264          1941#  3303*  3365   6050*  6052*
$BOADR  001122          1866#
$BODAT  001126          1868#
$BELL   001200          1892#  7176   7209   7540   7737
$CDW1   001270          1943#
$CDW2   001272          1944#
$CHARC  040110          7254*  7264*  7271   7280*  7285#
$CKSWR  041332          7573#  8071
$CMTAG  001100          1854#  3192   3193   3201   3207   3208   3209
$CM3  = 000000          1884#
$CM4  = 000006          1884#  1885#  1886#  1887#  1888#  1889#  1890#
$CNTLC  042242          7528   7609   7737#
$CNTLG  042254          7594   7739#
$CNTLU  042247          7619   7712   7738#
$CPUOP  001236          1915#
$CRLF   001205          1894#  3425   3483   3679   3720   3721   3928   3929   5023   5109   5110   7184   7209
                        7253   7288   7630   7717   7737   7798   9129   9134   9144   9157   9170
$DBLK   040330          7312   7346   7354#
$DB2D   042564          7850#  7912
$DB2O   042444          3926   7809#
$DDW0   001274          1945#
$DDW1   001276          1946#
$DDW2   001300          1947#
$DDW3   001302          1948#
$DDW4   001304          1949#
$DDW5   001306          1950#
$DDW6   001310          1951#
$DDW7   001312          1952#
$DECVL  042744          7852   7898#
$DEVCT  001220          1906#  3252*  3313*  3661   5930*  5931
$DEVM   001266          1942#  3419
$DOAGN  031646          5941   5950   5956#
$DTBL   040320          7315   7350#
$ENDAD  031636          1360   5952#  7204
$ENDCT  031604          3207   5943#
$ENDMG  031655          5945   5960#
$ENULL  031652          5948   5959#
$ENV    001230          1911#  5986   7186   7232   7367   7391
$ENVM   001231          1912#  3230   3411   7234   7239   7369
$EOP    031514          3973   4001   4054   4109   4163   4204   4310   4404   4460   4526   4579   4782   4794
                        4940   4994   5296   5366   5484   5539   5722   5881   5925#  9182
$EOPCT  031576          3207*  5940#  5944   6930*
$EOP1   031550          3387   3658   5932   5934#  6932   6975
$ERFLG  001103          1857#  6849   7096   7125   7127   7133*  7155   7171*  7209   9179
$ERMAX  001115          1863#  3210*  7127   7150*  7155
```

```
$ERROR  037432          3201    7169#
$ERRPC  001116          1864#   7178*   7179*   7180    7209    8880    8888    8896    8904    8912    8920    8928    8938
                        8949
$ERRTB  007512          2160#   9126
$ERTTL  001112          1861#   7177*   7209
$ESCAP  001176          1891#   3209*   3311*   4608*   4626*   5035*   5053*   5201*   5219*   5747*   5891*   6263    6267*
                        6277*   6287*   6297*   6317*   6321*   6931*   6979*   6982*   7149*   7200    7202    7209
$ETABL  001230          1910#
$ETEND  001314          1386    1953#
$FATAL  001212          1903#   7395*
$FFLG   040604          7358*   7361*   7389    7398*   7406#
$FILLC  001156          1882#   7257    7288
$FILLS  001155          1881#   7288
$GDADR  001120          1865#
$GDDAT  001124          1867#
$GET42  031626          5949#
$GTSWR  041422          7595#   8069
$HD     =  000000       1011
$HIBTS  001000          1381#
$HIOCT  042442          7786*   7797#
$ICNT   001104          1858#   7140*   7141    7143*   7154
$INTAG  001135          1872#   7592*   7611    7631    7744
$ITEMB  001114          1862#   7180*   7188    7209    9120
$LF     001206          1895#   7209    7288    7727    7737    7798
$LFLG   040603          7399*   7405#
$LPADR  001106          1859#   3211*   6962    7131*   7147*   7152    7154
$LPERR  001110          1860#   3212*   6322    6851    6854*   6988    7131    7148*   7154    7199
$MADR1  001242          1928#
$MADR2  001246          1932#
$MADR3  001252          1935#
$MADR4  001256          1938#
$MAIL   001210          1382    1386    1901#   3229    5986    7146    7186    7232
$MAMS1  001240          1922#
$MAMS2  001244          1930#
$MAMS3  001250          1933#
$MAMS4  001254          1936#
$MBADR  001002          1382#
$MFLG   040602          7359*   7365    7400*   7404#
$MMEM   042272          7598    7742#
$MSGAD  001224          1908#   7375*   7378
$MSGLG  001226          1909#   7380*
$MSGTY  001210          1902#   7373    7381*   7393    7397*
$MSWR   042261          7595    7740#
$MTYP1  001241          1923#
$MTYP2  001245          1931#
$MTYP3  001251          1934#
$MTYP4  001255          1937#
$MULT   043054          7954#
$MXCNT  037430          7144    7154#
$NULL   001154          1880#   7259    7288
$NWTST=  000001         3345#   3347    3390#   3392    3538#   3540    3640#   3642    3682#   3684    3723#   3725    3756#
                        3758    3955#   3957    4073#   4075    4365#   4367    4584#   4586    4867#   4869    5180#   5182
                        5424#   5426    5595#   5597    5670#   5672
$OCNT   041030          7444*   7473*   7486#
$OCTVL  042546          3702    7811    7836#
$OMODE  041032          7439*   7443*   7448    7451*   7462*   7488#
```

```
$OVER   037414          7108    7124    7132    7142    7151#
$PASS   001216          1905#   3229*   3693    3922    5937*   5938*   5946    5959    7138    7155
$PASTM  001006          1384#
$PWRCT  036772          7019*   7020*   7030#
$PWRDN  036706          3205    7013#   7022
$PWRUP  036720          7013    7019#
$QUES   001204          1893#   7209    7288    7649    7720    7737    7795    7798
$RDCHR  041674          7662#   8072
$RDDEC= ****** U        8075
$RDLIN  041764          7685#   8073
$RDOCT  042304          7759#   8074
$RDSZ = 000022          7678#
$RESRE  043224          8020#   8076
$RTNAD  031650          5958#
$R2A  = ****** U        8077
$SAVRE  043166          8004#   8075
$SB2D   042760          7910#
$SCOPE  037152          3199    7105#
$SETUP= 000137          3162#   3198    3199    3201    3203    3205    3207    3208    3209    3211    5935    5983    7106
                        7170    7196    7204    7532    7537    7538    7568    7744
$STUP = 177777          3162#
$SUPRS  043014          3719    3927    7927#
$SVLAD  037360          7116    7145#
$SVPC = 000244          1358#   1363
$SWR  = 167400           996#   1011    1016    1017    1018    1019    1020    1021    1022    1023    1890    1891    1892
                        3208    3209    3211    3212    3356    3405    3554    3650    3690    3730    3780    3966    4084
                        4374    4598    4889    5191    5441    5614    5690    5922    5936    5951    5957    5959    7097
                        7098    7099    7100    7101    7107    7119    7121    7122    7125    7126    7127    7134    7135
                        7136    7148    7151    7154    7161    7162    7163    7164    7165    7174    7181    7193    7197
                        7209
$SWREG  001232          1913#   3232
$SWRMK= 000000          1023    1024    7101    7102    7123
$TESTN  001214          1904#   3652*   7146*   8880    8887    8888    8896    8904    8912    8920    8928    8938    8949
$TIMES  001174          1890#   3208*   3356*   3405*   3554*   3650*   3690*   3730*   3780*   3966*   4084*   4374*   4598*
                        4889*   5191*   5441*   5614*   5690*   5928*   5936*   7134*   7141    7144*   7154
$TKB    001146          1877#   7492    7513    7524    7549    7577    7604
$TKCNT  041034          7493#   7508*   7538    7555*   7669    7671*
$TKINT  041044          3175    7508#   7529    7590
$TKQEN= 041043          7497#   7563    7674
$TKQIN  041036          7494#   7509*   7510    7561*   7562*   7563    7565*
$TKQOU  041040          7495#   7510*   7672    7673*   7674    7676*
$TKQSR  041042          7496#   7509    7565    7676
$TKS    001144          1876#   7492    7514*   7545*   7547    7553*   7575    7591*   7601    7613*   7633*
$TKSRV  041114          7511    7524#
$TMP0   001160          1884#
$TMP1   001162          1885#
$TMP2   001164          1886#
$TMP3   001166          1887#
$TMP4   001170          1888#
$TMP5   001172          1889#   6263*   6317    6321
$TN   = 000021           997#   1011    3333    3345    3356#   3383    3390    3405#   3453    3538    3554#   3583    3640
                        3650#   3652    3653    3682    3690#   3694    3723    3730#   3737    3751    3756    3780#   3887
                        3923    3955    3966#   4073    4084#   4365    4374#   4584    4598#   4864    4867    4889#   5135
                        5180    5191#   5424    5441#   5595    5614#   5619    5627    5665    5670    5690#
$TNPWR  042674          7857    7858    7878#
$TPB    001152          1879#   7277*   7288
```

```
$TPFLG  001157          1883#   7226    7288
$TPS    001150          1878#   7275    7288
$TRAP   043262          3203    8040#
$TRAP2  043304          8051#   8062
$TRP  = 000016          8055#   8064#   8065#   8066#   8067#   8068#   8069    8070#   8071    8072#   8073#   8074#   8075#
                        8076#   8077#   8078#
$TRPAD  043316          8045    8062#
$TSTM   001004          1383#
$TSTNM  001102          1856#   3653#   5935#   7096    7123    7145#   7146    7151    7155    7173    7209
$TTYIN  042220          7687    7688    7700    7718    7732    7736#
$TYPBN= ****** U        8068
$TYPDS  040114          7300#   8067
$TYPE   037632          7226#   7386    8055    8063
$TYPEC  040044          7256    7263    7270    7275#   7276    7635
$TYPEX  040112          7281    7283    7286#
$TYPOC  040632          7442#   8064
$TYPON  040646          7441    7444#   8066
$TYPOS  040606          7437#   8065
$UNIT   001222          1907#   3253#   3314#   3663#   3668    3672    3788    3806    4669    5076    5146    5164    5242
                        5393    5411    5564    5582    6140    6159    6185    6227    6228    6229    6230    6355    6369
                        6379    6389    6399    6421    6896    6914
$UNITM  001010          1385#
$USWR   001234          1914#
$VECT1  001260          1939#
$VECT2  001262          1940#
$XTSTR  037164          7110#
$$GET4= 000000          5951#
$DFILL  041031          7438#   7442#   7452    7487#
$40CAT= ****** U        7107    7183
.     = 055422          1339#   1343#   1348#   1351#   1358    1359#   1361#   1363#   1364#   1370    1371#   1373#   1375#
                        1853#   1896    2037#   2038#   2039#   2041#   2042#   2043#   2044#   2045#   3196    3211    3212
                        5959    5963    7015    7071    7154    7155    7209    7288    7354#   7407#   7492    7496#   7497
                        7498#   7736#   7737    7744#   7798    7836#   7898#   9194#   9195#   9340    9565    9569    9601
                        9625    9687    9695    9698#   9706#
.$ASTA= ****** U        7359    7362
.$X   = 001000          1370#   1375
```

```
CALIB   1476#  3786  5143  5390  5561  6894
CHECK   1431#  3849  4046  4065  4155  4174  4222  4239  4302  4321  4356  4452  4480  4518  4546
        4618   4639  4689  4741  4804  4932  4951  4986  5005  5045  5066  5211  5232  5288  5307
        5358   5377  5476  5495  5531  5550  5656  5788  5834
COMMEN  1147#
CWD2    1446#  5314
DRCLR   1457#  3804  4667  5074  5162  5240  5409  5580  6912
ENDCOM  1147#
EOPGM   1812#  5926
ERROR   1041#  3386  3444  3451  3465  3470  3492  3500  3515  3519  3530  3534  3565  3570  3586
        3594   3612  3616  3620  3624  3696  3734  3743  3748  3752  3784  3794  3800  3803  3810
        3813   3819  3836  3840  3852  3853  3854  3855  3858  3865  3870  3875  3878  3890  3932
        3938   3942  3946  3976  3995  3999  4009  4020  4033  4037  4049  4050  4051  4052  4055
        4068   4069  4070  4071  4088  4103  4107  4117  4129  4142  4146  4158  4159  4160  4161
        4164   4177  4178  4179  4180  4185  4195  4201  4208  4211  4225  4226  4227  4228  4242
        4243   4244  4245  4253  4272  4282  4290  4293  4305  4306  4307  4308  4324  4325  4326
        4327   4333  4341  4345  4359  4360  4361  4362  4378  4395  4399  4402  4411  4423  4427
        4439   4443  4455  4456  4457  4458  4461  4466  4469  4483  4484  4485  4486  4495  4499
        4506   4509  4521  4522  4523  4524  4532  4535  4549  4550  4551  4552  4557  4573  4577
        4604   4612  4621  4622  4623  4624  4629  4642  4643  4644  4645  4649  4659  4664  4673
        4676   4683  4692  4693  4694  4695  4703  4708  4715  4719  4723  4730  4744  4745  4746
        4747   4755  4760  4765  4777  4780  4792  4807  4808  4809  4810  4814  4820  4824  4828
        4831   4836  4840  4844  4893  4906  4919  4923  4935  4936  4937  4938  4941  4954  4955
        4956   4957  4966  4974  4977  4989  4990  4991  4992  5008  5009  5010  5011  5028  5039
        5048   5049  5050  5051  5056  5069  5070  5071  5072  5080  5083  5093  5152  5158  5161
        5168   5171  5197  5205  5214  5215  5216  5217  5222  5235  5236  5237  5238  5246  5249
        5262   5275  5279  5291  5292  5293  5294  5297  5310  5311  5312  5313  5317  5320  5327
        5338   5346  5349  5361  5362  5363  5364  5380  5381  5382  5383  5399  5405  5408  5415
        5418   5447  5457  5467  5479  5480  5481  5482  5485  5498  5499  5500  5501  5511  5519
        5522   5534  5535  5536  5537  5553  5554  5555  5556  5570  5576  5579  5586  5589  5623
        5626   5641  5643  5647  5659  5660  5661  5662  5697  5716  5720  5735  5739  5743  5758
        5768   5773  5777  5791  5792  5793  5794  5797  5804  5814  5819  5823  5837  5838  5839
        5840   5843  5860  5875  5879  5896  5900  5904  6335  6340  6344  6359  6374  6384  6394
        6403   6420  6469  6865  6877  6882  6886  6902  6908  6911  6918  6921  6981
ESCAPE  1147#
F.EAB   1411#  3841  4038  4057  4147  4166  4213  4231  4294  4313  4347  4444  4471  4510  4537
        4631   4732  4796  4924  4943  4978  4997  5058  5224  5280  5299  5350  5369  5468  5487
        5523   5542  5648  5779  5825
GETPRI  1147#
GETSWR  1147#  5983
HDCHK3  1594#  4771
HDTBL   1607#  4093  4385  4563  5865
LOOP    1395#  3458  3558  3739  4005  4113  4181  4267  4329  4407  4553  5323  5752  5799
MSG     3345#  3347  3390#  3392  3538#  3540  3640#  3642  3682#  3684  3723#  3725  3756#  3758  3955#
        3957   4073#  4075  4365#  4367  4583#  4586  4867#  4869  5180#  5182  5424#  5426  5595#  5597
        5670#  5672
MULT    1147#
NEWTST  1147#  3345  3390  3538  3640  3682  3723  3756  3955  4073  4365  4584  4867  5180  5424
        5595   5670
OFFDIR  1827#  4651  4696  4748
OFSET   1787#  4607  5034  5200
OWNTAG  1388#  1955
POP     1147#  7341  7401  7402  7787  7983  8025
PUSH    1147#  7300  7362  7364  7385  7761  7954  8005
QKSEEK  1619#  3934  4711  4832  5731  5892
QKSRT   1507#  6874
```

```
RDATA    1701#  4191
RDHDR    1561#  4772
REPORT   1147#
SCOPE    1042#  3355  3404  3553  3649  3689  3729  3779  3965  4083  4373  4597  4888  5190  5440
         5613   5689  5927  5934
SETPRI   1147#  7665
SETTRA   8055#  8064  8065  8066  8067  8069  8071  8072  8073  8074  8075  8076  8077
SETUP    1147#  3191
SKIP     1147#  3333  3383  3453  3583  3694  3737  3751  3887  3923  4864  5135  5619  5627  5665
SLASH    1147#
SPACE    1147#
STARS    1147#  1356  1367  1369  1376  1849  1896  1899  3345  3354  3390  3403  3538  3552  3640
         3648   3682  3688  3723  3728  3756  3778  3955  3964  4073  4082  4365  4372  4584  4596
         4867   4887  5180  5189  5424  5439  5595  5612  5670  5688  5919  7093  7157  7211  7290
         7357   7414  7491  7568  7583  7654  7678  7747  7900  7839  7901  7919  7942  7989  8034
         9109   9118
SWRSU    1147#  3213#
TRMTRP   8055#
TYPBIN   1147#
TYPDEC   1147#  5946
TYPNAM   1147#
TYPNUM   1147#
TYPOCS   1147#  3426  3484  3525  3673  5018  5104  6214  6954  6999
TYPOCT   1147#  7596
TYPTXT   1147#
WDATA    1647#  4016  4125  4419  4902  5258  5453
WRCHK    1749#  4278  4491  4962  5334  5507
WRHDR    1529#  3991  4099  4391  4569  5712  5871
$$CMRE   1847#
$$CMTM   1847#  1884  1885  1886  1887  1888  1889
$$ESCA   1147#
$$NEWT   1147#  3345  3390  3538  3640  3682  3723  3756  3955  4073  4365  4584  4867  5180  5424
         5595   5670
$$SET    8055#  8064  8065  8066  8067  8069  8071  8072  8073  8074  8075  8076  8077
$$SETM   3229#
$$SKIP   1147#  3333  3383  3453  3583  3694  3737  3751  3887  3923  4864  5135  5619  5627  5665
.EQUAT    996#  1037
.HEADE    996#  1001
.SETUP    996#  3162
.SWRHI    996#  1012
.SWRLO    996#  1024#
.$ACTI    996#  1354
.$APTB   1897#
.$APTH    996#  1365
.$APTY    996#  7355
.$CATC    996#  1337
.$CMTA    996#  1847
.$DB2D    996#  7837
.$DB2O    996#  7798
.$EOP     996#  5917
.$ERRO    996#  7155
.$MULT    996#  7940
.$ROOC    996#  7745
.$READ    996#  7489
.$SAVE    996#  7987
.$SB2D    996#  7899
```

```
.$SCOP    996#   7091
.$SUPR    996#   7917
.$TRAP    996#   8032
.$TYPD    996#   7288
.$TYPE    996#   7209
.$TYPO    996#   7412


. ABS.  057472     000


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

 DZR6ID,DZR6ID.SEQ/SOL/CRF/NL:TOC/DOC=DZR6ID.P11
 RUN-TIME: 24 22 2 SECONDS
 RUN-TIME RATIO: 72/49=1.4
 CORE USED:  31K  (62 PAGES)

 DOCUMENT PAGES:  203

EOF1DZR6IDSEQ              00010000      770323            PDP10 411
```