RK611

UNIBUS DRIVE PART 2

CZR6IE0

AH-9122E-MC

MAR 1978

digital

MADE IN USA

RK611

UNIBUS DRIVE PART 2
CZR6IE0

AH-9122E-MC

FICHE 2 OF 2

MAR 1978

digital

MADE IN USA

```
        .REM %

                        IDENTIFICATION
                        --------------

                PRODUCT CODE:           AC-9120E-MC

                PRODUCT NAME:           CZR6IED UNIBUS RK6 DR PRT 2

                DATE:                   FEB 1978

                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.

THIS PROGRAM WILL TEST RK06 & RK07 WITHOUT NEED OF OPERATOR INTERVENTION.

*****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/RK07 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: 1 MIN, 30 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.

                ALSO, THE PROGRAM WILL AUTOMATICALLY DETERMINE WHETHER
                THE DRIVE IS AN RK06 OR RK07.

        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
                CZR6IEO

                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
                        CZR6IED

                        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 AS AN RK06.  IF SET, THE PROGRAM WILL BYPASS
                                        TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                        MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                        DICATING THE OTHER PORT IS ACCESSED.
                                        IF CERR IS DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07

                                TEST 3  VERIFY OPERATOR DRIVE SELECTIONS

                                        THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
                                        DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
                                        CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
                                        PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.
                                        IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
                                        ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                                        NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                                        NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
                                        VERIFY IT WAS NOT SPECIFIED.
                                        IF CERR IS DUE TO DTYE, THE DRIVE WILL BE TESTED AS
                                        AN RK07.

                                TEST 4  FIND NEXT DRIVE TO BE TESTED

                                        THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                        ADDRESS IN 'DRVAD' & $TMP4 IS SET TO CDT IF THE DRIVE
                                        DRIVE IS AN RK07.
                                        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 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
          THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
          FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
          AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
          IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
          IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
          A 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  RK06
                CYLINDER Y:  64  128  192  256  320  384  RK06

                CYLINDER X:  127 255  383  511  639  767  RK07
                CYLINDER Y:  128 256  384  512  640  768  RK07
```

```
                THE ABOVE CYL NUMBERS ARE IN DECIMAL.

        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 AO 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 032 ***
                                 .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  CZR6IEO UNIBUS RK6 DR PRT2
                                 ;*COPYRIGHT (C) 1976,1977
                                 ;*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
```

```
        000002          SW01=   2
        000001          SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

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

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

                        .SBTTL  RK06 CONTROLLER REGISTER DEFINITION
```

```

                         ;       $BASE=177440

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

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

                         ;       DRIVE COMMANDS

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

        000001           GO=     BIT0            ;GO BIT
        000100           IE=     BIT6            ;INTERRUPT ENABLE
        000200           RDY=    BIT7            ;CONTROLLER READY
        000400           BA16=   BIT8            ;BUS ADDRESS BIT 16
        001000           BA17=   BIT9            ;BUS ADDRESS BIT 17
        002000           CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06,1=RK07)
        004000           CTO=    BIT11           ;CONTROLLER TIMEOUT
        010000           CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
        020000           DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
        040000           DI=     BIT14           ;DRIVE INTERRUPT
        100000           CERR=   BIT15           ;CONTROLLER ERROR
```

```
        100000          CCLR=   BIT15                   ;CONTROLLER CLEAR

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

        000007          DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
        000010          RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
        000020          BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
        000040          SCLR=   BIT5                    ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
        000100          IR=     BIT6                    ;INPUT READY
        000200          OR=     BIT7                    ;OUTPUT READY
        000400          UFE=    BIT8                    ;UNIT FIELD ERROR
        001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
        002000          PGE=    BIT10                   ;PROGRAMMING ERROR
        004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
        010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
        020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
        040000          WCE=    BIT14                   ;WRITE CHECK ERROR
        100000          DLT=    BIT15                   ;DATA LATE ERROR

                        .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)

        000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
        000002          SKI=    BIT1                    ;SEEK INCOMPLETE
        000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
        000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
        000020          FMTE=   BIT4                    ;FORMAT ERROR
        000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
        000100          ECH=    BIT6                    ;ECC HARD
        000200          BSE=    BIT7                    ;BAD SECTOR ERROR
        000400          HVRC=   BIT8                    ;HEADER VRC ERROR
        001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
        002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
        004000          WLE=    BIT11                   ;WRITE LOCK ERROR
        010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
        020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
        040000          UNS=    BIT14                   ;DRIVE UNSAFE
        100000          DCK=    BIT15                   ;DATA CHECK

                        .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)

        000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
                                                        ;    THIS BIT IS RESET)
        000004          OFST=   BIT2                    ;DRIVE OFFSET
        000010          ACLO=   BIT3                    ;AC LOW
        000020          DCLO=   BIT4                    ;DC LOW
        000040          DROT=   BIT5                    ;DRIVE OFF TRACK
        000100          VV=     BIT6                    ;VOLUME VALID
        000200          DRDY=   BIT7                    ;DRIVE READY
        000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06,1=RK07)
        004000          WRL=    BIT11                   ;WRITE LOCK
        020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
        040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
        100000          SVAL=   BIT15                   ;STATUS VALID

                        .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)

```

```
        000017          MESMSK= 17              ;MESSAGE MASK
        000020          PAT=    BIT4            ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
        000040          DMD=    BIT5            ;DIAGNOSTIC MODE
        000100          MSP=    BIT6            ;MAINTENANCE SECTOR PULSE
        000200          MIND=   BIT7            ;MAINTENANCE INDEX
        000400          MCLK=   BIT8            ;MAINTENANCE CLOCK
        001000          MERD=   BIT9            ;MAINTENANCE ENCODED READ DATA
        002000          MEWD=   BIT10           ;MAINTENANCE ENCODED WRITE DATA
        004000          PCA=    BIT11           ;PRECOMPENSATION ADVANCE
        010000          PCD=    BIT12           ;PRECOMPENSATION DELAY
        020000          ECCW=   BIT13           ;ECC WORD IS BEING READ OR WRITTEN
        040000          WRTGAT= BIT14           ;WRITE GATE
        100000          RDGATE= BIT15           ;READ GATE

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

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

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

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

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

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

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

```
000020          D.SECT= BIT4            ;SECTOR ERROR
000040          D.WCUR= BIT5            ;WRITE CURRENT AND NO WRITE GATE
000100          D.WGAT= BIT6            ;WRITE GATE AND NO TRANSISTIONS
000200          D.HDFL= BIT7            ;HEAD FAULT
000400          D.MHD=  BIT8            ;MULTIPLE HEAD SELECT
001000          D.XERR= BIT9            ;INDEX ERROR
002000          D.TIB=  BIT10           ;TRIBIT ERROR
004000          D.PLO=  BIT11           ;PLO ERROR
010000          D.NMOV= BIT12           ;SEEK AND NO MOTION
020000          D.LIMD= BIT13           ;LIMIT DETECT ON SEEK
040000          D.SUNS= BIT14           ;SERVO UNSAFE

                .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)

000007          M.DRV=  7               ;DRIVE CODE, ALL BYTES
077770          M.SER=  77770           ;DRIVE SERIAL #, BYTE 11

                .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)

000003          M.ID=   3               ;BYTE ID, ALL BYTES
040000          M.ALGN= BIT14           ;ALIGN SIGN, BYTE 10
000760          M.SECT= 760             ;SECTOR COUNT, BYTE 11
007000          M.HEAD= 7000            ;HEAD DECODE, BYTE 11
100000          M.PAR=  BIT15           ;PARITY, MESS A/B, ALL BYTES
```

```
                                   .SBTTL  TRAP CATCHER

                000000                     .=0
                                   ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                   ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                   ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                000174                     .=174
        000174  000000             DISPREG: .WORD  0                ;;SOFTWARE DISPLAY REGISTER
        000176  000000             SWREG:   .WORD  0                ;;SOFTWARE SWITCH REGISTER
                                   .SBTTL  STARTING ADDRESS(ES)
        000200  000137  012556             JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                000220                     .=220
        000220  000137  012546             JMP     PARSRT           ;INPUT ALL PARAMETERS & START TESTING

                000240                     .=240
        000240  000137  055500             JMP     0.ODT            ;ENTER ODT11

                                   .SBTTL  ACT11 HOOKS

                                   ;;***************************************************************
                                   ;HOOKS REQUIRED BY ACT11
                000244                     $SVPC=.                  ;SAVE PC
                000046                     .=46
        000046  031600                     $ENDAD                   ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                     .=52
        000052  100000                     .WORD   100000           ;;2)SET LOC.52 TO 100000
                000244                     .=$SVPC                  ;; RESTORE PC
                001000                     .=1000
                                   .SBTTL  APT PARAMETER BLOCK

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

        001000                     $APTHD:
        001000  000000             $HIBTS:  .WORD  0        ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
        001002  001210             $MBADR:  .WORD  $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
        001004  000430             $TSTM:   .WORD  280.     ;;RUN TIM OF LONGEST TEST
        001006  001130             $PASTM:  .WORD  600.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
        001010  001130             $UNITM:  .WORD  600.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        001012  000042                      .WORD  $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)


                                            .LIST   MD

                                   ;
```

```
;USE LOOP X TO OMIT SUBCLR
;

.MACRO  LOOP    A
        SCOP1
        MOV     #STACK,SP               ;RESTORE STK PTR

.IF     B       A
        JSR     PC,SUBCLR
        ERROR   24                      ;CERR AFTER SCLR

.ENDC
.ENDM   LOOP

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

        MOV     #<A!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
.ENDM   F.EAB

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

        JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
        .WORD   G!H!I                   ;& MSGS SPECIFIED HERE
        ERROR   A                       ;MSG A0 ERROR F
        ERROR   C                       ;MSH B0 ERROR
        ERROR   D                       ;MSG A1 ERROR
        ERROR   E                       ;MSG B1 ERROR
.ENDM   CHECK

;
;A=CYL DIFF/OFFSET ERROR #
;B=CYL ADDR ERROR #
;C=<ERROR DESCRIPTION>
;
.MACRO  CWD2    A,B,C,?D,?E

```

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

.MACRO	DRCLR	?A

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

.ENDM	DRCLR


;
;USE CALIB	X	TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
;
.MACRO	CALIB	A,?C

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

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

.ENDM	CALIB

;
;QUICK START SPINDLE
```

```
;
.MACRO  QKSRT

        JSR     PC,SUBCLR
        ERROR   24                      ;CERR AFTER SCLR

        MOV     #SRTSPL,HCS1
        JSR     PC,DOCMD                ;DO START SPINDLE CMD & GET CONTR RDY
        ERROR   121                     ;RDY NOT SET AFTER ST SPIN CMD.

        MOV     T100,TEMP2              ;SETUP TIMEOUT
        JSR     PC,FATT1                ;FIND ATTN
        ERROR   74                      ;NO ATTN AFTER ST SPIN CMD.

        CLR     UNLD

.ENDM   QKSRT

;
; A=WRHEAD/<CFMT!WRHEAD>
; USE WRHDR     <A>,X    TO OMIT CHECKING A0, B0, A1 & B1
;
.MACRO  WRHDR   A,C,K,?D

        MOV     #<A>,HCS1
        JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
        ERROR   200                     ;NO RDY AFTER WRITE HEADER CMD
.IF     B       K
        JSR     PC,GSTAT                ;GET FRESH STATUS
.ENDC
.IF     NB      K
        MOV     #<CFMT!SELDRV>,HCS1
        JSR     PC,DOCMD
        ERROR   117                     ;NO RDY AFTER SELDRV CMD
.ENDC
        BIT     #CERR,HCS1
        BEQ     D
        ERROR   201                     ;CERR AFTER WRITE HEADER CMD
        TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
        JMP     $EOP
D:
.IF B   C
        F.EAB   0
        CHECK   277,267,300,270,<AFTER WRITE HEADER CMD>,0,0,0
.ENDC

.ENDM   WRHDR

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

        MOV     #RHTAB,R0
        MOV     #<A>,HCS1
```

```
                JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
                ERROR   171                ;NO RDY AFTER READ HEADER CMD
                BIT     #CERR,HCS1
                BEQ     D
                ERROR   174                ;CERR AFTER READ HEADER CMD
                TYPE    ,MSG26             ;ABORTING BAL OF TESTS
                JMP     $EOF

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


                BIT     #DLT,RKCS2(R5)
                BEQ     E
                JSR     PC,GSTAT
                ERROR   173                ;DLT AFTER READ HEADER CMD
                TYPE    ,MSG26             ;ABORTING BAL OF TESTS
                JMP     $EOP
E:
.IF             B       C
                F.EAB   0
                CHECK 301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
.ENDC

.ENDM           RDHDR


.MACRO          HDCHK3  ?A

                RDHDR   RDHEAD
                CMP     RHTAB,TOCYL        ;CHECK WORD 0 ONLY, CYL#
                BEQ     A                  ;BR IF SAME
                ERROR   51                 ;WRONG CYL# ON HEADER
A:

.ENDM           HDCHK3

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

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

.ENDM           HDTBL

;
;QUICK SEEK.    ENTER WITH CYL# IN RKDC
;
.MACRO          QKSEEK  ?A

                MOV     #SEEK,HCS1
```

```
		JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
		ERROR	131			;NO RDY AFTER SEEK CMD

		MOV	T50000,TEMP1		;SETUP TIMEOUT
		JSR	PC,FATT2		;FIND ATTN
		ERROR	132			;NO ATTN AFTER SEEK CMD

		BIT	#CERR,HCS1
		BEQ	A
		ERROR	210			;CERR AFTER SEEK CMD

A:

.ENDM	QKSEEK


;
;A=WRDATA/<CFMT!WRDATA>
;C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
;D=ADDR TO JMP TO BYPASS TEST
;E: IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
;E: IF NON BLANK WILL OMIT CHECKING A0 THRU B1
;

.MACRO	WDATA	A,C,D,E,J,K,?F,?G,?H,?I

		MOV	#<A>,HCS1
		JSR	PC,DATCMD		;DO DATA X FOR CMD & GET CONTR RDY
		ERROR	11			;NO RDY AFTER WRITE DATA CMD
.IF		B	K
		JSR	PC,GSTAT		;GET FRESH STATUS
.ENDC
.IF		NB	K
		MOV	#<CFMT!SELDRV>,HCS1
		JSR	PC,DOCMD
		ERROR	117			;NO RDY AFTER SELDRV CMD
.ENDC
		BIT	#CERR,HCS1
		BEQ	I			;BR IF NO ERRORS

		BIT	#BSE,HER		;SEE IF BAD SECTOR FLAG
		BEQ	G			;BR IF NO
		JSR	PC,TRUERR		;ELSE SEE IF SECTOR LISTED IN BSE TABLE
		BR	H			;RETURN HERE IF NO
.IF		B	J

		INC	SECTOR			;RETURN HERE IF YES
		CMP	SECTOR,#10.		;ARE 10 CONSEC. SECTORS BAD
		BNE	F			;BR IF NO
		ERROR	46			;ABORTING TEST DETECTED 10 BAD SECTORS
		JMP	D			;BYPASS TEST
F:		MOV	#CCLR,RKCS1(R5)		;TRY ANOTHER SECTOR
		JMP	C
.ENDC
.IF		NB	J
		JMP	3$			;RET HERE IF YES
```

```
.ENDC
G:      ERROR   12                      ;CERR WITH WRITE DATA CMD
        F.EAB   0
        CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
        TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
        JMP     $EOP
H:      ERROR   63                      ;BAD SECTOR NOT LISTED IN TABLE
I:
.IF     B       E
        F.EAB   0
        CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
.ENDC
.ENDM   WDATA


;
;A=RDDATA/<CFMT!RDDATA>
;USE RDATA      <A>,X TO OMIT CHECKING A0, B0, A1 & B1
;

.MACRO  RDATA   A,C,K,?D,?E,?F,?G,?H

        MOV     #<A>,HCS1
        JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
        ERROR   13                      ;NO RDY AFTER READ DATA CMD
.IF     B       K
        JSR     PC,GSTAT                ;GET FRESH STATUS
.ENDC
.IF     NB      K
        MOV     #<CFMT!SELDRV>,HCS1
        JSR     PC,DOCMD
        ERROR   117                     ;NO RDY AFTER SELDRV CMD
.ENDC
        BIT     #CERR,HCS1
        BEQ     G
        BIT     #BSE,HER                ;SEE IF BAD SECTOR
        BEQ     E
        ERROR   65                      ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
        BR      H
D:      TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
        JMP     $EOP

E:      BIT     #DCK,HER                ;SEE IF DATA CHECK ERROR
        BEQ     F
        ERROR   21                      ;DATA CHECK ERROR AFTER READ CMD (ECC)
        BR      H

F:      ERROR   14                      ;CERR AFTER READ DATA CMD.

H:      F.EAB   0
        CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
        BR      D
G:
.IF     B       C
        F.EAB   0
        CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
```

```
	.ENDC
	.ENDM	RDATA


;
;A=WRTCHK/<CFMT!WRTCHK>
;C=EXPECTED DATA FOR TYPEOUT
;USE WRCHK	<A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
;

	.MACRO	WRCHK	A,C,D,K,?E,?F

		MOV	#<A>,HCS1
		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
		ERROR	15		;NO RDY AFTER WRITE CHECK CMD
	.IF	B	K
		JSR	PC,GSTAT	;GET FRESH STATUS
	.ENDC
	.IF	NB	K
		MOV	#<CFMT!SELDRV>,HCS1
		JSR	PC,DOCMD
		ERROR	117		;NO RDY AFTER SELDRV CMD
	.ENDC
		BIT	#CERR,HCS1
		BEQ	F
		BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
		BEQ	E
		MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
		MOV	C,WD2		;EXPECTED WORD FOR TYPEOUT
		ERROR	16		;WCE AFTER WRITE CMD
		BR	F

E:		ERROR	22		;CERR AFTER WRITE CHECK CMD
		F.EAB	0
		CHECK	57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
		TYPE	,MSG26		;ABORTING BAL OF TESTS
		JMP	$EOP

F:
	.IF	B	D
		F.EAB	0
		CHECK	57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
	.ENDC
	.ENDM	WRCHK

	.MACRO	OFSET	?A

		MOV	#A,$ESCAPE
		MOV	#OFFSET,HCS1
		JSR	PC,DOCMD	;DO RECAL CMD & GET CONTR RDY
		ERROR	33		;NO RDY AFTER OFFSET CMD

		MOV	#<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
		CLR	E.B0
		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
```

```
                MOV     #1,E.B1
                CHECK   35,61,36,62,<DURING OFFSET CMD>,0,0,0

A:              CLR     $ESCAPE
                MOV     T5000,TEMP1             ;SETUP TIMEOUT
                JSR     PC,FATT2                ;FIND ATTN
                ERROR   34                      ;NO ATTN AFTER OFFSET CMD

                F.EAB   <D.DSC!D.OFF>
                CHECK   260,261,37,40,<AFTER OFFSET CMD>,T.A2,T.B2,0

.ENDM           OFSET


.MACRO          EOPGM

                SCOPE
                CLR     $ESCAPE
                MOV     #1,$TIMES
                MOV     #STACK,SP
                INC     $DEVCT                  ;INCR COUNT FOR # DRIVES CHECKED
                CMP     DRIVS,$DEVCT            ;ARE ALL DRIVES PRESENT TESTED?
                BEQ     $EOP1+2                 ;BR IF YES
                JMP     NUDRV                   ;ELSE TEST NEXT DRIVE PRESENT
$EOP1:          SCOPE
.ENDM           EOPGM
;
;A= ERROR #
;B = ERROR CONDITION
;
.MACRO          OFFDIR  A,B,?C,?D
                MOV     R2,RKASOF(R5)           ;REFRESH RKASOF

                BIT     #BIT7,R2
                BNE     C                       ;BR IF NEG OFFSET

                CMP     R2,CYLDIF               ;CHECK POS OFFSET
                BEQ     D
                ERROR   A                       ;OFFSET IN A2 NOT = RKASOF
                BR      D                       ;B

C:              CMP     R1,CYLDIF               ;CHECK NEG OFFSET
                BEQ     D
                ERROR   A                       ;OFFSET IN A2 NOT = RKASOF
                                                ;B
D:
.ENDM           OFFDIR

                .NLIST  MD
```

```
                                .SBTTL  COMMON TAGS

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

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

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

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

```
001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.

001330  000100          LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
001332  000104          PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.

        000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
        172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
001334  000000          TRAPPC: 0               ;PC FOR MEM CHECK ENABLE TRAP

001336  000000          PARAM:  0               ;1 FOR 220 START, NO DEFAULT
001340  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE

001342  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
                                                ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
001344  000000          FRCYL:  0               ;FROM CYLINDER
001346  000000          TOCYL:  0               ;TO CYLINDER
001350  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
001352  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
001354  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
001356  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
001360  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
001362  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE

001364  000074          HZ:     60.             ;60 FOR 60 CPS
                                                ;50 FOR 50 CPS
001366  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
                                                ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
001370  000000          SEC:    0               ;SECOND COUNTER
001372  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
001374  000000          SECNT:  0               ;SECTOR COUNT
001376  000000          PSEC:   0               ;PREVIOUS SECTOR
001400  000000          ESEC:   0               ;EXPECTED SECTOR
001402  000000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3

001404  000001          T1:     1               ;TIMEOUT CONSTANTS
001406  000012          T10:    10.
001410  000062          T50:    50.
001412  000764          T500:   500.
001414  000144          T100:   100.
001416  011610          T5000:  5000.
001420  141520          T50000: 50000.

001422  000077          CYL:    63.             ;CYLINDER NUMBERS USED IN
001424  000177                  127.            ;CURRENT CROSSOVER TEST FOR RK06
001426  000277                  191.
001430  000377                  255.
001432  000477                  319.
001434  000577                  383.

001436  000177          CYL7:   127.            ;FOR RK07
001440  000377                  255.
001442  000577                  383.
001444  000777                  511.
001446  001177                  639.
```

```
001450  001377                        767.

001452  000000              WD1:      0                      ;ACTUAL HEADER/DATA WORD
001454  000000              WD2:      0                      ;EXPECTED DATA WORD

001456  000000              OFFERR: 0                        ;SET WHEN WRITE CHECK ERROR ON OFFSET


001460  000000              HEAD:     0                      ;HEAD NUMBER
001462  000000              HEADA:    0                      ;HEAD # FROM H.B3, RT. JUSTIFIED
001464  000000              HD1:      0                      ;SHIFTED HEAD# FOR FORMATTER ROUTINE
001466  000000              FORMAT:   0                      ;FORMAT TYPE
001470  000000              FMT1:     0                      ;SHIFTED FORMAT FOR FORMATTER ROUTINE
001472  000000              WDCNT:    0                      ;WORD COUNT

001474  000000              DATA0:    0                      ;ALL 0'S
001476  052525              DATA01:   52525                  ;0101 PATT
001500  177777              DATA1:    177777                 ;ALL 1'S
001502  133467              DPAT1:    133467
001504  070627              DPAT2:    70627

001506  000000              WORD:     0                      ;HEADER/DATA WORD
001510  000000              HDWD:     0                      ;HEADER WORD FROM RKDB

001512  000000              BSERR:    0                      ;CANNOT READ BSE INFO WHEN SET
001514  000000              LIMERR:   0                      ;LIMIT DETECT ERROR FLAG
001516  000000              BYPCERR:0                        ;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
001520  000000              CHKFLG:   0                      ;WORDS TO BE TESTED

001522  000102              HDTAB:    .BLKW 66.              ;CALCULATED HEADER WORD TABLE
001726  000102              RHTAB:    .BLKW 66.              ;FILLED AFTER READ HEADER CMD
002132  000102              SRTTAB:   .BLKW 66.              ;ABOVE RHTAB SORTED STARTING FORM
                                                             ;SECTOR 0 BY SORT ROUTINE
002336  000400              BSE20H:   .BLKW 256.             ;20 SECTOR HARDWARE BSE INFO
003336  000400              BSE22H:   .BLKW 256.             ;22 SECTOR HARDWARE BSE INFO.
004336  000400              BSE20S:   .BLKW      256.        ;20 SECTOR SOFTWARE BSE INFO.
005336  000400              BSE22S:   .BLKW     256.         ;22 SECTOR SOFTWARE BSE INFO.
006336  000400              RDTAB:    .BLKW 256.             ;FILLED AFTER READ DATA CMD

007336  000000              UNLD:     0                      ;SET TO 0 IF HEADS ARE LOADED
                                                             ;SET TO 1 IF HEADS UNLOADED
007340  000000              BADHDR: 0                        ;SET TO 0 IF FORMATTING OK
                                                             ;SET TO 1 IF FORMATTING ALTERED
007342  000000              HPEND:    0                      ;SET TO 0 IF HALT NOT PENDING
                                                             ;SET TO 1 IF HALT PENDING

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


007344     001     002     004  ATTN:     .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
007347     010     020     040
007352     100     200
                                          .EVEN
```

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

007354  000000                  HCS1:   0                       ;HOLD RKCS1
007356  000000                  HCS2:   0                       ;HOLD RKCS2
007360  000000                  HWC:    0                       ;HOLD RKWC
007362  000000                  HBA:    0                       ;ETC.
007364  000000                  HDA:    0
007366  000000                  HDS:    0
007370  000000                  HER:    0
007372  000000                  HASOF:  0
007374  000000                  HDC:    0
007376  000000                  HDB:    0
007400  000000                  HMR1:   0
007402  000000                  HMR2:   0
007404  000000                  HMR3:   0
007406  000000                  HPOS:   0
007410  000000                  HPAT:   0


007412  000000                  TEMP1:  0                       ;TEMPORARY STORAGE.
007414  000000                  TEMP2:  0
007416  000000                  TEMP3:  0
007420  000000                  TEMP4:  0
007422  000000                  TEMP5:  0
                                ;
                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
                                ;
007424  000000                  H.A0:   0
007426  000000                  H.B0:   0
007430  000000                  H.A1:   0
007432  000000                  H.B1:   0
007434  000000                  H.A2:   0
007436  000000                  H.B2:   0
007440  000000                  H.A3:   0
007442  000000                  H.B3:   0
                                ;
                                ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
                                ;
007444  000000                  E.A0:   0
007446  000000                  E.B0:   0
007450  000000                  E.A1:   0
007452  000000                  E.B1:   0
007454  000000                  E.A2:   0
007456  000000                  E.B2:   0
007460  000000                  E.A3:   0
007462  000000                  E.B3:   0
                                ;
                                ;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.
                                ;
        000001                  T.A2=BIT0                       ;TEST MSG A2 IF SET
        000002                  T.B2=BIT1
        000004                  T.B3=BIT2
```

```
                          ;
                          ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
                          ;

007464  000000            DDUMP:  0            ;FLAG - SET WHEN IN DDP DUMP MODE
007466  000000            DDPCH:  0            ;FLAG - SET WHEN IN DDP CHAIN MODE
007470  000000            ACT11:  0            ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
007472  000000            PPTP:   0            ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
007474  000000            DRIVS:  0            ;CONTAINS THE NUMBER OF DRIVES PRESENT

                          ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
                          ;IS PRESENT AND IS TO BE TESTED.

007476  000000            DRIV0:  0            ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
007500  000000            DRIV1:  0            ;FOR DRIVE 1
007502  000000            DRIV2:  0            ;FOR DRIVE 2
007504  000000            DRIV3:  0            ;FOR DRIVE 3
007506  000000            DRIV4:  0            ;FOR DRIVE 4
007510  000000            DRIV5:  0            ;FOR DRIVE 5
007512  000000            DRIV6:  0            ;FOR DRIVE 6
007514  000000            DRIV7:  0            ;FOR DRIVE 7

007516  000000            LCLKF:  0            ;L-CLOCK FLAG PRESENT FLAG
007520  000000            PCLKF:  0            ;P-CLOCK FLAG PRESENT FLAG
007522  000000            DOTIM:  0            ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
007524  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


        007526                  $ERRTB:

                                ;ERROR 1
        007526  046213                  EM2             ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
        007530  051647                  DH1
        007532  053732                  DT1
        007534  054534                  DF1

                                ;ERROR 2
        007536  046432                  EM5             ;DETECTED MDS
        007540  051647                  DH1
        007542  053732                  DT1
        007544  054534                  DF1

                                ;ERROR 3
        007546  046453                  EM6             ;DETECTED UFE
        007550  051647                  DH1
        007552  053732                  DT1
        007554  054534                  DF1

                                ;ERROR 4
        007556  046474                  EM7             ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
        007560  051647                  DH1
        007562  053732                  DT1
        007564  054534                  DF1
                                ;ERROR 5
        007566  000000                  0
        007570  000000                  0
        007572  000000                  0
        007574  000000                  0

                                ;ERROR 6
        007576  046637                  EM9             ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
        007600  051647                  DH1
        007602  053732                  DT1
        007604  054534                  DF1

                                ;ERROR 7
        007606  046713                  EM10            ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
        007610  051647                  DH1
        007612  053732                  DT1
        007614  054534                  DF1
```

```
                                ;ERROR 10
007616  046776                          EM11            ;DRA & NED BOTH SET
007620  051647                          DH1
007622  053732                          DT1
007624  054534                          DF1
                                ;ERR 11
007626  047042                          EM12            ;NO RDY
007630  052632                          DH27            ;AFTER WRITE DATA CMD
007632  053732                          DT1
007634  054664                          DF10
                                ;ERR 12
007636  047436                          EM21            ;CERR SET
007640  052632                          DH27
007642  053732                          DT1
007644  054664                          DF10
                                ;ERR 13
007646  047042                          EM12            ;NO RDY
007650  052602                          DH26            ;AFTER READ DATA CMD
007652  053732                          DT1
007654  054664                          DF10
                                ;ERR 14
007656  047436                          EM21            ;CERR SET
007660  052602                          DH26
007662  053732                          DT1
007664  054664                          DF10
                                ;ERR 15
007666  047042                          EM12            ;NO RDY
007670  052762                          DH32            ;AFTER WRITE CHECK CMD
007672  053732                          DT1
007674  054664                          DF10
                                ;ERR 16
007676  050624                          EM80            ;WRITE CHECK ERROR SET
007700  052762                          DH32            ;AFTER WRITE CHECK CMD
007702  054044                          DT6
007704  054554                          DF3
                                ;ERR 17
007706  050663                          EM81            ;WRITE CHECK CMD NOT FUNCTIONING
007710  053511                          DH52            ;WITH INTENTIONAL MISCOMPARE
007712  053732                          DT1
007714  054664                          DF10
                                ;ERR 20
007716  050727                          EM82            ;READ DATA NOT COMPARE WITH WRITE DATA
007720  052602                          DH26            ;AFTER READ DATA CMD
007722  054044                          DT6
007724  054554                          DF3
                                ;ERR 21
007726  051001                          EM83            ;DATA CHECK ERROR
007730  052602                          DH26
007732  053732                          DT1
007734  054664                          DF10
                                ;ERR 22
007736  047436                          EM21            ;CERR SET
007740  052762                          DH32            ;AFTER WRITE CHECK CMD
007742  053732                          DT1
007744  054664                          DF10
```

```
                              ;ERR 23
007746  047353                        EM18          ;MSG B0 ERROR
007750  052632                        DH27          ;AFTER WRITE DATA CMD
007752  054270                        DT13
007754  055014                        DF21
                              ;ERROR 24
007756  047436                        EM21          ;CERR SET
007760  052452                        DH21          ;AFTER SCLR
007762  053732                        DT1
007764  054664                        DF10
                              ;ERR 25
007766  047415                        EM20          ;MSG B1 ERROR
007770  052632                        DH27
007772  054270                        DT13
007774  055014                        DF21
                              ;ERR 26
007776  047353                        EM18
010000  052602                        DH26          ;AFTER READ DATA CMD
010002  054270                        DT13
010004  055014                        DF21

                              ;ERROR 27
010006  047665                        EM24          ;VOL VALID NOT SET
010010  052374                        DH19          ;AFTER PACK CMD
010012  053732                        DT1
010014  054664                        DF10
                              ;ERR 30
010016  047415                        EM20          ;MSG B1 ERROR
010020  052602                        DH26          ;AFTER READ DATA CMD.
010022  054270                        DT13
010024  055014                        DF21
                              ;ERR 31
010026  047353                        EM18          ;MSG B0 ERROR
010030  052762                        DH32          ;AFTER WRITE CHECK CMD
010032  054270                        DT13
010034  055014                        DF21
                              ;ERR 32
010036  047415                        EM20          ;MSG B1 ERROR
010040  052762                        DH32
010042  054270                        DT13
010044  055014                        DF21
                              ;ERR 33
010046  047042                        EM12          ;CONTR NOT READY
010050  052532                        DH24          ;AFTER OFFSET CMD
010052  053732                        DT1
010054  054664                        DF10
                              ;ERR 34
010056  047100                        EM13          ;NO ATTN
010060  052532                        DH24
010062  053732                        DT1
010064  054664                        DF10
                              ;ERR 35
010066  047332                        EM17          ;MSG A0 ERROR
010070  053545                        DH53          ;DURING OFFSET COMMAND
010072  054270                        DT13
010074  055014                        DF21
```

```
                               ;ERR 36
010076  047374                          EM19              ;MSG A1 ERROR
010100  053545                          DH53
010102  054270                          DT13
010104  055014                          DF21
                               ;ERR 37
010106  047374                          EM19              ;MSG A1 ERROR
010110  052532                          DH24              ;AFTER OFFSET CMD
010112  054270                          DT13
010114  055014                          DF21
                               ;ERR 40
```

```
010116  047415                    EM20          ;MSG B1 ERROR
010120  052532                    DH24
010122  054270                    DT13
010124  055014                    DF21
                    ;ERR 41
010126  047122                    EM14          ;UNEXP MEM PCRITY TRAP
010130  052040                    DH8           ;TEST #, TRAP PC
010132  053772                    DT3
010134  054550                    DF2
                    ;ERR 42
010136  050340                    EM41          ;CYL ADDR IN B2 DID NOT REMAIN CLEARED
010140  052532                    DH24
010142  054350                    DT14
010144  055050                    DF22
                    ;ERR 43
010146  051127                    EM85
010150  052500                    DH22
010152  053732                    DT1
010154  054664                    DF10
                    ;ERR 44
010156  047160                    EM15          ;WCE AT CYL 411,TRK 2, SEC 21
010160  051647                    DH1
010162  053732                    DT1
010164  054600                    DF4
                    ;ERR 45
010166  051127                    EM85          ;OFFSET BIT IN RKMR2 CLEARED
010170  053456                    DH51          ;AFTER SEEK TO SELF
010172  053732                    DT1
010174  054664                    DF10
                    ;ERR 46
010176  047720                    EM25          ;DETECTED 10 BAD SECTORS
010200  052632                    DH27          ;AFTER WRITE DATA CMD.
010202  053732                    DT1
010204  054664                    DF10
                    ;ERROR 47
010206  050233                    EM39          ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
010210  052350                    DH17          ;AFTER RECAL CMD
010212  054350                    DT14
010214  055050                    DF22
                    ;ERROR 50
010216  050302                    EM40          ;CYL ADDR IN RKMR3 NOT CLEARED
010220  052350                    DH17          ;AFTER RECAL COMD
010222  054350                    DT14
010224  055050                    DF22
                    ;ERR 51
010226  051271                    EM93          ;WRONG CYL# IN HEADER WORD (MISPOSITION)
010230  052557                    DH25          ;AFTER SEEK CMD
010232  054224                    DT9
010234  054770                    DF20
                    ;ERR 52
010236  047332                    EM17          ;MSG A0 ERROR
010240  052632                    DH27          ;AFTER WRITE DATA CMD
010242  054270                    DT13
010244  055014                    DF21
                    ;ERR 53
010246  047374                    EM19          ;MSG A1 ERROR
```

```
 010250  052632                         DH27
 010252  054270                         DT13
 010254  055014                         DF21
                       ;ERR 54
 010256  047332                         EM17                    ;MSG A0 ERROR
 010260  052602                         DH26                    ;AFTER READ DATA CMD
 010262  054270                         DT13
 010264  055014                         DF21
                       ;ERROR 55
 010266  047100                         EM13                    ;NO ATTN
 010270  052350                         DH17                    ;AFTER RECAL CMD
 010272  053732                         DT1
 010274  054664                         DF10
                       ;ERR 56
 010276  047374                         EM19                    ;MSG A1 ERROR
 010300  052602                         DH26
 010302  054270                         DT13
 010304  055014                         DF21
                       ;ERR 57
 010306  047332                         EM17                    ;MSG A0 ERROR
 010310  052762                         DH32                    ;AFTER WRITE CHECK CMD
 010312  054270                         DT13
 010314  055014                         DF21
                       ;ERR 60
 010316  047374                         EM19                    ;MSG A1 ERROR
 010320  052762                         DH32
 010322  054270                         DT13
 010324  055014                         DF21
                       ;ERR 61
 010326  047353                         EM18                    ;MSG B0 ERROR
 010330  053545                         DH53                    ;DURING OFFSET CMD
 010332  054270                         DT13
 010334  055014                         DF21
                       ;ERR 62
 010336  047415                         EM20                    ;MSG B1 ERROR
 010340  053545                         DH53
 010342  054270                         DT13
 010344  055014                         DF21
                       ;ERR 63
 010346  047772                         EM26                    ;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
 010350  052632                         DH27                    ;AFTER WRITE DATA CMD
 010352  053732                         DT1
 010354  054664                         DF10
                       ;ERR 64
 010356  051230                         EM88                    ;DID NOT FIND SECTOR 0 FROM INDEX
 010360  053573                         DH54                    ;AFTER FORMAT CHANGE AND READY REC'D
 010362  053732                         DT1
 010364  054664                         DF10
                       ;ERR 65
 010366  050051                         EM27                    ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
 010370  051647                         DH1
 010372  053732                         DT1
 010374  054534                         DF1
                       ;ERR 66
 010376  000000                         0
 010400  000000                         0
```

```
010402  000000                  0
010404  000000                  0

                ;ERR 67
010406  000000                  0
010410  000000                  0
010412  000000                  0
010414  000000                  0

                ;ERROR 70
010416  000000                  0
010420  000000                  0
010422  000000                  0
010424  000000                  0

                ;ERR 71
010426  000000                  0
010430  000000                  0
010432  000000                  0
010434  000000                  0

                ;ERROR 72
010436  000000                  0
010440  000000                  0
010442  000000                  0
010444  000000                  0

                ;ERR 73
010446  000000                  0
010450  000000                  0
010452  000000                  0
010454  000000                  0

                ;ERR 74
010456  047100                  EM13            ;NO ATTN
010460  052134                  DH10            ;AT END OF HEAD LOADING
010462  053732                  DT1
010464  054664                  DF10

                ;ERR 75
010466  047460                  EM22            ;NO DRIVS IN $DEVM
010470  051647                  DH1
010472  053732                  DT1
010474  054534                  DF1

                ;ERR 76
010476  047565                  EM23            ;NO DRIVS ON BUSS
010500  051647                  DH1
010502  053732                  DT1
010504  054534                  DF1

                ;ERR 77
010506  000000                  0
010510  000000                  0
010512  000000                  0
010514  000000                  0

                ;ERR 100
010516  000000                  0
010520  000000                  0
010522  000000                  0
010524  000000                  0

                ;ERROR 101
010526  051352                  EM94            ;OFFSET NOT CLEARED
010530  053331                  DH47            ;AFTER READ HEADER WITH MOVEMENT
010532  054350                  DT14
```

```
010534  055050                          DF22
                        ;ERROR 102
010536  051406                          EM95            ;FORMAT NOT SET
010540  052632                          DH27            ;AFTER WRITE DATA CMD
010542  053732                          DT1
010544  054664                          DF10
                        ;ERR 103
010546  051406                          EM95
010550  052762                          DH32            ;AFTER WRITE CHECK CMD
010552  053732                          DT1
010554  054664                          DF10
                        ;ERR 104
010556  050233                          EM39            ;OFFSET NOT RESET
010560  053671                          DH57            ;AFTER WRITE CMD WITH OFFSET
010562  054350                          DT14
010564  055050                          DF22
                        ;ERR 105
010566  050302                          EM40            ;CYL ADDR NOT 0
010570  053671                          DH57
010572  054350                          DT14
010574  055050                          DF22
                        ;ERR 106
010576  051442                          EM96            ;CANNOT FIND SECTOR 23(8)
010600  051647                          DH1
010602  053732                          DT1
010604  054534                          DF1
                        ;ERR 107
010606  051473                          EM97            ;HEAD SWITCHING TOO LONG
010610  052632                          DH27            ;AFTER WRITE DTA CMD
010612  053732                          DT1
010614  054724                          DF15
                        ;ERR 110
010616  000000                          0
010620  000000                          0
010622  000000                          0
010624  000000                          0
                        ;ERR 111
010626  000000                          0
010630  000000                          0
010632  000000                          0
010634  000000                          0
                        ;ERR 112
010636  051560                          EM100           ;DRIVE OFF TRACK SET
010640  052632                          DH27            ;AFTER WRITE DATA CMD
010642  053776                          DT4
010644  054640                          DF6
                        ;ERR 113
010646  050170                          EM36            ;CYL ADDR IN RKMR3 INCORRECT
010650  052632                          DH27
010652  053776                          DT4
010654  054640                          DF6
                        ;ERROR 114
010656  051172                          EM86            ;OFFSET IN A2 NOT = RKASOF
010660  052532                          DH24            ;AFTER OFFSET CMD
010662  054350                          DT14
010664  055050                          DF22
```

```
                                ;ERR 115
010666  051172                          EM86
010670  052500                          DH22            ;AFTER DRIVE CLEAR CMD
010672  054350                          DT14
010674  055050                          DF22
                                ;ERROR 116
010676  047042                          EM12            ;CONT NOT RDY
010700  052374                          DH19            ;AFTER PACK CMD
010702  053732                          DT1
010704  054664                          DF10
                                ;ERROR 117
010706  047042                          EM12            ;CONT NOT RDY
010710  052417                          DH20            ;AFTER SEL DR CMD
010712  053732                          DT1
010714  054664                          DF10
                                ;ERROR 120
010716  047042                          EM12
010720  052452                          DH21            ;AFTER SUBSYS CLEAR
010722  053732                          DT1
010724  054664                          DF10
                                ;ERROR 121
010726  047042                          EM12
010730  052061                          DH9             ;AFTER START SPINDLE CMD
010732  053732                          DT1
010734  054664                          DF10
                                ;ERROR 122
010736  051615                          EM101           ;DID NOT GO TO CYL 10
010740  052703                          DH30            ;AFTER READ HEADER CMD
010742  054350                          DT14
010744  055050                          DF22
                                ;ERROR 123
010746  051172                          EM86            ;A2 OFFSET NOT = RKASOF
010750  053456                          DH51            ;AFTER SEEK TO SELF
010752  054350                          DT14
010754  055050                          DF22
                                ;ERROR 124
010756  047042                          EM12
010760  052350                          DH17            ;AFTER RECAL CMD
010762  053732                          DT1
010764  054664                          DF10
                                ;ERR 125
010766  050535                          EM73            ;CTO SET
010770  051036                          EM84            ;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
010772  053732                          DT1
010774  054614                          DF5
                                ;ERR 126
010776  050603                          EM79            ;NED SET
011000  051036                          EM84
011002  053732                          DT1
011004  054614                          DF5
                                ;ERR 127
011006  046432                          EM5             ;MDS SET
011010  051036                          EM84
011012  053732                          DT1
011014  054614                          DF5
                                ;ERROR 130
```

```
011016  000000          0
011020  000000          0
011022  000000          0
011024  000000          0
                ;ERROR 131
011026  047042          EM12            ;NO RDY
011030  052557          DH25            ;AFTER SEEK CMD
011032  053732          DT1
011034  054664          DF10
                ;ERROR 132
011036  047100          EM13            ;NO ATTN
011040  052557          DH25
011042  053732          DT1
011044  054664          DF10
                ;ERROR 133
011046  000000          0
011050  000000          0
011052  000000          0
011054  000000          0
                ;ERROR 134
011056  000000          0
011060  000000          0
011062  000000          0
011064  000000          0
                ;ERROR 135
011066  000000          0
011070  000000          0
011072  000000          0
011074  000000          0
                ;ERROR 136
011076  000000          0
011100  000000          0
011102  000000          0
011104  000000          0
                ;ERROR 137
011106  050233          EM39            ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
011110  052557          DH25
011112  053732          DT1
011114  054664          DF10
                ;ERR 140
011116  047332          EM17            ;MSG AO ERROR
011120  053456          DH51            ;AFTER SEEK TO SELF
011122  054270          DT13
011124  055014          DF21
                ;ERR 141
011126  047353          EM18
011130  053456          DH51
011132  054270          DT13
011134  055014          DF21
                ;ERR 142
011136  047374          EM19
011140  053456          DH51
011142  054270          DT13
011144  055014          DF21
                ;ERR 143
011146  047415          EM20
```

```
2657 011150 053456            DH51
2658 011152 054270            DT13
2659 011154 055014            DF21
2660                   ;ERROR 144
2661 011156 000000            0
2662 011160 000000            0
2663 011162 000000            0
2664 011164 000000            0
2665                   ;ERROR 145
2666 011166 000000            0
2667 011170 000000            0
2668 011172 000000            0
2669 011174 000000            0
2670                   ;ERROR 146
2671 011176 000000            0
2672 011200 000000            0
2673 011202 000000            0
2674 011204 000000            0
2675                   ;ERROR 147
2676 011206 000000            0
2677 011210 000000            0
2678 011212 000000            0
2679 011214 000000            0
2680                   ;ERROR 150
2681 011216 000000            0
2682 011220 000000            0
2683 011222 000000            0
2684 011224 000000            0
2685                   ;ERROR 151
2686 011226 047042            EM12                    ;NO RDY
2687 011230 052500            DH22                    ;AFTER CLEAR CMD
2688 011232 053732            DT1
2689 011234 054664            DF10
2690                   ;ERROR 152
2691 011236 000000            0
2692 011240 000000            0
2693 011242 000000            0
2694 011244 000000            0
2695                   ;ERROR 153
2696 011246 000000            0
2697 011250 000000            0
2698 011252 000000            0
2699 011254 000000            0
2700                   ;ERROR 154
2701 011256 050406            EM55                    ;ATTN NOT CLEARED
2702 011260 052500            DH22
2703 011262 053732            DT1
2704 011264 054664            DF10
2705                   ;ERROR 155
2706 011266 000000            0
2707 011270 000000            0
2708 011272 000000            0
2709 011274 000000            0
2710                   ;ERROR 156
2711 011276 000000            0
2712 011300 000000            0
```

```
011302  000000              0
011304  000000              0
                    ;ERROR 157
011306  000000              0
011310  000000              0
011312  000000              0
011314  000000              0
                    ;ERROR 160
011316  000000              0
011320  000000              0
011322  000000              0
011324  000000              0
                    ;ERROR 161
011326  000000              0
011330  000000              0
011332  000000              0
011334  000000              0
                    ;ERROR 162
011336  000000              0
011340  000000              0
011342  000000              0
011344  000000              0
                    ;ERROR 163
011346  000000              0
011350  000000              0
011352  000000              0
011354  000000              0
                    ;ERROR 164
011356  000000              0
011360  000000              0
011362  000000              0
011364  000000              0
                    ;ERROR 165
011366  000000              0
011370  000000              0
```

```
011372  000000          0
011374  000000          0
                ;ERROR 166
011376  000000          0
011400  000000          0
011402  000000          0
011404  000000          0
                ;ERROR 167
011406  000000          0
011410  000000          0
011412  000000          0
011414  000000          0
                ;ERROR 170
011416  000000          0
011420  000000          0
011422  000000          0
011424  000000          0
                ;ERROR 171
011426  047042          EM12                    ;NO RDY
011430  052703          DH30                    ;AFTER READ HEADER CMD
011432  053732          DT1
011434  054664          DF10
                ;ERROR 172
011436  000000          0
011440  000000          0
011442  000000          0
011444  000000          0
                ;ERROR 173
011446  050441          EM63                    ;DLT SET
011450  052703          DH30
011452  053732          DT1
011454  054724          DF15
                ;ERROR 174
011456  047436          EM21                    ;CERR SET
011460  052703          DH30
011462  053732          DT1
011464  054724          DF15
                ;ERROR 175
011466  050233          EM39                    ;CYL DIFF NOT CLEARED
011470  052134          DH10                    ;AT END OF HEAD LOADING
011472  053732          DT1
011474  054664          DF10
                ;ERROR 176
011476  050302          EM40                    ;CYL ADDR NOT CLEARED.
011500  052134          DH10
011502  053732          DT1
011504  054664          DF10
                ;ERROR 177
011506  000000          0
011510  000000          0
011512  000000          0
011514  000000          0
                ;ERROR 200
011516  047042          EM12                    ;NO RDY
011520  053014          DH39                    ;AFTER WRITE HEADER CMD
011522  053732          DT1
```

```
011524  054724                  DF15
                ;ERROR 201
011526  047436                  EM21                    ;CERR SET
011530  053014                  DH39
011532  053732                  DT1
011534  054724                  DF15
                ;ERROR 202
011536  050462                  EM65                    ;READ HEADER ERROR
011540  051647                  DH1
011542  054110                  DT7
011544  054704                  DF14
                ;ERROR 203
011546  000000                  0
011550  000000                  0
011552  000000                  0
011554  000000                  0
                ;ERROR 204
011556  000000                  0
011560  000000                  0
011562  000000                  0
011564  000000                  0
                ;ERROR 205
011566  000000                  0
011570  000000                  0
011572  000000                  0
011574  000000                  0
                ;ERROR 206
011576  000000                  0
011600  000000                  0
011602  000000                  0
011604  000000                  0
                ;ERROR 207
011606  050170                  EM36                    ;CYL ADDR IN RKMR3 INCORRECT
011610  052557                  DH25                    ;AFTER SEEK CMD
011612  053776                  DT4
011614  054640                  DF6
                ;ERROR 210
011616  047436                  EM21                    ;CERR SET
011620  052557                  DH25
011622  053732                  DT1
011624  054664                  DF10
                ;ERROR 211
011626  000000                  0
011630  000000                  0
011632  000000                  0
011634  000000                  0
                ;ERROR 212
011636  000000                  0
011640  000000                  0
011642  000000                  0
011644  000000                  0
                ;ERROR 213
011646  000000                  0
011650  000000                  0
011652  000000                  0
011654  000000                  0
```

```
                                ;ERROR 214
011656  000000                         0
011660  000000                         0
011662  000000                         0
011664  000000                         0
                                ;ERROR 215
011666  000000                         0
011670  000000                         0
011672  000000                         0
011674  000000                         0
                                ;ERROR 216
011676  000000                         0
011700  000000                         0
011702  000000                         0
011704  000000                         0
                                ;ERROR 217
011706  000000                         0
011710  000000                         0
011712  000000                         0
011714  000000                         0
                                ;ERROR 220
011716  000000                         0
011720  000000                         0
011722  000000                         0
011724  000000                         0
                                ;ERROR 221
011726  047332                         EM17                    ;MSG A0 ERROR
011730  052350                         DH17
011732  054270                         DT13
011734  055014                         DF21
                                ;ERROR 222
011736  047374                         EM19                    ;MSG A1 ERROR
011740  052350                         DH17
011742  054270                         DT13
011744  055014                         DF21
                                ;ERROR 223
011746  000000                         0
011750  000000                         0
011752  000000                         0
011754  000000                         0
                                ;ERROR 224
011756  000000                         0
011760  000000                         0
011762  000000                         0
011764  000000                         0
                                ;ERROR 225
011766  000000                         0
011770  000000                         0
011772  000000                         0
011774  000000                         0
                                ;ERROR 226
011776  047042                         EM12                    ;NO RDY
012000  052602                         DH26                    ;AFTER READ DATA CMD
012002  053732                         DT1
012004  054664                         DF10
                                ;ERROR 227
```

```
      012006  047436          EM21              ;CERR SET
      012010  052602          DH26
      012012  053732          DT1
      012014  054724          DF15

                    ;ERROR 230
      012016  047267          EM16              ;CANNOT READ BSE INFO
      012020  052200          DH13              ;ON SEC 10, 12, 14, 16, 18, 20
      012022  053732          DT1
      012024  054744          DF17
                    ;ERROR 231
      012026  047267          EM16
      012030  052264          DH14              ;ON SEC 11, 13, 15, 17, 19, 21
      012032  053732          DT1
      012034  054744          DF17
                    ;ERROR 232
      012036  000000          0
      012040  000000          0
      012042  000000          0
      012044  000000          0
                    ;ERROR 233
      012046  047267          EM16              ;CANNOT READ BSE INFO
      012050  053130          DH42              ;ON SECT 0,2,4,6,8
      012052  053732          DT1
      012054  054744          DF17
                    ;ERROR 234
      012056  047267          EM16
      012060  053201          DH43              ;ON SECT 1,3,5,7,9
      012062  053732          DT1
      012064  054744          DF17
                    ;ERROR 235
      012066  050504          EM69              ;ALIGN CARTRIDGE USED
      012070  053252          DH44              ;WILL BYPASS FORMAT & ALL R/W TESTS
      012072  053732          DT1
      012074  054664          DF10
                    ;ERROR 236
      012076  000000          0
      012100  000000          0
      012102  000000          0
      012104  000000          0
                    ;ERROR 237
      012106  000000          0
      012110  000000          0
      012112  000000          0
      012114  000000          0
                    ;ERROR 240
      012116  000000          0
      012120  000000          0
      012122  000000          0
      012124  000000          0
                    ;ERROR 241
      012126  000000          0
      012130  000000          0
      012132  000000          0
      012134  000000          0
                    ;ERROR 242
      012136  000000          0
```

```
012140  000000              0
012142  000000              0
012144  000000              0

                   ;ERROR 243
012146  050170              EM36        ;CYL ADDR IN RKMR3 INCORRECT
012150  052557              DH25        ;AFTER SEEK CMD
012152  054156              DT8
012154  054640              DF6
                   ;ERR 244
012156  050556              EM74        ;RTZ NOT SET
012160  053103              DH41        ;DURING RECAL CMD
012162  053732              DT1
012164  054664              DF10
                   ;ERR 245
012166  000000              0
012170  000000              0
012172  000000              0
012174  000000              0
                   ;ERR 246
012176  000000              0
012200  000000              0
012202  000000              0
012204  000000              0
                   ;ERR 247
012206  000000              0
012210  000000              0
012212  000000              0
012214  000000              0
                   ;ERR 250
012216  000000              0
012220  000000              0
012222  000000              0
012224  000000              0
                   ;ERR 251
012226  000000              0
012230  000000              0
012232  000000              0
012234  000000              0
                   ;ERR 252
012236  000000              0
012240  000000              0
012242  000000              0
012244  000000              0
                   ;ERR 253
012246  000000              0
012250  000000              0
012252  000000              0
012254  000000              0
                   ;ERR 254
012256  000000              0
012260  000000              0
012262  000000              0
012264  000000              0
                   ;ERR 255
012266  000000              0
```

```
012270  000000                     0
012272  000000                     0
012274  000000                     0
                       ;ERR 256
012276  000000                     0
012300  000000                     0
012302  000000                     0
012304  000000                     0
                       ;ERR 257
012306  000000                     0
012310  000000                     0
012312  000000                     0
012314  000000                     0
                       ;ERR 260
012316  047332                     EM17          ;MSG A0 ERROR
012320  052532                     DH24          ;AFTER OFFSET CMD
012322  054270                     DT13
012324  055014                     DF21
                       ;ERR 261
012326  047353                     EM18          ;MSG B0 ERROR
012330  052532                     DH24
012332  054270                     DT13
012334  055014                     DF21
                       ;ERR 262
012336  000000                     0
012340  000000                     0
012342  000000                     0
012344  000000                     0
                       ;ERR 263
012346  000000                     0
012350  000000                     0
012352  000000                     0
012354  000000                     0
                       ;ERR 264
012356  000000                     0
012360  000000                     0
012362  000000                     0
012364  000000                     0
                       ;ERR 265
012366  047353                     EM18          ;MSG B0 ERROR
012370  052500                     DH22          ;AFTER DRIVE CLEAR CMD
012372  054270                     DT13
012374  055014                     DF21
                       ;ERR 266
012376  047415                     EM20          ;MSG B1 ERROR
012400  052500                     DH22
012402  054270                     DT13
012404  055014                     DF21
                       ;ERR 267
012406  047353                     EM18          ;MSG B0 ERROR
012410  053014                     DH39          ;AFTER WRITE HEADER CMD
012412  054270                     DT13
012414  055014                     DF21
                       ;ERR 270
012416  047415                     EM20          ;MSG B1 ERROR
012420  053014                     DH39
```

```
012422  054270                  DT13
012424  055014                  DF21
                        ;ERR 271
012426  047353                  EM18
012430  052703                  DH30            ;AFTER RD. HDR. CMD.
012432  054270                  DT13
012434  055014                  DF21
                        ;ERR 272
012436  047415                  EM20
012440  052703                  DH30
012442  054270                  DT13
012444  055014                  DF21
                        ;ERR 273
012446  047332                  EM17            ;MSG A0 ERROR
012450  052500                  DH22            ;AFTER DRV CLR CMD
012452  054270                  DT13
012454  055014                  DF21
                        ;ERR 274
012456  047374                  EM19            ;MSG A1 ERROR
012460  052500                  DH22
012462  054270                  DT13
012464  055014                  DF21
                        ;ERR 275
012466  047353                  EM18            ;MSG B0 ERROR
012470  052350                  DH17            ;AFTER RECAL CMD
012472  054270                  DT13
012474  055014                  DF21
                        ;ERR 276
012476  047415                  EM20            ;MSG B1 ERROR
012500  052350                  DH17
012502  054270                  DT13
012504  055014                  DF21
                        ;ERR 277
012506  047332                  EM17            ;MSG A0 ERROR
012510  053014                  DH39            ;AFTER WRITE HEADER CMD
012512  054270                  DT13
012514  055014                  DF21
                        ;ERR 300
012516  047374                  EM19            ;MSG A1 ERROR
012520  053014                  DH39
012522  054270                  DT13
012524  055014                  DF21
                        ;ERR 301
012526  047332                  EM17
012530  052703                  DH30            ;AFT RD HDR. CMD
012532  054270                  DT13
012534  055014                  DF21
                        ;ERR 302
012536  047374                  EM19
012540  052703                  DH30
012542  054270                  DT13
012544  055014                  DF21
```

```
                                        .SBTTL  PROGRAM SETUP
012546  012737  000001  001336  PARSRT: MOV     #1,PARAM                ;SET FLAG FOR 220 START
012554  000402                          BR      PRGSRT                  ;START PROGRAM
012556  005037  001336          START:  CLR     PARAM                   ;CLEAR FOR 200 START
012562  000005                  PRGSRT: RESET                           ;CLEAR ALL INT ENABLE & INIT
012564  012706  001100                  MOV     #STACK,SP               ;SETUP STACK POINTER
012570  012746  000000                  MOV     #PR0,-(SP)              ;PSW LOADED TO BE
012574  012746  012602                  MOV     #1$,-(SP)               ;LSI-11 COMPATABLE
012600  000002                          RTI                             ;ENABLE ALL INTERRUPTS
012602  004737  041112          1$:     JSR     PC,$TKINT               ;SETUP KB VECTOR ADDR, PRIORITY 4
                                                                        ;& TURN ON KB INTERRUPT
                                        ;*** CPU PRIORITY LEVEL NOW AT 0 ***
                                        ;*** ANY DEVICE WHICH SETS ITS   ***
                                        ;*** INTERRUPT ENABLE BIT WILL   ***
                                        ;*** SERVICED.                   ***
                                        ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
                                        ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
                                        ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
                                        ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
                                                                        ;SYSMAC 'SETUP'
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
012606  012706  001100                  MOV     #$CMTAG,R6              ;;FIRST LOCATION TO BE CLEARED
012612  005026                          CLR     (R6)+                   ;;CLEAR MEMORY LOCATION
012614  022706  001140                  CMP     #SWR,R6 ;;DONE?
012620  001374                          BNE     .-6                     ;;LOOP BACK IF NO
012622  012706  001100                  MOV     #STACK,SP               ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
012626  012737  037220  000020          MOV     #$SCOPE,@#IOTVEC        ;;IOT VECTOR FOR SCOPE ROUTINE
012634  012737  000340  000022          MOV     #340,@#IOTVEC+2         ;;LEVEL 7
012642  012737  037500  000030          MOV     #$ERROR,@#EMTVEC        ;;EMT VECTOR FOR ERROR ROUTINE
012650  012737  000340  000032          MOV     #340,@#EMTVEC+2         ;;LEVEL 7
012656  012737  043330  000034          MOV     #$TRAP,@#TRAPVEC        ;;TRAP VECTOR FOR TRAP CALLS
012664  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
012672  012737  036754  000024          MOV     #$PWRDN,@#PWRVEC        ;;POWER FAILURE VECTOR
012700  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;;LEVEL 7
012706  013737  031546  031540          MOV     $ENDCT,$EOPCT           ;;SETUP END-OF-PROGRAM COUNTER
012714  005037  001174                  CLR     $TIMES                  ;;INITIALIZE NUMBER OF ITERATIONS
012720  005037  001176                  CLR     $ESCAPE                 ;;CLEAR THE ESCAPE ON ERROR ADDRESS
012724  112737  000001  001115          MOVB    #1,$ERMAX               ;;ALLOW ONE ERROR PER TEST
012732  012737  012732  001106          MOV     #.,$LPADR               ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
012740  012737  012740  001110          MOV     #.,$LPERR               ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
012746  013746  000004                  MOV     @#ERRVEC,-(SP)          ;;SAVE ERROR VECTOR
012752  012737  013006  000004          MOV     #64$,@#ERRVEC           ;;SET UP ERROR VECTOR
012760  012737  177570  001140          MOV     #DSWR,SWR               ;;SETUP FOR A HARDWARE SWICH REGISTER
```

```
012766  012737  177570  001142           MOV     #DDISP,DISPLAY    ;;AND A HARDWARE DISPLAY REGISTER
012774  022777  177777  166136           CMP     #-1,@SWR          ;;TRY TO REFERENCE HARDWARE SWR
013002  001012                           BNE     66$               ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                   ;;AND  THE HARDWARE SWR IS NOT = -1
013004  000403                           BR      65$               ;;BRANCH IF NO TIMEOUT
013006  012716  013014           64$:    MOV     #65$,(SP)         ;;SET UP FOR TRAP RETURN
013012  000002                           RTI
013014  012737  000176  001140   65$:    MOV     #SWREG,SWR        ;;POINT TO SOFTWARE SWR
013022  012737  000174  001142           MOV     #DISPREG,DISPLAY
013030  012637  000004           66$:    MOV     (SP)+,@#ERRVEC    ;;RESTORE ERROR VECTOR

013034  005037  001216                   CLR     $PASS             ;;CLEAR PASS COUNT
013040  132737  000200  001231           BITB    #APTSIZE,$ENVM    ;;TEST USER SIZE UNDER APT
013046  001403                           BEQ     67$               ;;YES,USE NON-APT SWITCH
013050  012737  001232  001140           MOV     #$SWREG,SWR       ;;NO,USE APT SWITCH REGISTER
013056                           67$:
013056  012737  013122  000004   MEMPAR: MOV     #1$,ERRVEC        ;SETUP TIMEOUT VECTOR
013064  012737  000340  000006           MOV     #PR7,ERRVEC+2

013072  012701  172100                   MOV     #MEMBAS,R1        ;ADDR OF MEM CSR
013076  005011                   3$:     CLR     (R1)              ;SEE IF CAN REFERENCE
013100  012711  000001                   MOV     #1,(R1)           ;SET ENABLE BIT IF YES
013104  012737  036656  000114           MOV     #MEMERR,MEMVEC    ;LOAD VECTOR IF NO TIMEOUT
013112  012737  000340  000116           MOV     #PR7,MEMVEC+2
013120  000401                           BR      2$

013122  022626                   1$:     CMP     (SP)+,(SP)+       ;ADJ STACK
013124  062701  000002           2$:     ADD     #2,R1             ;TRY NEXT CSR
013130  020127  172140                   CMP     R1,#MEMBAS+40     ;SEE IF TRIED ALL
013134  001360                           BNE     3$                ;BR IF NO
013136  012737  000006  000004           MOV     #ERRVEC+2,ERRVEC          ;RESTORE TRAP CATCHER
013144  005037  000006                   CLR     ERRVEC+2

013150  004737  031666                   JSR     PC,CLRFLG         ;CLEAR DDUMP THRU SIZFLG
013154  005037  001220                   CLR     $DEVCT
013160  005037  001222                   CLR     $UNIT


                                 ;
                                 ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
                                 ;

013164  005737  000042           START1: TST     42
013170  001015                           BNE     1$                ;BR IF AUTO
013172  004737  031706                   JSR     PC,TITLE          ;MANUAL, TYPE PROG ID
013176  123727  000041  000013           CMPB    41,#13            ;13=LOADED BY XXDP
013204  001011                           BNE     2$
013206  005237  007464                   INC     DDUMP             ;SET RK06 DUMP MODE FLAG
013212  104401  044307                   TYPE    ,MSG2             ;REPLACE DRO PACK W/SCRATCH & DO<CR>
013216  000000                           HALT                      ;HALT
013220  000137  013234                   JMP     ST2
013224  000137  013300           1$:     JMP     ST3
013230  005237  007472           2$:     INC     PPTP              ;SET ACT/APT/PTP DUMP MODE FLAG


                                 ;
                                 ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
                                 ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
```

```
                                        ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
                                        ;           EX:     DRIVES TO BE TESTED:  1,2,4<CR>
                                        ;

 013234  005737  001336                 ST2:    TST     PARAM
 013240  001002                                 BNE     1$                      ;BR IF 220 START
 013242  000137  013332                         JMP     ST4                     ;200 START, DEFAULT & SIZE THE BUSS
 013246  104401  044370                 1$:     TYPE    ,MSG3                   ;DRIVES TO BE TESTED
 013252  004737  031766                         JSR     PC,GDRVS                ;GET DR NOS.
 013256  104401  044422                         TYPE    ,MSG4                   ;BUSS ADDR
 013262  004737  032126                         JSR     PC,GBA                  ;GET BA
 013266  104401  044467                         TYPE    ,MSG5                   ;CONT INT VECTOR
 013272  004737  032154                         JSR     PC,GINT                 ;GET INT VECTOR
 013276  000427                                 BR      ST5

                                        ;
                                        ;AUTO MODE
                                        ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
                                        ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
                                        ;ON THE BUSS
                                        ;

 013300  123727  000041  000013         ST3:    CMPB    41,#13                  ;13=LOADED BY XXDP
 013306  001007                                 BNE     1$
 013310  005237  007466                         INC     DDPCH                   ;SET RK06 CHAIN MODE FLAG
 013314  004737  031706                         JSR     PC,TITLE
 013320  104401  044604                         TYPE    ,MSG7                   ;DR0 NOT TSTD
 013324  000402                                 BR      ST4
 013326  005237  007470                 1$:     INC     ACT11                   ;SET ACT AUTO FLAG.

 013332  012737  177440  001264         ST4:    MOV     #177440,$BASE           ;DEFAULT VALUE
 013340  012737  000210  001314                 MOV     #210,RKVEC              ;DEFAULT VALUE
 013346  004737  032206                         JSR     PC,SETINT
 013352  005237  007524                         INC     SIZFLG                  ;DO "SIZE THE BUSS" TEST

 013356  005037  007336                 ST5:    CLR     UNLD                    ;INITIALIZE FLAGS
 013362  005037  007340                         CLR     BADHDR                  ;USED IN 'STOP' ROUTINE
 013366  005037  007342                         CLR     HPEND                   ;FOR VALID PROGRAM HALTS
 013372  005037  001176                         CLR     $ESCAPE
 013376  005037  001170                         CLR     $TMP4                   ;CLEAR RK07 FLAG
 013402  012737  007476  001342                 MOV     #DRIV0,DRVPTR           ;SETUP
 013410  005037  001220                         CLR     $DEVCT                  ;NO. OF DRVS DONE
 013414  005037  001222                         CLR     $UNIT                   ;CURRENT DRV UNDER TEST
 013420  012737  013466  000004                 MOV     #1$,ERRVEC              ;SETUP TIMEOUT ERROR VECTOR
 013426  005777  165674                         TST     @LKS                    ;SEE IF L-CLOCK THERE
 013432  005237  007516                         INC     LCLKF                   ;PRESENT, SET FLAG.
 013436  013700  001330                         MOV     LCVEC,R0                ;VECTOR ADDR
 013442  012737  013530  000004                 MOV     #2$,ERRVEC
 013450  005777  165644                         TST     @PKS                    ;SEE IF P-CLOCK THERE
 013454  005237  007520                         INC     PCLKF                   ;PRESENT, SET FLAG
 013460  013700  001332                         MOV     PCVEC,R0                ;VECTOR ADDR
 013464  000412                                 BR      3$

 013466  022626                         1$:     CMP     (SP)+,(SP)+             ;L-CLOCK NOT THERE, CLEAR STACK
 013470  012737  013534  000004                 MOV     #4$,ERRVEC
 013476  005777  165616                         TST     @PKS                    ;SEE IF P-CLOCK THERE
```

```
013502  005237  007520                  INC     PCLKF           ;PRESENT, SET FLAG
013506  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
013512  005237  007522          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
013516  012720  036100                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
013522  012710  000300                  MOV     #PR6,(R0)
013526  000407                          BR      TST1            ;;GO TO NEXT TEST

013530  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
013532  000767                          BR      3$

013534  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
013536  005037  007522                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
013542  104401  045145                  TYPE    ,MSG13          ;HEAD SW. TEST BYPASSED

```

```
                                         .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'
                                         ;*
                                         ;*
                                         ;;*****************************************************************
 013546  000004                          TST1:    SCOPE
 013550  012737  000001  001174                   MOV     #1,$TIMES          ;;DO 1 ITERATION
 013556  012706  001100                           MOV     #STACK,SP          ;RESTORE STK PTR

 013562  012746  000000                           MOV     #PR0,-(SP)         ;RESET PSW TO PRIORITY 0
 013566  012746  013574                           MOV     #5$,-(SP)          ;& MAKE IT LSI COMPATABLE
 013572  000002                                   RTI
 013574                                  5$:
```

```
  3362 013574 012737 013720 000004          MOV     #1$,ERRVEC          ;SETUP TIMOUT ERROR VECTOR
```

```
013602  013705  001264                  MOV     $BASE,R5        ;SETUP INDEX REG.
013606  005765  000000                  TST     RKCS1(R5)       ;REFERENCE ALL THE
013612  005765  000010                  TST     RKCS2(R5)       ;CONTROLLER REGISTERS
013616  005765  000002                  TST     RKWC(R5)
013622  005765  000004                  TST     RKBA(R5)
013626  005765  000006                  TST     RKDA(R5)
013632  005765  000012                  TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
013636  005765  000014                  TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
013642  005765  000016                  TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
013646  005765  000020                  TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
013652  005765  000024                  TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
013656  005765  000026                  TST     RKMR1(R5)
013662  005765  000034                  TST     RKMR2(R5)
013666  005765  000036                  TST     RKMR3(R5)
013672  005765  000030                  TST     RKECPS(R5)
013676  005765  000032                  TST     RKECPT(R5)

013702  012737  036570  000004          MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
013710  012737  000340  000006          MOV     #PR7,ERRVEC+2
013716  000404                          BR      TST2            ;;GO TO NEXT TEST

013720  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
013722  104007                          ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
013724  000137  031512                  JMP     $EOP1


                                ;;*****************************************************************
                                ;*TEST 2         SIZE THE BUSS
                                ;*
                                ;*        THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                                ;*        EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                                ;*        MANUAL MODE.
                                ;*        EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
                                ;*        CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
                                ;*        DRIVE WILL BE TESTED AS AN RK06.  IF SET, THE PROGRAM WILL BYPASS
                                ;*        TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                ;*        MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                ;*        DICATING THE OTHER PORT IS ACCESSED.
                                ;*        IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS RK07
                                ;*
                                ;;*****************************************************************
013730  000004                  TST2:   SCOPE
013732  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
013740  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

013744  005237  001516                  INC     BYPCERR         ;DO NOT TEST CERR IN 'FRDY'


013750  132737  000200  001231          BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
013756  001002                          BNE     14$             ;BR IF YES
013760  000137  014100                  JMP     12$             ;ELSE DO NORM SIZING OR VERIFY

013764  104401  044713          14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
013770  005037  007474                  CLR     DRIVS           ;# OF DRIVES PRESENT
013774  005000                          CLR     R0              ;DRV ADDR
013776  012701  007476                  MOV     #DRIV0,R1       ;DRV FLAG
```

```
        014002  013702  001266                    MOV     $DEVM,R2                ;APT DEVICE MAP

        014006  032702  000001          15$:      BIT     #BIT0,R2                ;SEE IF DRV IN DEVICE MAP
        014012  001410                            BEQ     16$                     ;BR IF NO
        014014  005237  007474                    INC     DRIVS                   ;ELSE INCR DRIVE COUNT
        014020  005211                            INC     (R1)                    ;& SET DRIVE PRESENT FLAG
        014022  104401  001205                    TYPE    ,$CRLF
        014026  010046                            MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
                                                                                  ;;TYPE DRIVE #
        014030  104403                            TYPOS                           ;;GO TYPE--OCTAL ASCII
        014032     001                            .BYTE   1                       ;;TYPE 1 DIGIT(S)
        014033     000                            .BYTE   0                       ;;SUPPRESS LEADING ZEROS

        014034  005721                  16$:      TST     (R1)+                   ;ADV POINTER TO NEXT FLAG
        014036  005200                            INC     R0                      ;INC DRIVE #
        014040  022700  000010                    CMP     #8.,R0                  ;ALL 8 TESTED?
        014044  001402                            BEQ     17$                     ;BR IF YES

        014046  006002                            ROR     R2                      ;ELSE GET NEXT BIT OFF DEVICE MAP
        014050  000756                            BR      15$                     ;& TRY AGAIN

        014052  005737  007474          17$:      TST     DRIVS                   ;SEE IF MORE DRIVES PRESENT
        014056  001402                            BEQ     18$                     ;BR IF NO
        014060  000137  014554                    JMP     VERIFY                  ;ELSE EXIT TEST & SETUP FOR RK07'S

        014064  104075                  18$:      ERROR   75                      ;NO DRIVES FOUND IN $DEVM
        014066  000000                            HALT                            ;SETUP CORRECTLY & PRESS 'CONTINUE'
        014070  000137  013356                    JMP     ST5                     ;TO TRY AGAIN
        014074  000137  014554          20$:      JMP     VERIFY                  ;DO NOT SIZE, GO TO NEXT TEST

        014100  012765  000040  000010  12$:      MOV     #SCLR,RKCS2(R5)         ;SUBSYSTEM CLEAR
        014106  013737  001406  007412            MOV     T10,TEMP1               ;SET TIMEOUT
        014114  004737  032320                    JSR     PC,FRDY                 ;FIND RDY
        014120  104120                            ERROR   120                     ;RDY NOT SET BY END OF SCLR
        014122  005737  007524                    TST     SIZFLG                  ;SIZE BUS?
        014126  001762                            BEQ     20$                     ;BR IF NO
        014130  104401  044713                    TYPE    ,MSG10                  ;WILL TEST DRIVES
        014134  005037  007474                    CLR     DRIVS                   ;# OF DRIVES PRESENT
        014140  005000                            CLR     R0                      ;DRV ADDR
        014142  012701  007476                    MOV     #DRIV0,R1               ;DRV FLAG
        014146                          1$:
        014146  104415                            SCOP1
        014150  012706  001100                    MOV     #STACK,SP               ;RESTORE STK PTR

        014154  012765  000040  000010            MOV     #SCLR,RKCS2(R5)         ;SUBSYSTEM CLEAR
        014162  013737  001406  007412            MOV     T10,TEMP1               ;SET TIMEOUT
        014170  004737  032320                    JSR     PC,FRDY                 ;FIND RDY
        014174  104120                            ERROR   120                     ;RDY NOT SET BY END OF SCLR
        014176  010065  000010                    MOV     R0,RKCS2(R5)            ;SELECT THE DRIVE ADDR
        014202  012737  000001  007354            MOV     #SELDRV,HCS1
        014210  053737  001170  007354            BIS     $TMP4,HCS1              ;ADD CDT IF RK07
        014216  013765  007354  000000            MOV     HCS1,RKCS1(R5)          ;GET STATUS
        014224  013737  001420  007412            MOV     T50000,TEMP1
        014232  004737  033010                    JSR     PC,DLY                  ;DO DELAY TO CATCH MDS
        014236  013737  001406  007412            MOV     T10,TEMP1
        014244  004737  032320                    JSR     PC,FRDY                 ;FIND RDY
```

```
014250  104117                          ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
014252  032737  100000  007354          BIT     #CERR,HCS1
014260  001052                          BNE     2$
014262  013737  007402  007412          MOV     HMR2,TEMP1
014270  042737  177770  007412          BIC     #↑C<DRVMSK>,TEMP1
014276  020037  007412                  CMP     R0,TEMP1        ;S/B SAME
014302  001020                          BNE     3$
014304  005700                          TST     R0
014306  001003                          BNE     4$
014310  005737  007466                  TST     DDPCH           ;SEE IF XXDP CHAIN MODE
014314  001016                          BNE     5$
014316  005237  007474          4$:     INC     DRIVS           ;INC DRIVE COUNT.
014322  005211                          INC     (R1)            ;SET DRIVE PRESENT FLAG
014324  053711  001170                  BIS     $TMP4,(R1)      ;ADD CDT IF RK07
014330  104401  001205                  TYPE    ,$CRLF
014334  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DR #
014336  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
014340     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
014341     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
014342  000403                          BR      5$

014344  004737  033026          3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
014350  104001                          ERROR   1               ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #

014352  005721                  5$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR. FLAG
014354  005200                          INC     R0              ;INC DR #
014356  005037  001170                  CLR     $TMP4           ;CLEAR RK07 FLAG
014362  022700  000010                  CMP     #8.,R0
014366  001267                          BNE     1$              ;MORE LEFT.
014370  005737  007474                  TST     DRIVS
014374  001065                          BNE     10$
014376  104076                          ERROR   76              ;NO DRIVES FOUND ON BUSS
014400  000000                          HALT                    ;SETUP CORRECTLY
014402  000137  013356                  JMP     ST5             ;AND PRESS 'CONTINUE'
014406  032737  000040  007370  2$:     BIT     #DTYE,HER
014414  001405                          BEQ     13$
014416  012737  002000  001170          MOV     #CDT,$TMP4      ;ADD CDT
014424  000137  014146                  JMP     1$              ;TRY AGAIN

014430  032737  001000  007356  13$:    BIT     #MDS,HCS2
014436  001015                          BNE     6$
014440  032737  000400  007356          BIT     #UFE,HCS2
014446  001015                          BNE     7$
014450  032737  000001  007366          BIT     #DRA,HDS
014456  001015                          BNE     8$
014460  032737  010000  007356          BIT     #NED,HCS2
014466  001424                          BEQ     9$
014470  000730                          BR      5$

014472  004737  033026          6$:     JSR     PC,BYP          ;TYPE BYP DR #
014476  104002                          ERROR   2               ;MDS DETECTED
014500  000724                          BR      5$

014502  004737  033026          7$:     JSR     PC,BYP
014506  104003                          ERROR   3               ;UFE DETECTED
```

```
014510  000720                                  BR      5$
014512  032737  010000  007356  8$:             BIT     #NED,HCS2
014520  001676                                  BEQ     4$
014522  104401  045266                          TYPE    ,MSG15                  ;DRV#
014526  010046                                  MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
                                                                                ;;TYPE DR#
014530  104403                                  TYPOS                           ;;GO TYPE--OCTAL ASCII
014532     001                                  .BYTE   1                       ;;TYPE 1 DIGIT(S)
014533     000                                  .BYTE   0                       ;;SUPPRESS LEADING ZEROS
014534  104010                                  ERROR   10                      ;DRA & NED BOTH SET
014536  000705                                  BR      5$
014540  004737  033026          9$:             JSR     PC,BYP
014544  104004                                  ERROR   4                       ;NO DRA & NO NED = OTHER PORT SELECTED
014546  000701                                  BR      5$
014550  000137  015146          10$:            JMP     NUDRV
014554                                  VERIFY:
                                        ;;*******************************************************
                                        ;*TEST 3         VERIFY OPERATOR DRIVE SELECTIONS
                                        ;*
                                        ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
                                        ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
                                        ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
                                        ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.
                                        ;*      IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
                                        ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                                        ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                                        ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
                                        ;*      VERIFY IT WAS NOT SPECIFIED.
                                        ;*      IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.
                                        ;*
                                        ;;*******************************************************
014554  000004                          TST3:   SCOPE
014556  012737  000001  001174                  MOV     #1,$TIMES               ;;DO 1 ITERATION
014564  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR
014570  005000                                  CLR     R0                      ;DRIVE ADDR
014572  012701  007476                          MOV     #DRIV0,R1               ;DRIVE FLAG
014576                                  1$:
014576  104415                                  SCOP1
014600  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR
014604  012765  000040  000010                  MOV     #SCLR,RKCS2(R5)         ;SUBSYSTEM CLEAR
014612  013737  001406  007412                  MOV     T10,TEMP1               ;SET TIME OUT
014620  004737  032320                          JSR     PC,FRDY                 ;FIND RDY
014624  104120                                  ERROR   120                     ;NO RDY AFTER SCLR
014626  010065  000010                          MOV     R0,RKCS2(R5)            ;DRV ADDR
014632  012737  000001  007354                  MOV     #SELDRV,HCS1
014640  053737  001170  007354                  BIS     $TMP4,HCS1              ;ADD CDT IF RK07
014646  013765  007354  000000                  MOV     HCS1,RKCS1(R5)          ;GET STATUS
014654  013737  001420  007412                  MOV     T5000,TEMP1
014662  004737  033010                          JSR     PC,DLY                  ;DO DELAY TO CATCH MDS
014666  013737  001406  007412                  MOV     T10,TEMP1
014674  004737  032320                          JSR     PC,FRDY                 ;FIND RDY
014700  104117                                  ERROR   117                     ;NO RDY AFTER SELECT DRIVE CMD.
```

```
014702  032737  100000  007354           BIT     #CERR,HCS1
014710  001036                           BNE     2$
014712  013737  007402  007412           MOV     HMR2,TEMP1
014720  042737  177770  007412           BIC     #^C<DRVMSK>,TEMP1
014726  020037  007412                   CMP     R0,TEMP1                ;S/B SAME
014732  001014                           BNE     3$
014734  005711                   11$:    TST     (R1)
014736  001402                           BEQ     4$
014740  053711  001170                   BIS     $TMP4,(R1)              ;SET CDT IF RK07.
014744  005721                   4$:     TST     (R1)+                   ;SHIFT PTR TO NEXT DR FLAG
014746  005200                           INC     R0                      ;INC DR#
014750  005037  001170                   CLR     $TMP4                   ;CLEAR CDT FOR NEXT DRIVE
014754  022700  000010                   CMP     #8.,R0
014760  001306                           BNE     1$                      ;MORE LEFT
014762  000475                           BR      TST4                    ;;GO TO NEXT TEST

014764  004737  033026           3$:     JSR     PC,BYP                  ;TRY BYPASS DRIVE#
014770  104001                           ERROR   1                       ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
014772  005711                           TST     (R1)
014774  001763                           BEQ     4$                      ;BRANCH IF NOT SPEC BY INPUT
014776  005337  007474           12$:    DEC     DRIVS                   ;DECREMENT TOTAL DRIVS
015002  005011                           CLR     (R1)                    ;CLEAR DRIVE FLAG
015004  000757                           BR      4$

015006  032737  000040  007370   2$:     BIT     #DTYE,HER
015014  001405                           BEQ     13$
015016  012737  002000  001170           MOV     #CDT,$TMP4              ;ADD CDT
015024  000137  014576                   JMP     1$                      ;TRY AGAIN

015030  032737  001000  007356   13$:    BIT     #MDS,HCS2
015036  001027                           BNE     6$
015040  032737  000400  007356           BIT     #UFE,HCS2
015046  001027                           BNE     7$
015050  032737  000001  007366           BIT     #DRA,HDS
015056  001005                           BNE     8$
015060  032737  010000  007356           BIT     #NED,HCS2
015066  001423                           BEQ     9$
015070  000404                           BR      10$
015072  032737  010000  007356   8$:     BIT     #NED,HCS2
015100  001715                           BEQ     11$
015102  005711                   10$:    TST     (R1)
015104  001717                           BEQ     4$

015106  004737  033026                   JSR     PC,BYP                  ;TYPE BYPASS DRIVE#
015112  104006                           ERROR   6
015114  000730                           BR      12$

015116  004737  033026           6$:     JSR     PC,BYP                  ;TYPE BYPASS DRIVE#
015122  104002                           ERROR   2                       ;MDS DETECTED
015124  000724                           BR      12$

015126  004737  033026           7$:     JSR     PC,BYP
015132  104003                           ERROR   3                       ;UFE DETECTED
015134  000720                           BR      12$

015136  004737  033026           9$:     JSR     PC,BYP
```

```
015142  104004                          ERROR   4                ;DRA & NED RESET - OTHER PORT SELECTED
015144  000714                          BR      12$




                                ;
                                ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
                                ;DRIVE PRESENT
                                ;
                                ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
                                ;UNDER TEST

015146  005037  001516          NUDRV:  CLR     BYPCERR          ;ENTER HERE FROM LAST TEST
                                                                 ;ALLOW CHECKING CERR IN 'FRDY'
015152  005037  001170                  CLR     $TMP4            ;CLEAR RK07 FLAG

                                ;;*****************************************************************
                                ;*TEST 4        FIND NEXT DRIVE TO BE TESTED
                                ;*
                                ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                ;*      ADDRESS IN '$UNIT' & $TMP4 IS SET TO CDT IF DRIVE IS RK07.
                                ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                                ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
                                ;*
                                ;;*****************************************************************
015156  000004                  TST4:   SCOPE
015160  012737  000001  001174          MOV     #1,$TIMES        ;;DO 1 ITERATION
015166  012706  001100                  MOV     #STACK,SP        ;RESTORE STK PTR
015172  012737  000004  001214          MOV     #$TN-1,$TESTN
015200  012737  000004  001102          MOV     #$TN-1,$TSTNM

015206  005737  007474                  TST     DRIVS            ;ANY DRIVES PRESENT?
015212  001004                          BNE     4$               ;YES BRANCH
015214  104401  045432                  TYPE    ,MSG27           ;ALL DRIVES TESTED
015220  000137  031512                  JMP     $EOP1            ;NO, GO TO END

015224  013701  001342          4$:     MOV     DRVPTR,R1        ;ADDR OF NEXT DRIVE FLAG
015230  005737  001220                  TST     $DEVCT           ;IS FIRST DRIVE BEING CHECKED
015234  001402                          BEQ     2$               ;YES, BRANCH
015236  005237  001222          1$:     INC     $UNIT            ;INCR DRIVE ADDR TO NEXT DRIVE
015242  005711                  2$:     TST     (R1)             ;IS DRIVE PRESENT?
015244  001002                          BNE     5$               ;BR IF YES
015246  005721                          TST     (R1)+            ;ELSE FIND NEXT DRIVE
015250  000772                          BR      1$
015252  005737  007466          5$:     TST     DDPCH            ;DDP CHAIN MODE?
015256  001405                          BEQ     3$               ;BR IF NO
015260  005737  001222                  TST     $UNIT            ;ELSE SEE IF DRV 0
015264  001002                          BNE     3$               ;BR IF NO
015266  005721                          TST(R1)+                 ;ELSE FIND NEXT DRIVE PRESENT
015270  000762                          BR      1$

015272  032721  002000          3$:     BIT     #CDT,(R1)+       ;SEE IF DRIVE UNDER TEST IS RK07
015276  001403                          BEQ     6$               ;BR IF NO
015300  012737  002000  001170          MOV     #CDT,$TMP4       ;ELSE SET RK07 FLAG
015306  010137  001342          6$:     MOV     R1,DRVPTR        ;STORE POINTER TO NEXT DR FLAG
015312  104401  045266                  TYPE    ,MSG15           ;"DRIVE"
```

```
015316  013700  001222                      MOV     $UNIT,R0
015322  010046                              MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
                                                                            ;;DRIVE #
015324  104403                              TYPOS                           ;;GO TYPE--OCTAL ASCII
015326     001                              .BYTE   1                       ;;TYPE 1 DIGIT(S)
015327     000                              .BYTE   0                       ;;SUPPRESS LEADING ZEROS

                                    ;       TYPE    ,$CRLF

015330  005737  001170                      TST     $TMP4                   ;SEE IF RK07 UNDER TEST
015334  001017                              BNE     7$                      ;BR IF YES
015336  012737  000632  015434              MOV     #632,LC                 ;ELSE LOAD RK06 PARAMETERS
015344  005037  015444                      CLR     E.DDT
015350  012737  001000  015436              MOV     #1000,MC1
015356  012737  000777  015440              MOV     #777,MASK
015364  012737  160017  015442              MOV     #160017,MASK1
015372  000425                              BR      TST5                    ;;GOTO NEXT TEST

015374  012737  001456  015434      7$:     MOV     #1456,LC                ;LOAD RK07 PARAMETERS
015402  012737  000400  015444              MOV     #D.DDT,E.DDT
015410  012737  002000  015436              MOV     #2000,MC1
015416  012737  001777  015440              MOV     #1777,MASK
015424  012737  140017  015442              MOV     #140017,MASK1
015432  000405                              BR      TST5                    ;;GOTO NEXT TEST

015434  000000                      LC:     0                               ;LAST CYL
015436  000000                      MC1:    0                               ;MAJ CYL + 1 SHIFT
015440  000000                      MASK:   0
015442  000000                      MASK1:  0
015444  000000                      E.DDT:  0                               ;EXPECTED DRIVE TYPE TO E.A0

015446                              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
                                    ;*
                                    ;;*************************************************************
015446  000004                      TST5:   SCOPE
015450  012737  000001  001174              MOV     #1,$TIMES               ;;DO 1 ITERATION
015456  012706  001100                      MOV     #STACK,SP               ;RESTORE STK PTR

015462  005737  001216                      TST     $PASS
015466  001042                              BNE     TST6                    ;;GO TO NEXT IF NOT FIRST PASS
015470  004737  034214                      JSR     PC,SUBCLR               ;DO SUBSYS CLEAR
015474  104024                              ERROR   24                      ;CERR AFTER SCLR

015476  104401  045300                      TYPE    ,MSG16                  ;DRIVE SERIAL NO.
015502  012765  000003  000026              MOV     #3,RKMR1(R5)            ;SELECT BYTE 3
015510  004737  033664                      JSR     PC,GSTAT                ;GET STATUS
015514  013701  007402                      MOV     HMR2,R1                 ;GET SERIAL #
015520  012704  042614                      MOV     #$OCTVL,R4              ;GET ADDR CHAR BUFF
015524  010446                              MOV     R4,-(SP)                ;STORE ON STACK FOR $SUPRS
015526  012703  000003                      MOV     #3,R3                   ;SETUP CHAR COOUNT
015532  006101                              ROL     R1                      ;INITIALIZE BIT POSITIONS
```

```
015534  006101                                  ROL     R1
015536  006101                          1$:     ROL     R1                      ;GET NEXT 4 BITS
015540  006101                                  ROL     R1
015542  006101                                  ROL     R1
015544  006101                                  ROL     R1
015546  010100                                  MOV     R1,R0                   ;GET WORKING COPY
015550  042700  177760                          BIC     #177760,R0              ;CLEAR ALL BUT LOW 4 BITS
015554  052700  000060                          BIS     #60,R0                  ;CONVERT TO ASCII DIGIT
015560  110024                                  MOVB    R0,(R4)+                ;PUT ASCII DIGIT INTO CHAR BUFF
015562  005303                                  DEC     R3
015564  001364                                  BNE     1$                      ;BR IF ALL 3 CHARS NOT DONE
015566  105014                                  CLRB    (R4)                    ;ELSE INSERT NULL TERMINATOR

015570  004737  043062                          JSR     PC,$SUPRS               ;TYPE
                                        ;       TYPE    ,$CRLF
                                        ;       TYPE    ,$CRLF

                                        ;;*****************************************************************
                                        ;*TEST 6        SET VV WITH PACK COMMAND
                                        ;*
                                        ;*      IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
                                        ;*
                                        ;;*****************************************************************
015574  000004                          TST6:   SCOPE
015576  012737  000001  001174                  MOV     #1,$TIMES               ;;DO 1 ITERATION
015604  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR

015610  004737  034214                          JSR     PC,SUBCLR
015614  104024                                  ERROR   24                      ;CERR AFTER SCLR

015616  032737  000100  007402                  BIT     #D.VV,HMR2
015624  001021                                  BNE     TST7                    ;;GO TO NEXT TEST IF VV SET

015626  104415                                  SCOP1
015630  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR

015634  004737  034214                          JSR     PC,SUBCLR
015640  104024                                  ERROR   24                      ;CERR AFTER SCLR

015642  012737  000003  007354                  MOV     #PACK,HCS1
015650  004737  032224                          JSR     PC,DOCMD                ;DO PACK CMD & GET CONTR RDY
015654  104116                                  ERROR   116                     ;RDY NOT SET AFTER PACK CMD

015656  032737  000100  007402                  BIT     #D.VV,HMR2
015664  001001                                  BNE     TST7                    ;;GO TO NEXT TEST IF VV NOW SET
015666  104027                                  ERROR   27                      ;PACK DID NOT SET V.V.



                                        ;;*****************************************************************
                                        ;*TEST 7        READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
                                        ;*
                                        ;*      THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
                                        ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
                                        ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
                                        ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
```

```
                                        ;*
                                        ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
                                        ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
                                        ;*
                                        ;*      SECTORS 1,3,5,7,9 CONTAIN IDENTICAL INFO FOR 20 SECTOR HARDWARE DETECTED BAD SEC
                                        ;*      SECTORS 11,13,15,17,19,21 CONTAIN IDENTICAL INFO FOR 20 SECTOR SOFTWARE DETECTED
                                        ;*
                                        ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
                                        ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
                                        ;*      A 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 WRITING
                                        ;*
                                        ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
                                        ;*
                                        ;;*************************************************************
015670  000004                          TST7:   SCOPE
015672  012737  000001  001174                  MOV     #1,$TIMES               ;;DO 1 ITERATION
015700  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR

015704  004737  034214                          JSR     PC,SUBCLR
015710  104024                                  ERROR   24                      ;CERR AFTER SCLR


015712  012765  100000  000000                  MOV     #CCLR,RKCS1(R5)
015720  013765  001222  000010                  MOV     $UNIT,RKCS2(R5)
015726  012737  000013  007354                  MOV     #RECAL,HCS1
015734  004737  032224                          JSR     PC,DOCMD                ;DO RECAL CMD & GET CONTR RDY
015740  104124                                  ERROR   124                     ;RDY NOT SET AFTER RECAL CMD

015742  012765  000001  000026                  MOV     #1,RKMR1(R5)            ;SELECT WORD 1
015750  004737  033664                          JSR     PC,GSTAT
015754  032737  020000  007402                  BIT     #D.RTZ,HMR2
015762  001001                                  BNE     64$
015764  104244                                  ERROR   244                     ;RTZ NOT SET DURING RECAL CMD
015766  013737  001406  007414  64$:            MOV     T10,TEMP2               ;SETUP TIMEOUT
015774  004737  032634                          JSR     PC,FATT1                ;FIND ATTN
016000  104055                                  ERROR   55                      ;NO ATTN AFTER RECAL CMD

016002  012765  100000  000000                  MOV     #CCLR,RKCS1(R5)
016010  013765  001222  000010                  MOV     $UNIT,RKCS2(R5)         ;DRIVE#
016016  012737  000005  007354                  MOV     #CLEAR,HCS1
016024  004737  032224                          JSR     PC,DOCMD                ;DO DRIVE CLEAR CMD & GET CONTR RDY
016030  104151                                  ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
016032  004737  032602                          JSR     PC,TSTATN               ;TEST FOR ATTN
016036  000401                                  BR      65$
016040  104154                                  ERROR   154                     ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
016042                                  65$:



016042  004737  034214                          JSR     PC,SUBCLR
016046  104024                                  ERROR   24                      ;CERR AFTER SCLR

016050  005037  007414                          CLR     TEMP2                   ;SECTOR CTR
016054  005037  007416                          CLR     TEMP3                   ;0=22 SECTOR HARDWARE DETECTED TABLE
```

```
                                                          ;1=22 SECTOR SOFTWARE DETECTED TABLE
                                                          ;2=20 SECTOR HARDWARE DETECTED TABLE
                                                          ;3=20 SECTOR SOFTWARE DETECTED TABLE
016060  012737  003336  007420        MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE ADDR.
016066  013765  007420  000004        MOV     TEMP4,RKBA(R5)
016074  012737  001000  007422        MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
016102  013765  007422  000006        MOV     TEMP5,RKDA(R5)

016110  013765  015434  000020  1$:   MOV     LC,RKDC(R5)     ;LAST CYL
016116  012765  177400  000002        MOV     #-256.,RKWC(R5) ;LOAD WORD CT
016124  012737  000021  007354        MOV     #RDDATA,HCS1
016132  004737  032262                JSR     PC,DATCMD       ;DO READ DATA CMD & GET CONTR RDY
016136  104226                        ERROR   226             ;NO RDY AFTER READ DATA CMD
016140  004737  033664                JSR     PC,GSTAT        ;GET FRESH DATA
016144  032737  100000  007354        BIT     #CERR,HCS1
016152  001504                        BEQ     8$
016154  104227                        ERROR   227             ;CERR AFTER READ DATA CMD

016156  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
016164  005037  007446                CLR     E.B0            ;EXPECTED MSG B0
016170  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
016176  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
016204  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
016210  012737  000002  007456        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
016216  012737  000003  007462        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

016224  004737  033042                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
016230  000000                        .WORD   0!0!0           ;& MSGS SPECIFIED HERE
016232  104054                        ERROR   54              ;MSG A0 ERROR AFTER READ DATA CMD
016234  104026                        ERROR   26              ;MSH B0 ERROR
016236  104056                        ERROR   56              ;MSG A1 ERROR
016240  104030                        ERROR   30              ;MSG B1 ERROR

016242  004737  034214                JSR     PC,SUBCLR
016246  104024                        ERROR   24              ;CERR AFTER SUBCLR

016250  005237  007414                INC     TEMP2
016254  023727  007414  000005        CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
016262  001023                        BNE     5$
016264  005737  007416                TST     TEMP3
016270  001002                        BNE     2$
016272  104233                        ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
016274  000430                        BR      3$

016276  023727  007416  000001  2$:   CMP     TEMP3,#1
016304  001002                        BNE     4$
016306  104230                        ERROR   230             ;CANT READ SECTORS 10,12...
016310  000422                        BR      3$

016312  023727  007416  000002  4$:   CMP     TEMP3,#2
016320  001002                        BNE     6$
016322  104234                        ERROR   234             ;CANT READ SECTORS 1,3,5 ...
016324  000414                        BR      3$

016326  104231                  6$:   ERROR   231             ;CANT READ SECTORS 11,13,15 ...
016330  000412                        BR      3$
```

```
016332 013765 007420 000004 5$:   MOV    TEMP4,RKBA(R5)   ;RESTORE TABLE ADDR
016340 062737 000002 007422       ADD    #2,TEMP5         ;READ 2 SECTORS FROM LAST
016346 013765 007422 000006       MOV    TEMP5,RKDA(R5)
016354 000655                     BR     1$

016356 005237 001512        3$:   INC    BSERR            ;SET BSE FLAG
016362 000550                     BR     TST10            ;;GO TO NEXT TEST
016364 005737 003344        8$:   TST    BSE22H+6         ;TEST CARTRIDGE TYPE
016370 001404                     BEQ    9$               ;BRANCH IF DATA CARTRIDGE
016372 104235                     ERROR  235              ;ALIGNMENT CARTRIDGE USED
016374 005237 001512              INC    BSERR            ;SET BSE ERROR FLAG
016400 000476                     BR     10$

016402 005237 007416        9$:   INC    TEMP3
016406 023727 007416 000001       CMP    TEMP3,#1
016414 001020                     BNE    11$
016416 005037 007414              CLR    TEMP2            ;SECTOR CTR
016422 012737 005336 007420       MOV    #BSE22S,TEMP4    ;STORE 22 SECTOR SOFTWARE BSE ADDR
016430 013765 007420 000004       MOV    TEMP4,RKBA(R5)
016436 012737 001012 007422       MOV    #1012,TEMP5      ;TRACK 2, SECTOR 12(8)
016444 013765 007422 000006       MOV    TEMP5,RKDA(R5)
016452 000137 016110              JMP    1$               ;REPEAT

016456 023727 007416 000002 11$:  CMP    TEMP3,#2
016464 001020                     BNE    12$
016466 005037 007414              CLR    TEMP2            ;SECTOR CTR
016472 012737 002336 007420       MOV    #BSE20H,TEMP4    ;STORE 20 SECTOR HARDWARE BSE ADDR.
016500 013765 007420 000004       MOV    TEMP4,RKBA(R5)
016506 012737 001001 007422       MOV    #1001,TEMP5      ;TRACK 2, SECTOR 1
016514 013765 007422 000006       MOV    TEMP5,RKDA(R5)
016522 000137 016110              JMP    1$               ;REPEAT

016526 023727 007416 000003 12$:  CMP    TEMP3,#3
016534 001020                     BNE    10$
016536 005037 007414              CLR    TEMP2            ;SECTOR CTR
016542 012737 004336 007420       MOV    #BSE20S,TEMP4    ;STORE 20 SECTOR SOFTWARE BSE ADDR
016550 013765 007420 000004       MOV    TEMP4,RKBA(R5)
016556 012737 001013 007422       MOV    #1013,TEMP5      ;TRACK 2, SECTOR 13(8)
016564 013765 007422 000006       MOV    TEMP5,RKDA(R5)
016572 000137 016110              JMP    1$               ;REPEAT

016576 005737 001216        10$:  TST    $PASS
016602 001040                     BNE    TST10            ;;GO TO NEXT TST IF NOT 1'ST PASS
016604 104401 045324              TYPE   ,MSG17           ;CART SERIAL #
016610 012746 003336              MOV    #BSE22H,-(SP)
016614 004737 042512              JSR    PC,$DB20         ;CONVERT DBL BINARY WORD TO OCTAL
016620 004737 043062              JSR    PC,$SUPRS        ;TYPE SERIAL #
016624 104401 001205              TYPE   ,$CRLF
016630 104401 001205              TYPE   ,$CRLF

016634 004737 034214              JSR    PC,SUBCLR
016640 104024                     ERROR  24               ;CERR AFTER SCLR
                                                          ;GO BACK TO CYL 0

016642 012737 000017 007354       MOV    #SEEK,HCS1
```

```
016650  004737  032224                   JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
016654  104131                           ERROR   131                 ;NO RDY AFTER SEEK CMD

016656  013737  001420  007412           MOV     T50000,TEMP1        ;SETUP TIMEOUT
016664  004737  032730                   JSR     PC,FATT2            ;FIND ATTN
016670  104132                           ERROR   132                 ;NO ATTN AFTER SEEK CMD

016672  032737  100000  007354           BIT     #CERR,HCS1
016700  001401                           BEQ     66$
016702  104210                           ERROR   210                 ;CERR AFTER SEEK CMD

016704                           66$:




                                 .SBTTL  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.
                                 ;*
                                 ;;*************************************************************
016704  000004                   TST10:  SCOPE
016706  012737  000001  001174           MOV     #1,$TIMES           ;;DO 1 ITERATION
016714  012706  001100                   MOV     #STACK,SP           ;RESTORE STK PTR

016720  005737  001512                   TST     BSERR               ;SEE IF ALIGN CART
016724  001406                           BEQ     2$                  ;BR IF NO
016726  104401  045662                   TYPE    ,MSG40              ;BSE OR ALIGN CART USED
016732  104401  045354                   TYPE    ,MSG26              ;ABORTING BAL OF TESTS
016736  000137  031452                   JMP     $EOP

016742  004737  034214           2$:     JSR     PC,SUBCLR
016746  104024                           ERROR   24                  ;CERR AFTER SCLR

016750  005237  007340                   INC     BADHDR              ;USED FOR VALID HALT

016754  012700  001522                   MOV     #HDTAB,R0           ;MAKE ALL CYL 0 HEADERS IDENTICAL

016760  005020                   1$:     CLR     (R0)+               ;HEADER WORD 0: CYL 0
016762  012720  140000                   MOV     #140000,(R0)+       ;HEADER WORD 1: SECTOR 0
016766  012720  140000                   MOV     #140000,(R0)+       ;HEADER WORD 2: XOR OF 1 & 2

016772  020027  001726                   CMP     R0,#HDTAB+132.      ;ALL HEADERS DONE? (22X6=132)
016776  001370                           BNE     1$                  ;BR IF NO

017000  012765  001522  000004           MOV     #HDTAB,RKBA(R5)     ;HEADER TABLE
017006  012765  177676  000002           MOV     #-66.,RKWC(R5)      ;WORD COUNT

017014  012737  000027  007354           MOV     #<WRHEAD>,HCS1
```

```
017022  004737  032262                  JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
017026  104200                          ERROR   200                ;NO RDY AFTER WRITE HEADER CMD
017030  004737  033664                  JSR     PC,GSTAT           ;GET FRESH STATUS
017034  032737  100000  007354          BIT     #CERR,HCS1
017042  001405                          BEQ     64$
017044  104201                          ERROR   201                ;CERR AFTER WRITE HEADER CMD
017046  104401  045354                  TYPE    ,MSG26             ;ABORTING BAL OF TESTS
017052  000137  031452                  JMP     $EOF
017056                          64$:


017056  104415                          SCOP1
017060  012706  001100                  MOV     #STACK,SP          ;RESTORE STK PTR

017064  004737  034214                  JSR     PC,SUBCLR
017070  104024                          ERROR   24                 ;CERR AFTER SCLR

017072  005037  001402                  CLR     SECTOR
017076  013765  001402  000006  3$:     MOV     SECTOR,RKDA(R5)    ;TRACK/SECTOR #
017104  012765  001500  000004          MOV     #DATA1,RKBA(R5)    ;DATA TO BE ALL 1'S
017112  012765  177777  000002          MOV     #-1,RKWC(R5)       ;WORD COUNT=1


017120  012737  000023  007354          MOV     #<WRDATA>,HCS1
017126  004737  032262                  JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
017132  104011                          ERROR   11                 ;NO RDY AFTER WRITE DATA CMD
017134  004737  033664                  JSR     PC,GSTAT           ;GET FRESH STATUS
017140  032737  100000  007354          BIT     #CERR,HCS1
017146  001465                          BEQ     68$                ;BR IF NO ERRORS

017150  032737  000200  007370          BIT     #BSE,HER           ;SEE IF BAD SECTOR FLAG
017156  001421                          BEQ     66$                ;BR IF NO
017160  004737  035700                  JSR     PC,TRUERR          ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
017164  000455                          BR      67$                ;RETURN HERE IF NO

017166  005237  001402                  INC     SECTOR             ;RETURN HERE IF YES
017172  023727  001402  000012          CMP     SECTOR,#10.        ;ARE 10 CONSEC. SECTORS BAD
017200  001003                          BNE     65$                ;BR IF NO
017202  104046                          ERROR   46                 ;ABORTING TEST DETECTED 10 BAD SECTORS
017204  000137  017406                  JMP     5$                 ;BYPASS TEST
017210  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5)    ;TRY ANOTHER SECTOR
017216  000137  017076                  JMP     3$
017222  104012                  66$:    ERROR   12                 ;CERR WITH WRITE DATA CMD

017224  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
017232  005037  007446                  CLR     E.B0               ;EXPECTED MSG B0
017236  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
017244  012737  000001  007452          MOV     #1,E.B1            ;MSG ID FOR EXPECTED MSG B1
017252  005037  007454                  CLR     E.A2               ;EXPECTED MSG A2
017256  012737  000002  007456          MOV     #2,E.B2            ;MSG ID FOR EXPECTED MSG B2
017264  012737  000003  007462          MOV     #3,E.B3            ;MSG ID FOR EXPECTED MSG B3

017272  004737  033042                  JSR     PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
017276  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
017300  104052                          ERROR   52                 ;MSG A0 ERROR AFTER WRITE DATA CMD
017302  104023                          ERROR   23                 ;MSH B0 ERROR
```

```
017304  104053                          ERROR   53                      ;MSG A1 ERROR
017306  104025                          ERROR   25                      ;MSG B1 ERROR
017310  104401  045354                  TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
017314  000137  031452                  JMP     $EOP
017320  104063                  67$:    ERROR   63                      ;BAD SECTOR NOT LISTED IN TABLE
017322                          68$:

017322  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
017330  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
017334  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
017342  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
017350  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
017354  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
017362  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

017370  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
017374  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
017376  104052                          ERROR   52                      ;MSG A0 ERROR AFTER WRITE DATA CMD
017400  104023                          ERROR   23                      ;MSH B0 ERROR
017402  104053                          ERROR   53                      ;MSG A1 ERROR
017404  104025                          ERROR   25                      ;MSG B1 ERROR
017406                          5$:
                                ;;*************************************************************
                                ;*TEST 11        BASIC WRITE DATA TEST; FULL SECTOR
                                ;*
                                ;*       THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
                                ;*       A FULL SECTOR.  ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
                                ;*       & CHECKED BY A RD DATA COMMAND.  A FURTHER CHECK IS PERFORMED
                                ;*       BY THE WRT CHK COMMAND.
                                ;*       THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.
                                ;*
                                ;;*************************************************************
017406  000004                  TST11:  SCOPE
017410  012737  000001  001174          MOV     #1,$TIMES               ;;DO 1 ITERATION
017416  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR

017422  004737  034214                  JSR     PC,SUBCLR
017426  104024                          ERROR   24                      ;CERR AFTER SCLR

017430  012765  001522  000004          MOV     #HDTAB,RKBA(R5) ;RESTORE TO 22 SECTOR
017436  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;STANDARD FORMAT
017444  005037  001346                  CLR     TOCYL

017450  013737  001346  001362          MOV     TOCYL,CALADD            ;SETUP
017456  012737  000000  001460          MOV     #0,HEAD                 ;TO FILL
017464  012737  000000  001466          MOV     #0,FORMAT               ;HEADER
017472  004737  035212                  JSR     PC,FHDTAB               ;TABLE

017476  012737  000027  007354          MOV     #<WRHEAD>,HCS1
017504  004737  032262                  JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
017510  104200                          ERROR   200                     ;NO RDY AFTER WRITE HEADER CMD
017512  004737  033664                  JSR     PC,GSTAT                ;GET FRESH STATUS
017516  032737  100000  007354          BIT     #CERR,HCS1
017524  001405                          BEQ     64$
017526  104201                          ERROR   201                     ;CERR AFTER WRITE HEADER CMD
```

```
017530  104401  045354                   TYPE    ,MSG26               ;ABORTING BAL OF TESTS
017534  000137  031452                   JMP     $EOP
017540                           64$:

017540  005037  007340                   CLR     BADHDR               ;USED FOR VALID HALT
017544  104415                           SCOP1
017546  012706  001100                   MOV     #STACK,SP            ;RESTORE STK PTR

017552  004737  034214                   JSR     PC,SUBCLR
017556  104024                           ERROR   24                   ;CERR AFTER SCLR

017560  005037  001402                   CLR     SECTOR
017564  013765  001402  000006   8$:     MOV     SECTOR,RKDA(R5)      ;SETUP SECTOR
017572  012765  001474  000004           MOV     #DATA0,RKBA(R5)      ;WRITE ALL 0'S
017600  013700  001474                   MOV     DATA0,R0
017604  052765  000020  000010   1$:     BIS     #BAI,RKCS2(R5)
017612  012765  177400  000002           MOV     #-256.,RKWC(R5)

017620  012737  000023  007354           MOV     #<WRDATA>,HCS1
017626  004737  032262                   JSR     PC,DATCMD            ;DO DATA X FOR CMD & GET CONTR RDY
017632  104011                           ERROR   11                   ;NO RDY AFTER WRITE DATA CMD
017634  004737  033664                   JSR     PC,GSTAT             ;GET FRESH STATUS
017640  032737  100000  007354           BIT     #CERR,HCS1
017646  001465                           BEQ     68$                  ;BR IF NO ERRORS

017650  032737  000200  007370           BIT     #BSE,HER             ;SEE IF BAD SECTOR FLAG
017656  001421                           BEQ     66$                  ;BR IF NO
017660  004737  035700                   JSR     PC,TRUERR            ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
017664  000455                           BR      67$                  ;RETURN HERE IF NO

017666  005237  001402                   INC     SECTOR               ;RETURN HERE IF YES
017672  023727  001402  000012           CMP     SECTOR,#10.          ;ARE 10 CONSEC. SECTORS BAD
017700  001003                           BNE     65$                  ;BR IF NO
017702  104046                           ERROR   46                   ;ABORTING TEST DETECTED 10 BAD SECTORS
017704  000137  021156                   JMP     7$                   ;BYPASS TEST
017710  012765  100000  000000   65$:    MOV     #CCLR,RKCS1(R5)      ;TRY ANOTHER SECTOR
017716  000137  017564                   JMP     8$
017722  104012                   66$:    ERROR   12                   ;CERR WITH WRITE DATA CMD

017724  012737  010340  007444           MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
017732  005037  007446                   CLR     E.B0                 ;EXPECTED MSG B0
017736  012737  001720  007450           MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
017744  012737  000001  007452           MOV     #1,E.B1              ;MSG ID FOR EXPECTED MSG B1
017752  005037  007454                   CLR     E.A2                 ;EXPECTED MSG A2
017756  012737  000002  007456           MOV     #2,E.B2              ;MSG ID FOR EXPECTED MSG B2
017764  012737  000003  007462           MOV     #3,E.B3              ;MSG ID FOR EXPECTED MSG B3

017772  004737  033042                   JSR     PC,CHKMSG            ;CHECK MSGS A0, B0, A1, B1
017776  000003                           .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
020000  104052                           ERROR   52                   ;MSG A0 ERROR AFTER WRITE DATA CMD
020002  104023                           ERROR   23                   ;MSH B0 ERROR
020004  104053                           ERROR   53                   ;MSG A1 ERROR
020006  104025                           ERROR   25                   ;MSG B1 ERROR
020010  104401  045354                   TYPE    ,MSG26               ;ABORTING BAL OF TESTS
020014  000137  031452                   JMP     $EOP
020020  104063                   67$:    ERROR   63                   ;BAD SECTOR NOT LISTED IN TABLE
```

```
020022                              68$:

020022  012737  010340  007444              MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
020030  005037  007446                      CLR     E.B0            ;EXPECTED MSG B0
020034  012737  001720  007450              MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
020042  012737  000001  007452              MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
020050  005037  007454                      CLR     E.A2            ;EXPECTED MSG A2
020054  012737  000002  007456              MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
020062  012737  000003  007462              MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

020070  004737  033042                      JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
020074  000003                              .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
020076  104052                              ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
020100  104023                              ERROR   23              ;MSH B0 ERROR
020102  104053                              ERROR   53              ;MSG A1 ERROR
020104  104025                              ERROR   25              ;MSG B1 ERROR
020106  104415                              SCOP1
020110  012706  001100                      MOV     #STACK,SP       ;RESTORE STK PTR

020114  004737  034214                      JSR     PC,SUBCLR
020120  104024                              ERROR   24              ;CERR AFTER SCLR

020122  013765  001402  000006              MOV     SECTOR,RKDA(R5) ;SETUP SECTOR
020130  012765  006336  000004              MOV     #RDTAB,RKBA(R5)
020136  012765  177400  000002              MOV     #-256.,RKWC(R5)


020144  012737  000021  007354              MOV     #<RDDATA>,HCS1
020152  004737  032262                      JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
020156  104013                              ERROR   13              ;NO RDY AFTER READ DATA CMD
020160  004737  033664                      JSR     PC,GSTAT        ;GET FRESH STATUS
020164  032737  100000  007354              BIT     #CERR,HCS1
020172  001454                              BEQ     72$
020174  032737  000200  007370              BIT     #BSE,HER        ;SEE IF BAD SECTOR
020202  001406                              BEQ     70$
020204  104065                              ERROR   65              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
020206  000413                              BR      73$
020210  104401  045354              69$:    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
020214  000137  031452                      JMP     $EOP

020220  032737  100000  007370      70$:    BIT     #DCK,HER        ;SEE IF DATA CHECK ERROR
020226  001402                              BEQ     71$
020230  104021                              ERROR   21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
020232  000401                              BR      73$

020234  104014                      71$:    ERROR   14              ;CERR AFTER READ DATA CMD.

020236                              73$:

020236  012737  010340  007444              MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
020244  005037  007446                      CLR     E.B0            ;EXPECTED MSG B0
020250  012737  001720  007450              MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
020256  012737  000001  007452              MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
020264  005037  007454                      CLR     E.A2            ;EXPECTED MSG A2
020270  012737  000002  007456              MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
020276  012737  000003  007462              MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
```

```
        020304  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
        020310  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
        020312  104054                          ERROR   54                      ;MSG A0 ERROR AFTER READ DATA CMD
        020314  104026                          ERROR   26                      ;MSH B0 ERROR
        020316  104056                          ERROR   56                      ;MSG A1 ERROR
        020320  104030                          ERROR   30                      ;MSG B1 ERROR
        020322  000732                          BR      69$
        020324                          72$:

        020324  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
        020332  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
        020336  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
        020344  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
        020352  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
        020356  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
        020364  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

        020372  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
        020376  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
        020400  104054                          ERROR   54                      ;MSG A0 ERROR AFTER READ DATA CMD
        020402  104026                          ERROR   26                      ;MSH B0 ERROR
        020404  104056                          ERROR   56                      ;MSG A1 ERROR
        020406  104030                          ERROR   30                      ;MSG B1 ERROR

        020410  012701  006336                  MOV     #RDTAB,R1
        020414  011137  001452          2$:     MOV     (R1),WD1                ;ACTUAL WORD FOR TYPEOUT
        020420  010037  001454                  MOV     R0,WD2                  ;EXPECTED DATA FOR TYPEOUT

        020424  020021                          CMP     R0,(R1)+                ;COMPARE READ DATA TABLE AGAINST
        020426  001401                          BEQ     3$                      ;THE 'SHOULD BE' VALUE
        020430  104020                          ERROR   20                      ;READ DATA DID NOT COMPARE WITH WRITE DATA

        020432  020127  007336          3$:     CMP     R1,#RDTAB+512.          ;SEE IF REACHED END OF TABLE
        020436  001366                          BNE     2$                      ;BR IF NO & DO NEXT WORD

        020440  020037  001474                  CMP     R0,DATA0                ;SEE IF DID ALL 0'S
        020444  001401                          BEQ     4$                      ;BR IF YES
        020446  000412                          BR      5$

        020450  012765  001500  000004  4$:     MOV     #DATA1,RKBA(R5)         ;WRITE ALL 1'S
        020456  013700  001500                  MOV     DATA1,R0
        020462  013765  001402  000006          MOV     SECTOR,RKDA(R5)
        020470  000137  017604                  JMP     1$

        020474                          5$:
        020474  104415                          SCOP1
        020476  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR

        020502  004737  034214                  JSR     PC,SUBCLR
        020506  104024                          ERROR   24                      ;CERR AFTER SCLR

        020510  052765  000020  000010          BIS     #BAI,RKCS2(R5)          ;THIS PORTION OF THE TEST CHECKS
        020516  012765  001500  000004          MOV     #DATA1,RKBA(R5)         ;OUT THE WRITE CHECK CMD
        020524  012765  177400  000002          MOV     #-256.,RKWC(R5)         ;ALL 1'S WERE PREVIOUSLY WRITTEN
        020532  013765  001402  000006          MOV     SECTOR,RKDA(R5)
```

```

020540 012737 000031 007354          MOV    #<WRTCHK>,HCS1
020546 004737 032262                 JSR    PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
020552 104015                        ERROR  15                      ;NO RDY AFTER WRITE CHECK CMD
020554 004737 033664                 JSR    PC,GSTAT                ;GET FRESH STATUS
020560 032737 100000 007354          BIT    #CERR,HCS1
020566 001453                        BEQ    75$
020570 032737 040000 007356          BIT    #WCE,HCS2               ;SEE IF WRITE CHECK ERROR
020576 001410                        BEQ    74$
020600 016537 000024 001452          MOV    RKDB(R5),WD1            ;ACTUAL WORD FOR PRINTOUT
020606 013737 001500 001454          MOV    DATA1,WD2                       ;EXPECTED WORD FOR TYPEOUT
020614 104016                        ERROR  16                      ;WCE AFTER WRITE CMD
020616 000437                        BR     75$

020620 104022                 74$:   ERROR  22                      ;CERR AFTER WRITE CHECK CMD

020622 012737 010340 007444          MOV    #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
020630 005037 007446                 CLR    E.B0                    ;EXPECTED MSG B0
020634 012737 001720 007450          MOV    #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
020642 012737 000001 007452          MOV    #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
020650 005037 007454                 CLR    E.A2                    ;EXPECTED MSG A2
020654 012737 000002 007456          MOV    #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
020662 012737 000003 007462          MOV    #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

020670 004737 033042                 JSR    PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
020674 000003                        .WORD  T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
020676 104057                        ERROR  57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
020700 104031                        ERROR  31                      ;MSH B0 ERROR
020702 104060                        ERROR  60                      ;MSG A1 ERROR
020704 104032                        ERROR  32                      ;MSG B1 ERROR
020706 104401 045354                 TYPE   ,MSG26                  ;ABORTING BAL OF TESTS
020712 000137 031452                 JMP    $EOP

020716                        75$:

020716 012737 010340 007444          MOV    #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
020724 005037 007446                 CLR    E.B0                    ;EXPECTED MSG B0
020730 012737 001720 007450          MOV    #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
020736 012737 000001 007452          MOV    #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
020744 005037 007454                 CLR    E.A2                    ;EXPECTED MSG A2
020750 012737 000002 007456          MOV    #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
020756 012737 000003 007462          MOV    #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

020764 004737 033042                 JSR    PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
020770 000003                        .WORD  T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
020772 104057                        ERROR  57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
020774 104031                        ERROR  31                      ;MSH B0 ERROR
020776 104060                        ERROR  60                      ;MSG A1 ERROR
021000 104032                        ERROR  32                      ;MSG B1 ERROR

021002 104415                        SCOP1
021004 012706 001100                 MOV    #STACK,SP               ;RESTORE STK PTR

021010 004737 034214                 JSR    PC,SUBCLR
021014 104024                        ERROR  24                      ;CERR AFTER SCLR

```

```
021016  012765  001474  000004          MOV     #DATA0,RKBA(R5)  ;SETUP TO CHECK AGAINST WRONG DATA
021024  012765  177400  000002          MOV     #-256.,RKWC(R5)
021032  013765  001402  000006          MOV     SECTOR,RKDA(R5)
021040  012737  000031  007354          MOV     #WRTCHK,HCS1
021046  004737  032262                  JSR     PC,DATCMD        ;DO WRITE CHECK CMD. & GET CONTR RDY.
021052  104015                          ERROR   15               ;NO RDY AFTER WRITE CHECK CMD
021054  004737  033664                  JSR     PC,GSTAT         ;GET FRESH STATUS
021060  032737  040000  007356          BIT     #WCE,HCS2        ;EXPECT MISCOMPARE
021066  001001                          BNE     6$
021070  104017                          ERROR   17               ;WRITE CHECK CMD NOT FUNCTIONING
                                                                 ;WITH INTENTIONAL MISCOMPARE
021072                          6$:

021072  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
021100  005037  007446                  CLR     E.B0             ;EXPECTED MSG B0
021104  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
021112  012737  000001  007452          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
021120  005037  007454                  CLR     E.A2             ;EXPECTED MSG A2
021124  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
021132  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3

021140  004737  033042                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
021144  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
021146  104057                          ERROR   57               ;MSG A0 ERROR AFT WRT CHK CMD
021150  104031                          ERROR   31               ;MSH B0 ERROR
021152  104060                          ERROR   60               ;MSG A1 ERROR
021154  104032                          ERROR   32               ;MSG B1 ERROR
021156                          7$:

                                ;;*****************************************************
                                ;*TEST 12        20 SECTOR FORMAT TEST
                                ;*
                                ;*       ALL 1'S ARE WRITTEN ON A FULL SECTOR IN 20 SECTOR FORMAT.
                                ;*       MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
                                ;*       CYL 0, TRACK 0, & SECTOR 0 IS USED.
                                ;*
                                ;;*****************************************************
021156  000004                  TST12:  SCOPE
021160  012737  000001  001174          MOV     #1,$TIMES        ;;DO 1 ITERATION
021166  012706  001100                  MOV     #STACK,SP        ;RESTORE STK PTR

021172  004737  034214                  JSR     PC,SUBCLR
021176  104024                          ERROR   24               ;CERR AFTER SCLR

021200  012765  001522  000004          MOV     #HDTAB,RKBA(R5)  ;HEADER WORD TABLE
021206  012765  177704  000002          MOV     #-60.,RKWC(R5)   ;WORD COUNT FOR 20 SECTOR FMT
021214  005037  001346                  CLR     TOCYL
021220  005237  007340                  INC     BADHDR           ;USED FOR VALID HALT

021224  013737  001346  001362          MOV     TOCYL,CALADD     ;SETUP
021232  012737  000000  001460          MOV     #0,HEAD          ;TO FILL
021240  012737  000001  001466          MOV     #1,FORMAT        ;HEADER
021246  004737  035212                  JSR     PC,FHDTAB        ;TABLE
```

```
021252  012737  010027  007354          MOV     #<CFMT!WRHEAD>,HCS1
021260  004737  032262                  JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
021264  104200                          ERROR   200                ;NO RDY AFTER WRITE HEADER CMD
021266  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
021274  004737  032224                  JSR     PC,DOCMD
021300  104117                          ERROR   117                ;NO RDY AFTER SELDRV CMD
021302  032737  100000  007354          BIT     #CERR,HCS1
021310  001405                          BEQ     64$
021312  104201                          ERROR   201                ;CERR AFTER WRITE HEADER CMD
021314  104401  045354                  TYPE    ,MSG26             ;ABORTING BAL OF TESTS
021320  000137  031452                  JMP     $EOP
021324                          64$:
021324  104415                          SCOP1
021326  012706  001100                  MOV     #STACK,SP          ;RESTORE STK PTR
021332  004737  034214                  JSR     PC,SUBCLR
021336  104024                          ERROR   24                 ;CERR AFTER SCLR
021340  005037  001402                  CLR     SECTOR
021344  013765  001402  000006  4$:     MOV     SECTOR,RKDA(R5)
021352  012765  001500  000004          MOV     #DATA1,RKBA(R5)    ;WRITE ALL 1'S
021360  052765  000020  000010          BIS     #BAI,RKCS2(R5)     ;BUSS ADDR INCR INHIBIT
021366  012765  177400  000002          MOV     #-256.,RKWC(R5)    ;DO FULL SECTOR
021374  012737  010023  007354          MOV     #<CFMT!WRDATA>,HCS1
021402  004737  032262                  JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
021406  104011                          ERROR   11                 ;NO RDY AFTER WRITE DATA CMD
021410  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
021416  004737  032224                  JSR     PC,DOCMD
021422  104117                          ERROR   117                ;NO RDY AFTER SELDRV CMD
021424  032737  100000  007354          BIT     #CERR,HCS1
021432  001465                          BEQ     68$                ;BR IF NO ERRORS
021434  032737  000200  007370          BIT     #BSE,HER           ;SEE IF BAD SECTOR FLAG
021442  001421                          BEQ     66$                ;BR IF NO
021444  004737  035700                  JSR     PC,TRUERR          ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
021450  000455                          BR      67$                ;RETURN HERE IF NO
021452  005237  001402                  INC     SECTOR             ;RETURN HERE IF YES
021456  023727  001402  000012          CMP     SECTOR,#10.        ;ARE 10 CONSEC. SECTORS BAD
021464  001003                          BNE     65$                ;BR IF NO
021466  104046                          ERROR   46                 ;ABORTING TEST DETECTED 10 BAD SECTORS
021470  000137  022374                  JMP     3$                 ;BYPASS TEST
021474  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5)    ;TRY ANOTHER SECTOR
021502  000137  021344                  JMP     4$
021506  104012                  66$:    ERROR   12                 ;CERR WITH WRITE DATA CMD
021510  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
021516  005037  007446                  CLR     E.B0               ;EXPECTED MSG B0
021522  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
021530  012737  000001  007452          MOV     #1,E.B1            ;MSG ID FOR EXPECTED MSG B1
021536  005037  007454                  CLR     E.A2               ;EXPECTED MSG A2
021542  012737  000002  007456          MOV     #2,E.B2            ;MSG ID FOR EXPECTED MSG B2
021550  012737  000003  007462          MOV     #3,E.B3            ;MSG ID FOR EXPECTED MSG B3
```

```

021556  004737  033042                  JSR     PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
021562  000003                          .WORD   T.A2!T.B2!0                ;& MSGS SPECIFIED HERE
021564  104052                          ERROR   52                 ;MSG A0 ERROR AFTER WRITE DATA CMD
021566  104023                          ERROR   23                 ;MSH B0 ERROR
021570  104053                          ERROR   53                 ;MSG A1 ERROR
021572  104025                          ERROR   25                 ;MSG B1 ERROR
021574  104401  045354                  TYPE    ,MSG26             ;ABORTING BAL OF TESTS
021600  000137  031452                  JMP     $EOP
021604  104063                  67$:    ERROR   63                 ;BAD SECTOR NOT LISTED IN TABLE
021606                          68$:
021606  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
021614  004737  032224                  JSR     PC,DOCMD
021620  104117                          ERROR   117                ;RDY NOT FOUND AFTER SELDRV CMD
021622  032737  001000  007402          BIT     #D.FORM,HMR2
021630  001001                          BNE     1$
021632  104102                          ERROR   102                ;FORMAT NOT SET

021634                          1$:

021634  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
021642  005037  007446                  CLR     E.B0               ;EXPECTED MSG B0
021646  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
021654  012737  000001  007452          MOV     #1,E.B1            ;MSG ID FOR EXPECTED MSG B1
021662  005037  007454                  CLR     E.A2               ;EXPECTED MSG A2
021666  012737  000002  007456          MOV     #2,E.B2            ;MSG ID FOR EXPECTED MSG B2
021674  012737  000003  007462          MOV     #3,E.B3            ;MSG ID FOR EXPECTED MSG B3

021702  004737  033042                  JSR     PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
021706  000000                          .WORD   0!0!0              ;& MSGS SPECIFIED HERE
021710  104052                          ERROR   52                 ;MSG A0 ERROR AFTER WRITE DATA CMD
021712  104023                          ERROR   23                 ;MSH B0 ERROR
021714  104053                          ERROR   53                 ;MSG A1 ERROR
021716  104025                          ERROR   25                 ;MSG B1 ERROR
021720  012765  177400  000002          MOV     #-256.,RKWC(R5)
021726  012765  001500  000004          MOV     #DATA1,RKBA(R5)
021734  052765  000020  000010          BIS     #BAI,RKCS2(R5)
021742  013765  001402  000006          MOV     SECTOR,RKDA(R5)

021750  012737  010031  007354          MOV     #<CFMT!WRTCHK>,HCS1
021756  004737  032262                  JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
021762  104015                          ERROR   15                 ;NO RDY AFTER WRITE CHECK CMD
021764  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
021772  004737  032224                  JSR     PC,DOCMD
021776  104117                          ERROR   117                ;NO RDY AFTER SELDRV CMD
022000  032737  100000  007354          BIT     #CERR,HCS1
022006  001453                          BEQ     70$
022010  032737  040000  007356          BIT     #WCE,HCS2          ;SEE IF WRITE CHECK ERROR
022016  001410                          BEQ     69$
022020  016537  000024  001452          MOV     RKDB(R5),WD1       ;ACTUAL WORD FOR PRINTOUT
022026  013737  001500  001454          MOV     DATA1,WD2                  ;EXPECTED WORD FOR TYPEOUT
022034  104016                          ERROR   16                 ;WCE AFTER WRITE CMD
022036  000437                          BR      70$

022040  104022                  69$:    ERROR   22                 ;CERR AFTER WRITE CHECK CMD

```

```
022042  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
022050  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
022054  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
022062  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
022070  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
022074  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
022102  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

022110  004737  033042                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
022114  000003                          .WORD   T.A2!T.B2!0     ;& MSGS SPECIFIED HERE
022116  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
022120  104031                          ERROR   31              ;MSH B0 ERROR
022122  104060                          ERROR   60              ;MSG A1 ERROR
022124  104032                          ERROR   32              ;MSG B1 ERROR
022126  104401  045354                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
022132  000137  031452                  JMP     $EOP

022136                          70$:
022136  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
022144  004737  032224                  JSR     PC,DOCMD
022150  104117                          ERROR   117             ;NO RDY AFTER SELDRV CMD
022152  032737  001000  007402          BIT     #D.FORM,HMR2
022160  001001                          BNE     2$
022162  104103                          ERROR   103             ;FORMAT NOT SET

022164                          2$:

022164  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
022172  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
022176  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
022204  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
022212  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
022216  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
022224  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

022232  004737  033042                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
022236  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
022240  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
022242  104031                          ERROR   31              ;MSH B0 ERROR
022244  104060                          ERROR   60              ;MSG A1 ERROR
022246  104032                          ERROR   32              ;MSG B1 ERROR
022250  104415                          SCOP1
022252  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

022256  004737  034214                  JSR     PC,SUBCLR
022262  104024                          ERROR   24              ;CERR AFTER SCLR

022264  012765  001522  000004          MOV     #HDTAB,RKBA(R5) ;RESTORE CYL 0 TO 22 SECTOR FMT
022272  012765  177676  000002          MOV     #-66.,RKWC(R5)
022300  005037  001346                  CLR     TOCYL


022304  013737  001346  001362          MOV     TOCYL,CALADD    ;SETUP
022312  012737  000000  001460          MOV     #0,HEAD         ;TO FILL
022320  012737  000000  001466          MOV     #0,FORMAT       ;HEADER
022326  004737  035212                  JSR     PC,FHDTAB       ;TABLE
```

```


022332  012737  000027  007354          MOV     #<WRHEAD>,HCS1
022340  004737  032262                  JSR     PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
022344  104200                          ERROR   200                ;NO RDY AFTER WRITE HEADER CMD
022346  004737  033664                  JSR     PC,GSTAT           ;GET FRESH STATUS
022352  032737  100000  007354          BIT     #CERR,HCS1
022360  001405                          BEQ     71$
022362  104201                          ERROR   201                ;CERR AFTER WRITE HEADER CMD
022364  104401  045354                  TYPE    ,MSG26             ;ABORTING BAL OF TESTS
022370  000137  031452                  JMP     $EOP
022374                          71$:

022374  005037  007340          3$:     CLR     BADHDR             ;USED FOR VALID HALT

                                ;;*************************************************************
                                ;*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,D0) 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
                                ;*
                                ;*************************************************************
022400  000004                  TST13:  SCOPE
022402  012737  000001  001174          MOV     #1,$TIMES          ;;DO 1 ITERATION
022410  012706  001100                  MOV     #STACK,SP          ;RESTORE STK PTR

022414  012702  000001                  MOV     #1,R2              ;MIN POS OFFSET

022420  004737  034214          1$:     JSR     PC,SUBCLR
022424  104024                          ERROR   24                 ;CERR AFTER SCLR

022426  010265  000016                  MOV     R2,RKASOF(R5)      ;SET OFFSET

022432  012737  022520  001176          MOV     #64$,$ESCAPE
022440  012737  000015  007354          MOV     #OFFSET,HCS1
022446  004737  032224                  JSR     PC,DOCMD           ;DO RECAL CMD & GET CONTR RDY
022452  104033                          ERROR   33                 ;NO RDY AFTER OFFSET CMD

022454  012737  032140  007444          MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
022462  005037  007446                  CLR     E.B0
022466  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
022474  012737  000001  007452          MOV     #1,E.B1

022502  004737  033042                  JSR     PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
022506  000000                          .WORD   0!0!0              ;& MSGS SPECIFIED HERE
022510  104035                          ERROR   35                 ;MSG A0 ERROR DURING OFFSET CMD
022512  104061                          ERROR   61                 ;MSH B0 ERROR
022514  104036                          ERROR   36                 ;MSG A1 ERROR
022516  104062                          ERROR   62                 ;MSG B1 ERROR

```

```
022520  005037  001176           64$:    CLR     $ESCAPE
022524  013737  001416  007412           MOV     T5000,TEMP1          ;SETUP TIMEOUT
022532  004737  032730                   JSR     PC,FATT2             ;FIND ATTN
022536  104034                           ERROR   34                   ;NO ATTN AFTER OFFSET CMD


022540  012737  052340  007444           MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
022546  005037  007446                   CLR     E.B0                 ;EXPECTED MSG B0
022552  012737  001720  007450           MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
022560  012737  000001  007452           MOV     #1,E.B1              ;MSG ID FOR EXPECTED MSG B1
022566  005037  007454                   CLR     E.A2                 ;EXPECTED MSG A2
022572  012737  000002  007456           MOV     #2,E.B2              ;MSG ID FOR EXPECTED MSG B2
022600  012737  000003  007462           MOV     #3,E.B3              ;MSG ID FOR EXPECTED MSG B3

022606  004737  033042                   JSR     PC,CHKMSG            ;CHECK MSGS A0, B0, A1, B1
022612  000003                           .WORD   T.A2!T.B2!0               ;& MSGS SPECIFIED HERE
022614  104260                           ERROR   260                  ;MSG A0 ERROR AFTER OFFSET CMD
022616  104261                           ERROR   261                  ;MSH B0 ERROR
022620  104037                           ERROR   37                   ;MSG A1 ERROR
022622  104040                           ERROR   40                   ;MSG B1 ERROR

022624  005737  001360                   TST     CYLADD
022630  001401                           BEQ     17$
022632  104042                           ERROR   42                   ;CYL ADDR IN B2 WAS NOT 0
                                                                      ;AFTER OFFSET CMD FROM CYL 0
022634  042737  001000  001356   17$:    BIC     #1000,CYLDIF         ;GET RID OF HIGH BIT
022642  010265  000016                   MOV     R2,RKASOF(R5)        ;REFRESH RKASOF

022646  032702  000200                   BIT     #BIT7,R2
022652  001005                           BNE     65$                  ;BR IF NEG OFFSET

022654  020237  001356                   CMP     R2,CYLDIF            ;CHECK POS OFFSET
022660  001406                           BEQ     66$
022662  104114                           ERROR   114                  ;OFFSET IN A2 NOT = RKASOF
022664  000404                           BR      66$                  ;AFTER OFFSET CMD

022666  020137  001356           65$:    CMP     R1,CYLDIF            ;CHECK NEG OFFSET
022672  001401                           BEQ     66$
022674  104114                           ERROR   114                  ;OFFSET IN A2 NOT = RKASOF
                                                                      ;AFTER OFFSET CMD
022676                           66$:

022676  012765  100000  000000           MOV     #CCLR,RKCS1(R5)
022704  013765  001222  000010           MOV     $UNIT,RKCS2(R5)      ;DRIVE#
022712  012737  000005  007354           MOV     #CLEAR,HCS1
022720  004737  032224                   JSR     PC,DOCMD             ;DO DRIVE CLEAR CMD & GET CONTR RDY
022724  104151                           ERROR   151                  ;NO RDY AFTER DRIVE CLEAR CMD
022726  004737  032602                   JSR     PC,TSTATN            ;TEST FOR ATTN
022732  000401                           BR      67$
022734  104154                           ERROR   154                  ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
022736                           67$:

022736  012765  100000  000000           MOV     #CCLR,RKCS1(R5)
022744  004737  033664                   JSR     PC,GSTAT
022750  032737  002000  007402           BIT     #D.OFF,HMR2
022756  001001                           BNE     4$
```

```
022760  104043                                  ERROR   43                      ;OFFSET BIT IN RKMR2 CLEARED
                                                                                ;AFTER DRIVE CLEAR CMD & SELECT DRV CMD
022762  012737  012340  007444  4$:     MOV     #<D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
022770  005037  007446                  CLR     E.B0
022774  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
023002  012737  000001  007452          MOV     #1,E.B1

023010  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
023014  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
023016  104273                          ERROR   273                     ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
023020  104265                          ERROR   265                     ;MSH B0 ERROR
023022  104274                          ERROR   274                     ;MSG A1 ERROR
023024  104266                          ERROR   266                     ;MSG B1 ERROR
023026  042737  001000  001356          BIC     #1000,CYLDIF            ;GET RID OF HIGH BIT
023034  010265  000016                  MOV     R2,RKASOF(R5)           ;REFRESH RKASOF

023040  032702  000200                  BIT     #BIT7,R2
023044  001005                          BNE     68$                     ;BR IF NEG OFFSET

023046  020237  001356                  CMP     R2,CYLDIF               ;CHECK POS OFFSET
023052  001406                          BEQ     69$
023054  104115                          ERROR   115                     ;OFFSET IN A2 NOT = RKASOF
023056  000404                          BR      69$                     ;AFTER DRIVE CLEAR CMD

023060  020137  001356          68$:    CMP     R1,CYLDIF               ;CHECK NEG OFFSET
023064  001401                          BEQ     69$
023066  104115                          ERROR   115                     ;OFFSET IN A2 NOT = RKASOF
                                                                        ;AFTER DRIVE CLEAR CMD
023070                          69$:

023070  012737  000017  007354          MOV     #SEEK,HCS1
023076  004737  032224                  JSR     PC,DOCMD                ;DO SEEK CMD & GET CONTR READY
023102  104131                          ERROR   131                     ;NO RDY AFTER SEEK CMD

023104  013737  001420  007412          MOV     T50000,TEMP1            ;SETUP TIMEOUT
023112  004737  032730                  JSR     PC,FATT2                ;FIND ATTN
023116  104132                          ERROR   132                     ;NO ATTN AFTER SEEK CMD

023120  032737  100000  007354          BIT     #CERR,HCS1
023126  001401                          BEQ     70$
023130  104210                          ERROR   210                     ;CERR AFTER SEEK CMD

023132                          70$:


023132  032737  002000  007402          BIT     #D.OFF,HMR2
023140  001001                          BNE     7$
023142  104045                          ERROR   45                      ;OFFSET BIT CLEARED IN RKMR2 AFTER SEEK TO SELF.

023144                          7$:

023144  012737  052340  007444          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
023152  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
023156  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
023164  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
023172  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
```

```
023176  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
023204  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

023212  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
023216  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
023220  104140                          ERROR   140                     ;MSG A0 ERROR AFTER SEEK TO SELF
023222  104141                          ERROR   141                     ;MSH B0 ERROR
023224  104142                          ERROR   142                     ;MSG A1 ERROR
023226  104143                          ERROR   143                     ;MSG B1 ERROR
023230  042737  001000  001356          BIC     #1000,CYLDIF            ;GET RID OF HIGH BIT
023236  010265  000016                  MOV     R2,RKASOF(R5)           ;REFRESH RKASOF

023242  032702  000200                  BIT     #BIT7,R2
023246  001005                          BNE     71$                     ;BR IF NEG OFFSET

023250  020237  001356                  CMP     R2,CYLDIF               ;CHECK POS OFFSET
023254  001406                          BEQ     72$
023256  104123                          ERROR   123                     ;OFFSET IN A2 NOT = RKASOF
023260  000404                          BR      72$                     ;AFTER SEEK TO SELF

023262  020137  001356          71$:    CMP     R1,CYLDIF               ;CHECK NEG OFFSET
023266  001401                          BEQ     72$
023270  104123                          ERROR   123                     ;OFFSET IN A2 NOT = RKASOF
                                                                        ;AFTER SEEK TO SELF
023272                          72$:

023272  004737  034214                  JSR     PC,SUBCLR
023276  104024                          ERROR   24                      ;CERR AFTER SCLR

023300  012737  000012  001346          MOV     #10.,TOCYL
023306  012765  000012  000020          MOV     #10.,RKDC(R5)           ;SETUP CYL 10
                                                                        ;DO ACTUAL IMPLIED SEEK TO CYL 10 TO VERIFY
                                                                        ;OFFSET BIT IN RKMR2 CLEARED


023314  012700  001726                  MOV     #RHTAB,R0
023320  012737  000025  007354          MOV     #<RDHEAD>,HCS1
023326  004737  032262                  JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
023332  104171                          ERROR   171                     ;NO RDY AFTER READ HEADER CMD
023334  032737  100000  007354          BIT     #CERR,HCS1
023342  001405                          BEQ     74$
023344  104174                          ERROR   174                     ;CERR AFTER READ HEADER CMD
023346  104401  045354                  TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
023352  000137  031452                  JMP     $EOP

023356  016520  000024          74$:    MOV     RKDB(R5),(R0)+          ;1'ST WORD FROM SILO TO RHTAB
023362  016520  000024                  MOV     RKDB(R5),(R0)+          ;2'ND WORD
023366  016520  000024                  MOV     RKDB(R5),(R0)+          ;3'RD WORD


023372  032765  100000  000010          BIT     #DLT,RKCS2(R5)
023400  001407                          BEQ     75$
023402  004737  033664                  JSR     PC,GSTAT
023406  104173                          ERROR   173                     ;DLT AFTER READ HEADER CMD
023410  104401  045354                  TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
023414  000137  031452                  JMP     $EOP
```

```
023420                                  75$:

023420  012737  010340  007444                  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
023426  005037  007446                          CLR     E.B0                ;EXPECTED MSG B0
023432  012737  001720  007450                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
023440  012737  000001  007452                  MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
023446  005037  007454                          CLR     E.A2                ;EXPECTED MSG A2
023452  012737  000002  007456                  MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
023460  012737  000003  007462                  MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3

023466  004737  033042                          JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
023472  000003                                  .WORD   T.A2!T.B2!0         ;& MSGS SPECIFIED HERE
023474  104301                                  ERROR   301                 ;MSG A0 ERROR AFTER READ HEADER CMD
023476  104271                                  ERROR   271                 ;MSH B0 ERROR
023500  104302                                  ERROR   302                 ;MSG A1 ERROR
023502  104272                                  ERROR   272                 ;MSG B1 ERROR

023504  023737  001726  001346                  CMP     RHTAB,TOCYL         ;CHECK WORD 0 ONLY, CYL#
023512  001401                                  BEQ     73$                 ;BR IF SAME
023514  104051                                  ERROR   51                  ;WRONG CYL# ON HEADER
023516                                  73$:


023516  032737  002000  007402                  BIT     #D.OFF,HMR2
023524  001401                                  BEQ     9$
023526  104101                                  ERROR   101                 ;OFFSET NOT CLEARED AFTER READ HEADER WITH MOVEMENT

023530  023727  001360  000012          9$:     CMP     CYLADD,#10.
023536  001401                                  BEQ     10$
023540  104122                                  ERROR   122                 ;DID NOT GO TO CYL 10

023542  005737  001356                  10$:    TST     CYLDIF
023546  001401                                  BEQ     16$
023550  104101                                  ERROR   101                 ;OFFSET NOT CLEARED IN RKMR2

023552  004737  034214                  16$:    JSR     PC,SUBCLR
023556  104024                                  ERROR   24                  ;CERR AFTER SCLR

023560  012737  000017  007354                  MOV     #SEEK,HCS1
023566  004737  032224                          JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
023572  104131                                  ERROR   131                 ;NO RDY AFTER SEEK CMD

023574  013737  001420  007412                  MOV     T50000,TEMP1        ;SETUP TIMEOUT
023602  004737  032730                          JSR     PC,FATT2            ;FIND ATTN
023606  104132                                  ERROR   132                 ;NO ATTN AFTER SEEK CMD

023610  032737  100000  007354                  BIT     #CERR,HCS1
023616  001401                                  BEQ     76$
023620  104210                                  ERROR   210                 ;CERR AFTER SEEK CMD

023622                                  76$:


023622  032702  000200                          BIT     #BIT7,R2            ;SEE IF DOING NEG OFFSETS
023626  001014                                  BNE     18$                 ;BR IF YES
```

```
023630  005202                          INC     R2
023632  020227  000061                  CMP     R2,#61          ;SEE IF JUST DID MAX POS OFFSET
023636  001402                          BEQ     20$             ;BR IF YES
023640  000137  022420                  JMP     1$              ;ELSE DO NEXT POS OFFSET

023644  012702  000201          20$:    MOV     #201,R2         ;SETUP NEG OFFSET FOR RKASOF
023650  012701  000101                  MOV     #101,R1         ;SETUP NEG OFFSET OFOR MSG A
023654  000137  022420                  JMP     1$              ;DO NEG OFFSET

023660  005201                  18$:    INC     R1
023662  005202                          INC     R2
023664  020227  000261                  CMP     R2,#261         ;SEE IF ALL NEG OFFSETS DONE
023670  001402                          BEQ     TST14           ;;GO TO NEXT TST
023672  000137  022420                  JMP     1$              ;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.
                                ;*
                                ;;*****************************************************************
023676  000004                  TST14:  SCOPE
023700  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
023706  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

023712  004737  034214                  JSR     PC,SUBCLR
023716  104024                          ERROR   24              ;CERR AFTER SCLR

023720  005037  001402                  CLR     SECTOR
023724  013765  001402  000006  11$:    MOV     SECTOR,RKDA(R5)
023732  012765  001500  000004          MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
023740  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUS ADDR INCR INHIB
023746  012765  177400  000002          MOV     #-256.,RKWC(R5) ;SECTOR 0 ONLY
                                                                ;WILL DO IMPLIED SEEK TO CYL 0
                                                                ;WAS ON CYL 1 FROM LAST TEST

023754  012737  000023  007354          MOV     #<WRDATA>,HCS1
023762  004737  032262                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
023766  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
023770  004737  033664                  JSR     PC,GSTAT        ;GET FRESH STATUS
023774  032737  100000  007354          BIT     #CERR,HCS1
```

```
024002  001465                          BEQ     67$                     ;BR IF NO ERRORS
024004  032737  000200  007370          BIT     #BSE,HER                ;SEE IF BAD SECTOR FLAG
024012  001421                          BEQ     65$                     ;BR IF NO
024014  004737  035700                  JSR     PC,TRUERR               ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
024020  000455                          BR      66$                     ;RETURN HERE IF NO
024022  005237  001402                  INC     SECTOR                  ;RETURN HERE IF YES
024026  023727  001402  000012          CMP     SECTOR,#10.             ;ARE 10 CONSEC. SECTORS BAD
024034  001003                          BNE     64$                     ;BR IF NO
024036  104046                          ERROR   46                      ;ABORTING TEST DETECTED 10 BAD SECTORS
024040  000137  025436                  JMP     10$                     ;BYPASS TEST
024044  012765  100000  000000  64$:    MOV     #CCLR,RKCS1(R5)         ;TRY ANOTHER SECTOR
024052  000137  023724                  JMP     11$
024056  104012                  65$:    ERROR   12                      ;CERR WITH WRITE DATA CMD
024060  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
024066  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
024072  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
024100  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
024106  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
024112  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
024120  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
024126  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
024132  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
024134  104052                          ERROR   52                      ;MSG A0 ERROR AFTER WRITE DATA CMD
024136  104023                          ERROR   23                      ;MSH B0 ERROR
024140  104053                          ERROR   53                      ;MSG A1 ERROR
024142  104025                          ERROR   25                      ;MSG B1 ERROR
024144  104401  045354                  TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
024150  000137  031452                  JMP     $EOP
024154  104063                  66$:    ERROR   63                      ;BAD SECTOR NOT LISTED IN TABLE
024156                          67$:
024156  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
024164  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
024170  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
024176  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
024204  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
024210  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
024216  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
024224  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
024230  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
024232  104052                          ERROR   52                      ;MSG A0 ERROR AFTER WRITE DATA CMD
024234  104023                          ERROR   23                      ;MSH B0 ERROR
024236  104053                          ERROR   53                      ;MSG A1 ERROR
024240  104025                          ERROR   25                      ;MSG B1 ERROR
024242  012765  001500  000004          MOV     #DATA1,RKBA(R5)
024250  052765  000020  000010          BIS     #BAI,RKCS2(R5)
024256  012765  177400  000002          MOV     #-256.,RKWC(R5)
024264  013765  001402  000006          MOV     SECTOR,RKDA(R5)
024272  012737  000031  007354          MOV     #<WRTCHK>,HCS1
024300  004737  032262                  JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
```

```
024304  104015                          ERROR   15                      ;NO RDY AFTER WRITE CHECK CMD
024306  004737  033664                  JSR     PC,GSTAT                ;GET FRESH STATUS
024312  032737  100000  007354          BIT     #CERR,HCS1
024320  001453                          BEQ     69$
024322  032737  040000  007356          BIT     #WCE,HCS2               ;SEE IF WRITE CHECK ERROR
024330  001410                          BEQ     68$
024332  016537  000024  001452          MOV     RKDB(R5),WD1            ;ACTUAL WORD FOR PRINTOUT
024340  013737  001500  001454          MOV     DATA1,WD2                       ;EXPECTED WORD FOR TYPEOUT
024346  104016                          ERROR   16                      ;WCE AFTER WRITE CMD
024350  000437                          BR      69$

024352  104022                  68$:    ERROR   22                      ;CERR AFTER WRITE CHECK CMD

024354  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
024362  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
024366  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
024374  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
024402  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
024406  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
024414  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

024422  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
024426  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
024430  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
024432  104031                          ERROR   31                      ;MSH B0 ERROR
024434  104060                          ERROR   60                      ;MSG A1 ERROR
024436  104032                          ERROR   32                      ;MSG B1 ERROR
024440  104401  045354                  TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
024444  000137  031452                  JMP     $EOP

024450                          69$:

024450  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
024456  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
024462  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
024470  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
024476  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
024502  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
024510  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

024516  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
024522  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
024524  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
024526  104031                          ERROR   31                      ;MSH B0 ERROR
024530  104060                          ERROR   60                      ;MSG A1 ERROR
024532  104032                          ERROR   32                      ;MSG B1 ERROR

024534  104401  044641                  TYPE    ,MSG8                   ;READ WITH OFFSET TEST
024540  005001                          CLR     R1                      ;HEAD #

024542  012700  000001          9$:     MOV     #1,R0                   ;INIT OFFSET COUNTER
024546  104401  044700                  TYPE    ,MSG9                   ;HEAD #
024552  010146                          MOV     R1,-(SP)                ;;SAVE R1 FOR TYPEOUT
                                                                        ;;TYPE HEAD #
024554  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
024556     001                          .BYTE   1                       ;;TYPE 1 DIGIT(S)
```

```
024557     000                                 .BYTE   0                       ;;SUPPRESS LEADING ZEROS
024560  104401  001205                         TYPE    ,$CRLF

024564  005037  001456                 1$:     CLR     OFFERR                  ;WRITE CHECK ERROR FLAG

024570  004737  034214                         JSR     PC,SUBCLR
024574  104024                                 ERROR   24                      ;CERR AFTER SCLR

024576  010065  000016                         MOV     RO,RKASOF(R5)           ;OFFSET VALUE
024602  000301                                 SWAB    R1
024604  010165  000006                         MOV     R1,RKDA(R5)             ;HEAD NO.
024610  000301                                 SWAB    R1

024612  012737  024700  001176                 MOV     #70$,$ESCAPE
024620  012737  000015  007354                 MOV     #OFFSET,HCS1
024626  004737  032224                         JSR     PC,DOCMD                ;DO RECAL CMD & GET CONTR RDY
024632  104033                                 ERROR   33                      ;NO RDY AFTER OFFSET CMD

024634  012737  032140  007444                 MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
024642  005037  007446                         CLR     E.B0
024646  012737  001720  007450                 MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
024654  012737  000001  007452                 MOV     #1,E.B1

024662  004737  033042                         JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
024666  000000                                 .WORD   0!0!0                   ;& MSGS SPECIFIED HERE
024670  104035                                 ERROR   35                      ;MSG A0 ERROR DURING OFFSET CMD
024672  104061                                 ERROR   61                      ;MSH B0 ERROR
024674  104036                                 ERROR   36                      ;MSG A1 ERROR
024676  104062                                 ERROR   62                      ;MSG B1 ERROR

024700  005037  001176                 70$:    CLR     $ESCAPE
024704  013737  001416  007412                 MOV     T5000,TEMP1             ;SETUP TIMEOUT
024712  004737  032730                         JSR     PC,FATT2                ;FIND ATTN
024716  104034                                 ERROR   34                      ;NO ATTN AFTER OFFSET CMD


024720  012737  052340  007444                 MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
024726  005037  007446                         CLR     E.B0                    ;EXPECTED MSG B0
024732  012737  001720  007450                 MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
024740  012737  000001  007452                 MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
024746  005037  007454                         CLR     E.A2                    ;EXPECTED MSG A2
024752  012737  000002  007456                 MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
024760  012737  000003  007462                 MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

024766  004737  033042                         JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
024772  000003                                 .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
024774  104260                                 ERROR   260                     ;MSG A0 ERROR AFTER OFFSET CMD
024776  104261                                 ERROR   261                     ;MSH B0 ERROR
025000  104037                                 ERROR   37                      ;MSG A1 ERROR
025002  104040                                 ERROR   40                      ;MSG B1 ERROR


025004  012765  100000  000000                 MOV     #CCLR,RKCS1(R5)
025012  013765  001222  000010                 MOV     $UNIT,RKCS2(R5)         ;DRIVE#
025020  012737  000005  007354                 MOV     #CLEAR,HCS1
025026  004737  032224                         JSR     PC,DOCMD                ;DO DRIVE CLEAR CMD & GET CONTR RDY
```

```
025032  104151                             ERROR   151              ;NO RDY AFTER DRIVE CLEAR CMD
025034  004737  032602                     JSR     PC,TSTATN        ;TEST FOR ATTN
025040  000401                             BR      71$
025042  104154                             ERROR   154              ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
025044                             71$:
025044  012765  001500  000004             MOV     #DATA1,RKBA(R5)
025052  052765  000020  000010             BIS     #BAI,RKCS2(R5)
025060  012765  177400  000002             MOV     #-256.,RKWC(R5)
025066  013765  001402  000006             MOV     SECTOR,RKDA(R5)
025074  012737  000031  007354             MOV     #WRTCHK,HCS1
025102  004737  032262                     JSR     PC,DATCMD        ;DO WRITE CHECK CMD. & GET CONTR RDY.
025106  104015                             ERROR   15               ;NO RDY AFTER WRITE CHECK CMD
025110  004737  033664                     JSR     PC,GSTAT         ;GET FRESH STATUS
025114  032737  040000  007356             BIT     #WCE,HCS2
025122  001421                             BEQ     2$
025124  016537  000024  001452             MOV     RKDB(R5),WD1     ;GET MISCOMPARED WORD
025132  005237  001456                     INC     OFFERR           ;BAD WRITE CHK ERROR=SET ERR FLG.
025136  005737  001456                     TST     OFFERR
025142  001411                             BEQ     2$
025144  104401  045620                     TYPE    ,MSG39           ;WRITE CHECK FAILURE AT OFFSET
025150  010046                             MOV     R0,-(SP)         ;;SAVE R0 FOR TYPEOUT
                                                                    ;;TYPE OFFSET VALUE
025152  104403                             TYPOS                    ;;GO TYPE--OCTAL ASCII
025154     006                             .BYTE   6                ;;TYPE 6 DIGITS
025155     000                             .BYTE   0                ;;SUPPRESS LEADING ZEROS
025156  104401  001205                     TYPE    ,$CRLF
025162  104401  001205                     TYPE    ,$CRLF
025166  032700  000200             2$:     BIT     #BIT7,R0         ;SEE IF OFFSET IS + OR -
025172  001023                             BNE     5$               ;BR IF - OFFSET
025174  020027  000060                     CMP     R0,#60
025200  001412                             BEQ     4$
025202  005737  001456                     TST     OFFERR
025206  001404                             BEQ     3$
025210  012700  000200             8$:     MOV     #200,R0          ;SETUP FOR NEG OFFSET
025214  000137  024564                     JMP     1$
025220  005200                     3$:     INC     R0
025222  000137  024564                     JMP     1$
025226  005737  001456             4$:     TST     OFFERR
025232  001366                             BNE     8$               ;DO NEG OFFSETS
025234  104401  045462                     TYPE    ,MSG37           ;NO WRITE CHECK ERROR AT MAX POS OFFSET
                                                                    ;NOTE! EITHER HEADS DID NOT MOVE
                                                                    ;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
025240  000763                             BR      8$               ;DO NEG OFFSETS
025242  020027  000260             5$:     CMP     R0,#260
025246  001404                             BEQ     6$
025250  005737  001456                     TST     OFFERR
025254  001072                             BNE     TST15            ;;GO TO NEXT TST
025256  000760                             BR      3$
```

```
025260  005737  001456           6$:     TST     OFFERR
025264  001002                           BNE     7$
025266  104401  045540                   TYPE    ,MSG38          ;NO WRITE CHECK ERROR AT MAX NEG OFFSET
                                                                 ;NOTE! EITHER HEADS DID NOT MOVE
                                                                 ;OR READ/WRITE AMP IS EXECEPTIONALLY GOOD.
025272                           7$:

025272  012765  100000  000000           MOV     #CCLR,RKCS1(R5)
025300  013765  001222  000010           MOV     $UNIT,RKCS2(R5)
025306  012737  000013  007354           MOV     #RECAL,HCS1
025314  004737  032224                   JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
025320  104124                           ERROR   124             ;RDY NOT SET AFTER RECAL CMD

025322  012765  000001  000026           MOV     #1,RKMR1(R5)    ;SELECT WORD 1
025330  004737  033664                   JSR     PC,GSTAT
025334  032737  020000  007402           BIT     #D.RTZ,HMR2
025342  001001                           BNE     72$
025344  104244                           ERROR   244             ;RTZ NOT SET DURING RECAL CMD
025346  013737  001406  007414   72$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
025354  004737  032634                   JSR     PC,FATT1        ;FIND ATTN
025360  104055                           ERROR   55              ;NO ATTN AFTER RECAL CMD

025362  012765  100000  000000           MOV     #CCLR,RKCS1(R5)
025370  013765  001222  000010           MOV     $UNIT,RKCS2(R5) ;DRIVE#
025376  012737  000005  007354           MOV     #CLEAR,HCS1
025404  004737  032224                   JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
025410  104151                           ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
025412  004737  032602                   JSR     PC,TSTATN       ;TEST FOR ATTN
025416  000401                           BR      73$
025420  104154                           ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
025422                           73$:


025422  005201                           INC     R1              ;HEAD CTR
025424  020127  000003                   CMP     R1,#3           ;SEE IF ALL HEADS DONE
025430  001402                           BEQ     10$             ;BR IF YES
025432  000137  024542                   JMP     9$              ;ELSE REPEAT ALL FOR NEXT HEAD
025436  104401  045000           10$:    TYPE    ,MSG12          ;OFFSET FAILURES ARE NOT ERRORS
                                 ;;*****************************************************************
                                 ;*TEST 15        WRITE WITH HEADS OFFSET
                                 ;*
                                 ;*       THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
                                 ;*       WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
                                 ;*       & THE DRIVE WILL WRITE
                                 ;*       SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
                                 ;*       THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY
                                 ;*
                                 ;;*****************************************************************
025442  000004                   TST15:  SCOPE
025444  012737  000001  001174           MOV     #1,$TIMES       ;;DO 1 ITERATION
025452  012706  001100                   MOV     #STACK,SP       ;RESTORE STK PTR

025456  012700  000260                   MOV     #260,R0         ;MAX NEG OFFSET

025462  004737  034214           1$:     JSR     PC,SUBCLR
```

```
025466  104024                          ERROR   24                      ;CERR AFTER SCLR
025470  010065  000016                  MOV     R0,RKASOF(R5)           ;SET OFFSET
025474  012737  025562  001176          MOV     #64$,$ESCAPE
025502  012737  000015  007354          MOV     #OFFSET,HCS1
025510  004737  032224                  JSR     PC,DOCMD                ;DO RECAL CMD & GET CONTR RDY
025514  104033                          ERROR   33                      ;NO RDY AFTER OFFSET CMD
025516  012737  032140  007444          MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
025524  005037  007446                  CLR     E.B0
025530  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
025536  012737  000001  007452          MOV     #1,E.B1
025544  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
025550  000000                          .WORD   0!0!0                   ;& MSGS SPECIFIED HERE
025552  104035                          ERROR   35                      ;MSG A0 ERROR DURING OFFSET CMD
025554  104061                          ERROR   61                      ;MSH B0 ERROR
025556  104036                          ERROR   36                      ;MSG A1 ERROR
025560  104062                          ERROR   62                      ;MSG B1 ERROR
025562  005037  001176          64$:    CLR     $ESCAPE
025566  013737  001416  007412          MOV     T5000,TEMP1             ;SETUP TIMEOUT
025574  004737  032730                  JSR     PC,FATT2                ;FIND ATTN
025600  104034                          ERROR   34                      ;NO ATTN AFTER OFFSET CMD
025602  012737  052340  007444          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
025610  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
025614  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
025622  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
025630  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
025634  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
025642  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
025650  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
025654  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
025656  104260                          ERROR   260                     ;MSG A0 ERROR AFTER OFFSET CMD
025660  104261                          ERROR   261                     ;MSH B0 ERROR
025662  104037                          ERROR   37                      ;MSG A1 ERROR
025664  104040                          ERROR   40                      ;MSG B1 ERROR
025666  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
025674  013765  001222  000010          MOV     $UNIT,RKCS2(R5)         ;DRIVE#
025702  012737  000005  007354          MOV     #CLEAR,HCS1
025710  004737  032224                  JSR     PC,DOCMD                ;DO DRIVE CLEAR CMD & GET CONTR RDY
025714  104151                          ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
025716  004737  032602                  JSR     PC,TSTATN               ;TEST FOR ATTN
025722  000401                          BR      65$
025724  104154                          ERROR   154                     ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
025726                          65$:
025726  005037  001402                  CLR     SECTOR
025732  013765  001402  000006  4$:     MOV     SECTOR,RKDA(R5)
```

```
025740  012765  001474  000004          MOV     #DATA0,RKBA(R5)  ;WRITE ALL 0'S
025746  052765  000020  000010          BIS     #BAI,RKCS2(R5)   ;BUS ADDR INCR INH
025754  012765  177400  000002          MOV     #-256.,RKWC(R5)  ;FULL SECTOR
025762  012737  000023  007354          MOV     #<WRDATA>,HCS1
025770  004737  032262                  JSR     PC,DATCMD        ;DO DATA X FOR CMD & GET CONTR RDY
025774  104011                          ERROR   11               ;NO RDY AFTER WRITE DATA CMD
025776  004737  033664                  JSR     PC,GSTAT         ;GET FRESH STATUS
026002  032737  100000  007354          BIT     #CERR,HCS1
026010  001465                          BEQ     69$              ;BR IF NO ERRORS
026012  032737  000200  007370          BIT     #BSE,HER         ;SEE IF BAD SECTOR FLAG
026020  001421                          BEQ     67$              ;BR IF NO
026022  004737  035700                  JSR     PC,TRUERR        ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
026026  000455                          BR      68$              ;RETURN HERE IF NO
026030  005237  001402                  INC     SECTOR           ;RETURN HERE IF YES
026034  023727  001402  000012          CMP     SECTOR,#10.      ;ARE 10 CONSEC. SECTORS BAD
026042  001003                          BNE     66$              ;BR IF NO
026044  104046                          ERROR   46               ;ABORTING TEST DETECTED 10 BAD SECTORS
026046  000137  026756                  JMP     3$               ;BYPASS TEST
026052  012765  100000  000000  66$:    MOV     #CCLR,RKCS1(R5)  ;TRY ANOTHER SECTOR
026060  000137  025732                  JMP     4$
026064  104012                  67$:    ERROR   12               ;CERR WITH WRITE DATA CMD
026066  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
026074  005037  007446                  CLR     E.B0             ;EXPECTED MSG B0
026100  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
026106  012737  000001  007452          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
026114  005037  007454                  CLR     E.A2             ;EXPECTED MSG A2
026120  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
026126  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
026134  004737  033042                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
026140  000003                          .WORD   T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
026142  104052                          ERROR   52               ;MSG A0 ERROR AFTER WRITE DATA CMD
026144  104023                          ERROR   23               ;MSH B0 ERROR
026146  104053                          ERROR   53               ;MSG A1 ERROR
026150  104025                          ERROR   25               ;MSG B1 ERROR
026152  104401  045354                  TYPE    ,MSG26           ;ABORTING BAL OF TESTS
026156  000137  031452                  JMP     $EOP
026162  104063                  68$:    ERROR   63               ;BAD SECTOR NOT LISTED IN TABLE
026164                          69$:
026164  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
026172  005037  007446                  CLR     E.B0             ;EXPECTED MSG B0
026176  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
026204  012737  000001  007452          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
026212  005037  007454                  CLR     E.A2             ;EXPECTED MSG A2
026216  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
026224  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
026232  004737  033042                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
026236  000003                          .WORD   T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
026240  104052                          ERROR   52               ;MSG A0 ERROR AFTER WRITE DATA CMD
026242  104023                          ERROR   23               ;MSH B0 ERROR
```

```
026244  104053                                  ERROR   53                      ;MSG A1 ERROR
026246  104025                                  ERROR   25                      ;MSG B1 ERROR

026250  012765  000002  000026                  MOV     #2,RKMR1(R5)            ;SELECT WORD 2
026256  004737  033664                          JSR     PC,GSTAT
026262  005737  001356                          TST     CYLDIF                  ;SEE IF MSG A2=0
026266  001401                                  BEQ     70$                     ;BR IF YES
026270  104104                                  ERROR   104                     ;MSG A2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
026272  005737  001360                  70$:    TST     CYLADD                  ;SEE IF MSG B2=0
026276  001401                                  BEQ     71$                     ;BR IF YES
026300  104105                                  ERROR   105                     ;MSG B2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
026302                                  71$:

026302  104415                                  SCOP1
026304  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR

026310  004737  034214                          JSR     PC,SUBCLR
026314  104024                                  ERROR   24                      ;CERR AFTER SCLR


026316  012765  001474  000004                  MOV     #DATA0,RKBA(R5)
026324  052765  000020  000010                  BIS     #BAI,RKCS2(R5)
026332  012765  177400  000002                  MOV     #-256.,RKWC(R5)
026340  013765  001402  000006                  MOV     SECTOR,RKDA(R5)

026346  012737  000031  007354                  MOV     #<WRTCHK>,HCS1
026354  004737  032262                          JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
026360  104015                                  ERROR   15                      ;NO RDY AFTER WRITE CHECK CMD
026362  004737  033664                          JSR     PC,GSTAT                ;GET FRESH STATUS
026366  032737  100000  007354                  BIT     #CERR,HCS1
026374  001453                                  BEQ     73$
026376  032737  040000  007356                  BIT     #WCE,HCS2               ;SEE IF WRITE CHECK ERROR
026404  001410                                  BEQ     72$
026406  016537  000024  001452                  MOV     RKDB(R5),WD1            ;ACTUAL WORD FOR PRINTOUT
026414  013737  001474  001454                  MOV     DATA0,WD2
                                                                                        ;EXPECTED WORD FOR TYPEOUT
026422  104016                                  ERROR   16                      ;WCE AFTER WRITE CMD
026424  000437                                  BR      73$

026426  104022                          72$:    ERROR   22                      ;CERR AFTER WRITE CHECK CMD

026430  012737  010340  007444                  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
026436  005037  007446                          CLR     E.B0                    ;EXPECTED MSG B0
026442  012737  001720  007450                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
026450  012737  000001  007452                  MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
026456  005037  007454                          CLR     E.A2                    ;EXPECTED MSG A2
026462  012737  000002  007456                  MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
026470  012737  000003  007462                  MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

026476  004737  033042                          JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
026502  000003                                  .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
026504  104057                                  ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
026506  104031                                  ERROR   31                      ;MSH B0 ERROR
026510  104060                                  ERROR   60                      ;MSG A1 ERROR
026512  104032                                  ERROR   32                      ;MSG B1 ERROR
026514  104401  045354                          TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
026520  000137  031452                          JMP     $EOP
```

```
026524                            73$:
026524  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
026532  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
026536  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
026544  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
026552  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
026556  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
026564  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

026572  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
026576  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
026600  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
026602  104031                          ERROR   31                      ;MSH B0 ERROR
026604  104060                          ERROR   60                      ;MSG A1 ERROR
026606  104032                          ERROR   32                      ;MSG B1 ERROR

026610  020027  000260                  CMP     R0,#260
026614  001004                          BNE     2$                      ;BR IF JUST DID POS OFFSET
026616  012700  000060                  MOV     #60,R0                  ;ELSE SETUP FOR POS OFFSET
026622  000137  025462                  JMP     1$

026626                            2$:

026626  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
026634  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
026642  012737  000013  007354          MOV     #RECAL,HCS1
026650  004737  032224                  JSR     PC,DOCMD                ;DO RECAL CMD & GET CONTR RDY
026654  104124                          ERROR   124                     ;RDY NOT SET AFTER RECAL CMD

026656  012765  000001  000026          MOV     #1,RKMR1(R5)            ;SELECT WORD 1
026664  004737  033664                  JSR     PC,GSTAT
026670  032737  020000  007402          BIT     #D.RTZ,HMR2
026676  001001                          BNE     74$
026700  104244                          ERROR   244                     ;RTZ NOT SET DURING RECAL CMD
026702  013737  001406  007414    74$:  MOV     T10,TEMP2               ;SETUP TIMEOUT
026710  004737  032634                  JSR     PC,FATT1                ;FIND ATTN
026714  104055                          ERROR   55                      ;NO ATTN AFTER RECAL CMD

026716  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
026724  013765  001222  000010          MOV     $UNIT,RKCS2(R5)         ;DRIVE#
026732  012737  000005  007354          MOV     #CLEAR,HCS1
026740  004737  032224                  JSR     PC,DOCMD                ;DO DRIVE CLEAR CMD & GET CONTR RDY
026744  104151                          ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
026746  004737  032602                  JSR     PC,TSTATN               ;TEST FOR ATTN
026752  000401                          BR      75$
026754  104154                          ERROR   154                     ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
026756                            75$:


026756                            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  RK06
                                                ;*      CYLINDER Y:  64  128  192  256  320  384  RK06
                                                ;*
                                                ;*      CYLINDER X:  127 255  383  511  639  767  RK07
                                                ;*      CYLINDER Y:  128 256  384  512  640  768  RK07
                                                ;*
                                                ;*      ALL ABOVE CYLINDERS IN DECIMAL
                                                ;*
                                                ;;*******************************************************************
 026756  000004                                 TST16:  SCOPE
 026760  012737  000001  001174                         MOV     #1,$TIMES           ;;DO 1 ITERATION
 026766  012706  001100                                 MOV     #STACK,SP

 026772  005737  001170                                 TST     $TMP4               ;SEE IF RK07
 026776  001403                                         BEQ     5$                  ;BR IF NO
 027000  012700  001436                                 MOV     #CYL7,R0            ;ELSE LOAD UP RK07 TABLE
 027004  000402                                         BR      1$
 027006  012700  001422                         5$:     MOV     #CYL,R0             ;RK06 CYL ADDR TABLE

 027012  004737  034214                         1$:     JSR     PC,SUBCLR
 027016  104024                                         ERROR   24                  ;CERR AFTER SCLR

 027020  011065  000020                                 MOV     (R0),RKDC(R5)       ;CYL #
 027024  012765  001502  000004                         MOV     #DPAT1,RKBA(R5)     ;DATA PATTERN
 027032  052765  000020  000010                         BIS     #BAI,RKCS2(R5)      ;BUSS ADDR INCREMENT INHIBIT
 027040  012765  076000  000002                         MOV     #-6*22.*256.,RKWC(R5) ;WORD COUNT TO SPIRAL & FILL 2 CYLINDERS

 027046  012737  000023  007354                         MOV     #<WRDATA>,HCS1
 027054  004737  032262                                 JSR     PC,DATCMD           ;DO DATA X FOR CMD & GET CONTR RDY
 027060  104011                                         ERROR   11                  ;NO RDY AFTER WRITE DATA CMD
 027062  004737  033664                                 JSR     PC,GSTAT            ;GET FRESH STATUS
 027066  032737  100000  007354                         BIT     #CERR,HCS1
 027074  001451                                         BEQ     67$                 ;BR IF NO ERRORS

 027076  032737  000200  007370                         BIT     #BSE,HER            ;SEE IF BAD SECTOR FLAG
 027104  001405                                         BEQ     65$                 ;BR IF NO
 027106  004737  035700                                 JSR     PC,TRUERR           ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
 027112  000441                                         BR      66$                 ;RETURN HERE IF NO
 027114  000137  027574                                 JMP     3$                  ;RET HERE IF YES
 027120  104012                                 65$:    ERROR   12                  ;CERR WITH WRITE DATA CMD

 027122  012737  010340  007444                         MOV     #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.AO  ;EXPECTED MSG AO
 027130  005037  007446                                 CLR     E.BO                ;EXPECTED MSG BO
```

```
027134  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
027142  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
027150  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
027154  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
027162  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

027170  004737  033042                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
027174  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
027176  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
027200  104023                          ERROR   23              ;MSH B0 ERROR
027202  104053                          ERROR   53              ;MSG A1 ERROR
027204  104025                          ERROR   25              ;MSG B1 ERROR
027206  104401  045354                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
027212  000137  031452                  JMP     $EOP
027216  104063                  66$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
027220                          67$:

027220  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
027226  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
027232  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
027240  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
027246  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
027252  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
027260  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

027266  004737  033042                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
027272  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
027274  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
027276  104023                          ERROR   23              ;MSH B0 ERROR
027300  104053                          ERROR   53              ;MSG A1 ERROR
027302  104025                          ERROR   25              ;MSG B1 ERROR

027304  011065  000020                  MOV     (R0),RKDC(R5)
027310  012765  001502  000004          MOV     #DPAT1,RKBA(R5)
027316  052765  000020  000010          BIS     #BAI,RKCS2(R5)
027324  012765  076000  000002          MOV     #-6*22.*256.,RKWC(R5)

027332  012737  000031  007354          MOV     #<WRTCHK>,HCS1
027340  004737  032262                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
027344  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
027346  004737  033664                  JSR     PC,GSTAT        ;GET FRESH STATUS
027352  032737  100000  007354          BIT     #CERR,HCS1
027360  001453                          BEQ     69$
027362  032737  040000  007356          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
027370  001410                          BEQ     68$
027372  016537  000024  001452          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
027400  013737  001502  001454          MOV     DPAT1,WD2               ;EXPECTED WORD FOR TYPEOUT
027406  104016                          ERROR   16              ;WCE AFTER WRITE CMD
027410  000437                          BR      69$

027412  104022                  68$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD

027414  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
027422  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
027426  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
027434  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
```

```
027442  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
027446  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
027454  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

027462  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
027466  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
027470  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
027472  104031                          ERROR   31                      ;MSH B0 ERROR
027474  104060                          ERROR   60                      ;MSG A1 ERROR
027476  104032                          ERROR   32                      ;MSG B1 ERROR
027500  104401  045354                  TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
027504  000137  031452                  JMP     $EOP

027510                          69$:

027510  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0   ;EXPECTED MSG A0
027516  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
027522  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1   ;EXPECTED A1
027530  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
027536  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
027542  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
027550  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3

027556  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
027562  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
027564  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
027566  104031                          ERROR   31                      ;MSH B0 ERROR
027570  104060                          ERROR   60                      ;MSG A1 ERROR
027572  104032                          ERROR   32                      ;MSG B1 ERROR
027574  022027  000577          3$:     CMP     (R0)+,#383.             ;ALL CYLINDERS DONE?
027600  001402                          BEQ     4$                      ;BR IF YES
027602  000137  027012                  JMP     1$                      ;ELSE REPEAT

027606                          4$:

027606  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
027614  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
027622  012737  000013  007354          MOV     #RECAL,HCS1
027630  004737  032224                  JSR     PC,DOCMD                ;DO RECAL CMD & GET CONTR RDY
027634  104124                          ERROR   124                     ;RDY NOT SET AFTER RECAL CMD

027636  012765  000001  000026          MOV     #1,RKMR1(R5)            ;SELECT WORD 1
027644  004737  033664                  JSR     PC,GSTAT
027650  032737  020000  007402          BIT     #D.RTZ,HMR2
027656  001001                          BNE     70$
027660  104244                          ERROR   244                     ;RTZ NOT SET DURING RECAL CMD
027662  013737  001406  007414  70$:    MOV     T10,TEMP2               ;SETUP TIMEOUT
027670  004737  032634                  JSR     PC,FATT1                ;FIND ATTN
027674  104055                          ERROR   55                      ;NO ATTN AFTER RECAL CMD

027676  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
027704  013765  001222  000010          MOV     $UNIT,RKCS2(R5)         ;DRIVE#
027712  012737  000005  007354          MOV     #CLEAR,HCS1
027720  004737  032224                  JSR     PC,DOCMD                ;DO DRIVE CLEAR CMD & GET CONTR RDY
027724  104151                          ERROR   151                     ;NO RDY AFTER DRIVE CLEAR CMD
027726  004737  032602                  JSR     PC,TSTATN               ;TEST FOR ATTN
```

```
027732  000401                                  BR      71$
027734  104154                                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
027736                                  71$:


027736                                  2$:

                                        ;;**************************************************************
                                        ;*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
                                        ;*
                                        ;;**************************************************************
027736  000004                          TST17:  SCOPE
027740  012737  000001  001174                  MOV     #1,$TIMES       ;;DO 1 ITERATION
027746  012706  001100                          MOV     #STACK,SP

027752  005737  007522                          TST     DOTIM           ;BYPASS THIS TEST IF
```

```
 5630  027756  001001                BNE     1$                     ;NEITHER L OR P CLOCK PRESENT
 5631  027760  000526                BR      TST20                  ;;GO TO NEXT TEST
```

```
027762  012737  000025  007422  1$:     MOV     #25,TEMP5              ;HEAD 0, SECTOR 21 TO BE PUT IN RKDA
027770  004737  034214          2$:     JSR     PC,SUBCLR
027774  104024                          ERROR   24                     ;CERR AFTER SCLR
027776  004737  035000                  JSR     PC,FSEC23              ;FIND SECTOR 23
030002  104106                          ERROR   106                    ;CANNOT FIND SECTOR 23
030004  000514                          BR      TST20                  ;;GO TO NEXT TEST
030006  012765  001476  000004          MOV     #DATA01,RKBA(R5)       ;DATA 0101
030014  052765  000020  000010          BIS     #BAI,RKCS2(R5)         ;BUSS ADDR INCREMENT INHIBIT
030022  012765  177000  000002          MOV     #-512.,RKWC(R5)        ;WORD COUNT
030030  013765  007422  000006          MOV     TEMP5,RKDA(R5)         ;HEAD & SECTOR
030036  012737  000023  007354          MOV     #WRDATA,HCS1
030044  053737  001170  007354          BIS     $TMP4,HCS1
030052  013765  007354  000000          MOV     HCS1,RKCS1(R5)         ;DO WRITE DATA CMD
030060  012737  004000  007412          MOV     #4000,TEMP1            ;2000 IS IN VER 30,
                                                                       ;CHANGE TO 4000 7-DEC-77
030066  004737  033010                  JSR     PC,DLY                 ;DO DELAY
030072  032765  000200  000000  7$:     BIT     #RDY,RKCS1(R5)         ;LOOK FOR CONTROLLER READY
030100  001006                          BNE     8$
030102  004737  032320                  JSR     PC,FRDY                ;FIND RDY AND GET FRESH STATUS
030106  104011                          ERROR   11                     ;NO RDY AFTER SEL DRV CMD
030110  004737  033664                  JSR     PC,GSTAT
030114  104107                          ERROR   107                    ;HEAD SWITCHING LONGER THAN DELAY
030116  032737  100000  007354  8$:     BIT     #CERR,HCS1
030124  001444                          BEQ     3$
030126  104012                          ERROR   12                     ;CERR AFTER WRITE DATA
030130  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
030136  005037  007446                  CLR     E.B0                   ;EXPECTED MSG B0
030142  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
030150  012737  000001  007452          MOV     #1,E.B1                ;MSG ID FOR EXPECTED MSG B1
030156  005037  007454                  CLR     E.A2                   ;EXPECTED MSG A2
030162  012737  000002  007456          MOV     #2,E.B2                ;MSG ID FOR EXPECTED MSG B2
030170  012737  000003  007462          MOV     #3,E.B3                ;MSG ID FOR EXPECTED MSG B3
030176  004737  033042                  JSR     PC,CHKMSG              ;CHECK MSGS A0, B0, A1, B1
030202  000000                          .WORD   0!0!0                  ;& MSGS SPECIFIED HERE
030204  104052                          ERROR   52                     ;MSG A0 ERROR AFTER WRITE DATA CMD
030206  104023                          ERROR   23                     ;MSH B0 ERROR
030210  104053                          ERROR   53                     ;MSG A1 ERROR
030212  104025                          ERROR   25                     ;MSG B1 ERROR
030214  023727  007422  000425          CMP     TEMP5,#425             ;HEAD 1,SECTOR 21 DONE?
030222  001405                          BEQ     TST20                  ;;GO TO NEXT TEST
030224  012737  000425  007422          MOV     #425,TEMP5
030232  000137  027770                  JMP     2$                     ;ELSE REPEAT FOR HEAD 1, SECTOR 21
030236                          3$:
                                ;;*****************************************************************
                                ;*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 1.
                                        ;*          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 CYLINDERS.
                                        ;*      100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
                                        ;*      BEFORE DOING THE NEXT CYLINDER
                                        ;*
                                        ;;*********************************************************
030236  000004                          TST20:  SCOPE
030240  012737  000001  001174                  MOV     #1,$TIMES       ;;DO 1 ITERATION
030246  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR
030252  005237  007340                          INC     BADHDR          ;USED FOR VALID HALT
030256  005037  001346                          CLR     TOCYL
030262  012737  100000  007422                  MOV     #BIT15,TEMP5

030270  004737  034214                  1$:     JSR     PC,SUBCLR
030274  104024                                  ERROR   24              ;CERR AFTER SCLR

030276  012700  001522                          MOV     #HDTAB,R0       ;FORMAT HEADERS ON ALL MAJOR CYL.

030302  013720  001346                  2$:     MOV     TOCYL,(R0)+     ;HEADER WORD 0: CYL #
030306  012720  140000                          MOV     #140000,(R0)+   ;HEADER WORD 1: ALL SECTOR 0
030312  012710  140000                          MOV     #140000,(R0)    ;HEADER WORD 2: XOR OF 0 & 1
030316  053720  001346                          BIS     TOCYL,(R0)+     ;ADD CYL # TO WORD 2

030322  020027  001726                          CMP     R0,#HDTAB+132.  ;ALL 22 SECTORS DONE? (22X6=132)
030326  001365                                  BNE     2$              ;BR IF NO

030330  012765  001522  000004                  MOV     #HDTAB,RKBA(R5)
030336  012765  177676  000002                  MOV     #-66.,RKWC(R5)
030344  013765  001346  000020                  MOV     TOCYL,RKDC(R5)

030352  012737  000027  007354                  MOV     #<WRHEAD>,HCS1
030360  004737  032262                          JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
030364  104200                                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
030366  004737  033664                          JSR     PC,GSTAT        ;GET FRESH STATUS
030372  032737  100000  007354                  BIT     #CERR,HCS1
030400  001405                                  BEQ     64$
030402  104201                                  ERROR   201             ;CERR AFTER WRITE HEADER CMD
030404  104401  045354                          TYPE    ,MSG26          ;ABORTING BAL OF TESTS
030410  000137  031452                          JMP     $EOP
030414                                  64$:


030414  006137  007422                          ROL     TEMP5           ;SET CARRY ONLY ONCE
030420  006137  001346                          ROL     TOCYL           ;SELECT NEXT MAJOR CYL
030424  023737  001346  015436                  CMP     TOCYL,MC1       ;ALL MAJOR CYL FORMATTED?
```

```
030432  001316                                  BNE     1$                      ;BR IF NO
030434  005065  000020                          CLR     RKDC(R5)                ;SETUP TO RETURN TO CYL 0

030440  012737  000017  007354                  MOV     #SEEK,HCS1
030446  004737  032224                          JSR     PC,DOCMD                ;DO SEEK CMD & GET CONTR READY
030452  104131                                  ERROR   131                     ;NO RDY AFTER SEEK CMD

030454  013737  001420  007412                  MOV     T50000,TEMP1            ;SETUP TIMEOUT
030462  004737  032730                          JSR     PC,FATT2                ;FIND ATTN
030466  104132                                  ERROR   132                     ;NO ATTN AFTER SEEK CMD

030470  032737  100000  007354                  BIT     #CERR,HCS1
030476  001401                                  BEQ     65$
030500  104210                                  ERROR   210                     ;CERR AFTER SEEK CMD

030502                                  65$:

030502  005000                                  CLR     R0                      ;ITERATION COUNTER
030504  012737  000001  001346                  MOV     #1,TOCYL                ;SETUP TO CYL #
030512  005037  001344                          CLR     FRCYL

030516  104415                                  SCOP1
030520  012706  001100                          MOV     #STACK,SP               ;RESTORE STK PTR

030524  013737  001346  001354                  MOV     TOCYL,CALDIF            ;SETUP FOR ERROR PRINTOUT

030532  004737  034214                  3$:     JSR     PC,SUBCLR
030536  104024                                  ERROR   24                      ;CERR AFTER SCLR

030540  012737  031174  001176                  MOV     #FORM,$ESCAPE
030546  013765  001346  000020                  MOV     TOCYL,RKDC(R5)          ;GO TO CYL #
030554  012765  001500  000004                  MOV     #DATA1,RKBA(R5)         ;ALL 1'S
030562  052765  000020  000010                  BIS     #BAI,RKCS2(R5)
030570  012765  177400  000002                  MOV     #-256.,RKWC(R5)         ; SECTOR TO BE ALL 1'S

030576  012737  000023  007354                  MOV     #WRDATA,HCS1
030604  004737  032262                          JSR     PC,DATCMD               ;DO WRITE DATA CMD & GET CONTR RDY
030610  104011                                  ERROR   11                      ;NO RDY AFTER WRITE DATA CMD.

030612  004737  033664                          JSR     PC,GSTAT                ;GET FRESH STATUS
030616  032737  020000  007404                  BIT     #D.DROT,HMR3            ;SEE IF DRIVE OFF TRACK
030624  001401                                  BEQ     5$
030626  104112                                  ERROR   112                     ;DRIVE OFF TRACK AFTER WRITE DATA CMD

030630  032737  100000  007354          5$:     BIT     #CERR,HCS1
030636  001401                                  BEQ     6$
030640  104012                                  ERROR   12                      ;CERR SET AFTER WRITE DATA CMD

030642                                  6$:

030642  012737  010340  007444                  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
030650  005037  007446                          CLR     E.B0                    ;EXPECTED MSG B0
030654  012737  001720  007450                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
030662  012737  000001  007452                  MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
030670  005037  007454                          CLR     E.A2                    ;EXPECTED MSG A2
030674  012737  000002  007456                  MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
```

```
030702  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
030710  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
030714  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
030716  104052                          ERROR   52                      ;MSG A0 ERROR AFTER WRITE DATA CMD
030720  104023                          ERROR   23                      ;MSH B0 ERROR
030722  104053                          ERROR   53                      ;MSG A1 ERROR
030724  104025                          ERROR   25                      ;MSG B1 ERROR
030726  023737  001360  001346          CMP     CYLADD,TOCYL
030734  001401                          BEQ     7$
030736  104113                          ERROR   113                     ;CYL ADDR IN RKMR3 NOT = RKDC
030740                          7$:
030740  104415                          SCOP1
030742  012706  001100                  MOV     #STACK,SP               ;RESTORE STK PTR
030746  004737  034214                  JSR     PC,SUBCLR
030752  104024                          ERROR   24                      ;CERR AFTER SCLR
                                                                        ;RETURN TO CYL 0
030754  012765  001500  000004          MOV     #DATA1,RKBA(R5)
030762  052765  000020  000010          BIS     #BAI,RKCS2(R5)
030770  012765  177400  000002          MOV     #-256.,RKWC(R5)
030776  012737  000023  007354          MOV     #WRDATA,HCS1
031004  004737  032262                  JSR     PC,DATCMD               ;DO WRITE DATA CMD & GET CONTR RDY
031010  104011                          ERROR   11                      ;NO RDY AFTER WRITE DATA CMD
031012  004737  033664                  JSR     PC,GSTAT                ;GET FRESH STATUS
031016  032737  020000  007404          BIT     #D.DROT,HMR3
031024  001401                          BEQ     8$
031026  104112                          ERROR   112                     ;DRIVE OFF TRACK AFTER WRITE DATA CMD
031030  032737  100000  007354  8$:     BIT     #CERR,HCS1
031036  001401                          BEQ     9$
031040  104012                          ERROR   12                      ;CERR AFTER WRITE DATA CMD
031042                          9$:
031042  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
031050  005037  007446                  CLR     E.B0                    ;EXPECTED MSG B0
031054  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
031062  012737  000001  007452          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
031070  005037  007454                  CLR     E.A2                    ;EXPECTED MSG A2
031074  012737  000002  007456          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
031102  012737  000003  007462          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
031110  004737  033042                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
031114  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
031116  104052                          ERROR   52                      ;MSG A0 ERROR AFTER WRITE DATA CMD
031120  104023                          ERROR   23                      ;MSH B0 ERROR
031122  104053                          ERROR   53                      ;MSG A1 ERROR
031124  104025                          ERROR   25                      ;MSG B1 ERROR
031126  005737  001360                  TST     CYLADD
031132  001401                          BEQ     10$
031134  104042                          ERROR   42                      ;NOT BACK TO CYL 0
```

```
031136  005200                    10$:    INC     R0
031140  020027  000144                    CMP     R0,#100.                ;ALL ITERATIONS DONE?
031144  001402                            BEQ     13$                     ;BR IF YES
031146  000137  030532                    JMP     3$                      ;ELSE DO AGAIN
031152  005000                    13$:    CLR     R0                      ;RESET ITERATION CTR
031154  006337  001346                    ASL     TOCYL
031160  023737  001346  015436            CMP     TOCYL,MC1               ;ALL MAJOR CYL DONE?
031166  001402                            BEQ     FORM                    ;BR IF YES
031170  000137  030532                    JMP     3$                      ;ELSE DO NEXT CYL
031174  005037  001176            FORM:   CLR     $ESCAPE
031200  005037  001346                    CLR     TOCYL                   ;RESTORE TO ORIG 22 SECTOR FORMAT.
031204  012737  100000  007422            MOV     #BIT15,TEMP5
031212  004737  034214            12$:    JSR     PC,SUBCLR
031216  104024                            ERROR   24                      ;CERR AFTER SCLR
031220  012765  001522  000004            MOV     #HDTAB,RKBA(R5) ;REFORMAT ALL MAJOR CYLINDERS
031226  012765  177676  000002            MOV     #-66.,RKWC(R5)
031234  013765  001346  000020            MOV     TOCYL,RKDC(R5)
031242  013737  001346  001362            MOV     TOCYL,CALADD            ;SETUP
031250  012737  000000  001460            MOV     #0,HEAD                 ;TO FILL
031256  012737  000000  001466            MOV     #0,FORMAT               ;HEADER
031264  004737  035212                    JSR     PC,FHDTAB               ;TABLE
031270  012737  000027  007354            MOV     #<WRHEAD>,HCS1
031276  004737  032262                    JSR     PC,DATCMD               ;DO DATA X FOR CMD & GET CONTR RDY
031302  104200                            ERROR   200                     ;NO RDY AFTER WRITE HEADER CMD
031304  004737  033664                    JSR     PC,GSTAT                ;GET FRESH STATUS
031310  032737  100000  007354            BIT     #CERR,HCS1
031316  001405                            BEQ     64$
031320  104201                            ERROR   201                     ;CERR AFTER WRITE HEADER CMD
031322  104401  045354                    TYPE    ,MSG26                  ;ABORTING BAL OF TESTS
031326  000137  031452                    JMP     $EOP
031332                            64$:
031332  006137  007422                    ROL     TEMP5
031336  006137  001346                    ROL     TOCYL
031342  023737  001346  015436            CMP     TOCYL,MC1               ;ALL MAJOR CYL REFORMATTED?
031350  001320                            BNE     12$                     ;BR IF NO
031352  005065  000020                    CLR     RKDC(R5)                ;SETUP TO RETURN TO CYL 0
031356  005037  001176                    CLR     $ESCAPE
031362  012737  000017  007354            MOV     #SEEK,HCS1
031370  004737  032224                    JSR     PC,DOCMD                ;DO SEEK CMD & GET CONTR READY
031374  104131                            ERROR   131                     ;NO RDY AFTER SEEK CMD
031376  013737  001420  007412            MOV     T50000,TEMP1            ;SETUP TIMEOUT
031404  004737  032730                    JSR     PC,FATT2                ;FIND ATTN
031410  104132                            ERROR   132                     ;NO ATTN AFTER SEEK CMD
```

```

031412  032737  100000  007354          BIT     #CERR,HCS1
031420  001401                          BEQ     65$
031422  104210                          ERROR   210             ;CERR AFTER SEEK CMD

031424                          65$:


031424  005737  007342                  TST     HPEND           ;SEE IF HALT PENDING
031430  001406                          BEQ     4$              ;BR IF NO
031432  005037  007342                  CLR     HPEND           ;CLEAR FOR FUTURE FORMATTING
031436  005037  007340                  CLR     BADHDR          ;HEADERS NOW OK
031442  000137  036262                  JMP     STOP            ;GO BACK & HALT CPU

031446  005037  007340          4$:     CLR     BADHDR          ;HEADERS NOW OK

```

```
                                        .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 STAPT

 031452                                 $EOP:

 031452  000004                                  SCOPE
 031454  005037  001176                          CLR     $ESCAPE
 031460  012737  000001  001174                  MOV     #1,$TIMES
 031466  012706  001100                          MOV     #STACK,SP
 031472  005237  001220                          INC     $DEVCT              ;INCR COUNT FOR # DRIVES CHECKED
 031476  023737  007474  001220                  CMP     DRIVS,$DEVCT        ;ARE ALL DRIVES PRESENT TESTED?
 031504  001403                                  BEQ     $EOP1+2             ;BR IF YES
 031506  000137  015146                          JMP     NUDRV               ;ELSE TEST NEXT DRIVE PRESENT
 031512  000004                         $EOP1:   SCOPE
 031514  005037  001102                          CLR     $TSTNM              ;;ZERO THE TEST NUMBER
 031520  005037  001174                          CLR     $TIMES              ;;ZERO THE NUMBER OF ITERATIONS
 031524  005237  001216                          INC     $PASS               ;;INCREMENT THE PASS NUMBER
 031530  042737  100000  001216                  BIC     #100000,$PASS       ;;DON'T ALLOW A NEG. NUMBER
 031536  005327                                  DEC     (PC)+               ;;LOOP?
 031540  000001                         $EOPCT:  .WORD   1
 031542  003022                                  BGT     $DOAGN              ;;YES
 031544  012737                                  MOV     (PC)+,@(PC)+        ;;RESTORE COUNTER
 031546  000001                         $ENDCT:  .WORD   1
 031550  031540                                  $EOPCT
 031552  104401  031617                          TYPE    ,$ENDMG             ;;TYPE "END PASS #"
 031556  013746  001216                          MOV     $PASS,-(SP)         ;;SAVE $PASS FOR TYPEOUT
 031562  104405                                  TYPDS                       ;;GO TYPE--DECIMAL ASCII WITH SIGN
 031564  104401  031614                          TYPE    ,$ENULL             ;;TYPE A NULL CHARACTER
 031570  013700  000042                 $GET42:  MOV     @#42,R0             ;;GET MONITOR ADDRESS
 031574  001405                                  BEQ     $DOAGN              ;;BRANCH IF NO MONITOR
 031576  000005                                  RESET                       ;;CLEAR THE WORLD
 031600  004710                         $ENDAD:  JSR     PC,(R0)             ;;GO TO MONITOR
 031602  000240                                  NOP                         ;;SAVE ROOM
 031604  000240                                  NOP                         ;;FOR
 031606  000240                                  NOP                         ;;ACT11
 031610                                 $DOAGN:
 031610  000137                                  JMP     @(PC)+              ;;RETURN
 031612  031634                         $RTNAD:  .WORD   STAPT
 031614     377     377     000         $ENULL:  .BYTE   -1,-1,0             ;;NULL CHARACTER STRING
 031617     015  042412  042116         $ENDMG:  .ASCIZ  <15><12>/END PASS #/
 031624  050040  051501  020123
 031632  000043

                                        ;ADD WAITING LOOP FOR THE APT MODE
 031634  122737  000001  001230         STAPT:   CMPB    #APTENV,$ENV
 031642  001007                                  BNE     2$
 031644  023727  001216  000002                  CMP     $PASS,#2            ;TWO PASS DONE ?,AS REQUIRED BY
 031652  103403                                  BLO     2$                  ;NOT YET
 031654  005237  001102                 1$:      INC     $TSTNM  ;INCREMENT THE TEST NUMBER
 031660  000775                                  BR      1$                  ;WAITING LOOP FOR APT TO DUMP NEXT PROG
 031662  000137  013356                 2$:      JMP     ST5                 ;RETURN TO MAIN LOOP
```

```
                                        .SBTTL  SUBROUTINES

                                        ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                        ;

031666  012700  007464                  CLRFLG: MOV     #DDUMP,R0
031672  012701  177757                          MOV     #-17.,R1
031676  005020                          1$:     CLR     (R0)+
031700  005201                                  INC     R1
031702  001375                                  BNE     1$
031704  000207                                  RTS     PC

                                        ;
                                        ;TYPE PROGRAM ID IF FTITLE=0
                                        ;

031706  005737  001340                  TITLE:  TST     FTITLE
031712  001024                                  BNE     1$
031714  005237  001340                          INC     FTITLE
031720  104401  043420                          TYPE    ,MSG1           ;PROGRAM ID
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
031724  005737  000042                          TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
031730  001012                                  BNE     64$             ;;BRANCH IF YES
031732  123727  001230  000001                  CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
031740  001406                                  BEQ     64$             ;;BRANCH IF YES
031742  023727  001140  000176                  CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
031750  001005                                  BNE     65$             ;;BRANCH IF NO
031752  104406                                  GTSWR                   ;;GET SOFT-SWR SETTINGS
031754  000403                                  BR      65$
031756  112737  000001  001134  64$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
031764                                  65$:
031764  000207                          1$:     RTS     PC

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

031766  104411                          GDRVS:  RDLIN
031770  012600                                  MOV     (SP)+,R0        ;GET STARTING ADDR OF ASCII STRING
031772  012701  177770                          MOV     #-8.,R1         ;SET UP COUNT
031776  112002                          1$:     MOVB    (R0)+,R2        ;GET ASCII CHAR
032000  042702  177400                          BIC     #177400,R2      ;MASK HI BYTE
032004  012703  007476                          MOV     #DRIV0,R3       ;DRIVE FLAG ADDR
032010  012704  000060                          MOV     #60,R4

032014  020402                          2$:     CMP     R4,R2           ;WAS TYPED CHAR 0 THRU 7?
032016  001415                                  BEQ     3$              ;BRANCH IF YES
032020  005723                                  TST     (R3)+           ;NO, INCREMENT DR FLAG ADDR
032022  005204                                  INC     R4
032024  020427  000070                          CMP     R4,#70
032030  001371                                  BNE     2$              ;S/B 0-7 OR TERMINATOR
032032  005702                                  TST     R2
032034  001022                                  BNE     4$
032036  020127  177770                          CMP     R1,#-8.
032042  001426                                  BEQ     6$              ;DEFAULT ALL DRIVES
```

```
032044  005037  007524           7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
032050  000207                           RTS     PC              ;FOUND TERMINATOR, EXIT

032052  005213                   3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
032054  005237  007474                   INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
032060  112002                           MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
032062  042702  177400                   BIC     #177400,R2      ;MASK
032066  022702  000054                   CMP     #54,R2          ;IS IT A COMMA?
032072  001407                           BEQ     5$              ;YES, GO TO NEXT WORD.
032074  005702                           TST     R2              ;NO, IS IT A TERMINATOR?
032076  001001                           BNE     4$              ;IF NOT, SOMETHING WRONG.
032100  000761                           BR      7$              ;FOUND TERMINATOR, EXIT

032102  104401  046137           4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
032106  000137  012562                   JMP     PRGSRT          ;START ALL OVER

032112  005201                   5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
032114  001330                           BNE     1$              ;DRIVES TYPED IN.
032116  000771                           BR      4$              ;IF MORE, HAVE ERROR.

032120  005237  007524           6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
032124  000207                           RTS     PC              ;EXIT.

                                 ;
                                 ;ROUTINE TO INPUT RKBAS OR DEFAULT.
                                 ;

032126  104412                   GBA:    RDOCT
032130  012600                           MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
032132  005700                           TST     R0
032134  001403                           BEQ     1$              ;BRANCH IF DEFAULT.
032136  010037  001264                   MOV     R0,$BASE
032142  000207                           RTS     PC
032144  012737  177440  001264   1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
032152  000207                           RTS     PC

                                 ;
                                 ;ROUTINE TO INPUT RKVEC OR DEFAULT
                                 ;

032154  104412                   GINT:   RDOCT
032156  012600                           MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
032160  005700                           TST     R0
032162  001405                           BEQ     1$              ;BRANCH IF DEFAULT
032164  010037  001314                   MOV     R0,RKVEC
032170  004737  032206           2$:     JSR     PC,SETINT
032174  000207                           RTS     PC
032176  012737  000210  001314   1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
032204  000771                           BR      2$

                                 ;
                                 ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
                                 ;

032206  013700  001314           SETINT: MOV     RKVEC,R0
032212  012720  036720                   MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
032216  013710  001316                          MOV     RKPRI,(R0)              ;PR5 TO RKVEC+2
032222  000207                                  RTS     PC


                                        ;
                                        ;THIS ROUTINE CDT IN RKCS1 IF DRIVE UNDER TEST IS AN RK07.
                                        ;ENTER WITH COMMAND IN HCS1
                                        ;
032224  053737  001170  007354          DOCMD:  BIS     $TMP4,HCS1              ;SET CDT IF RK07
032232  013765  007354  000000                  MOV     HCS1,RKCS1(R5)          ;DO COMMAND
032240  013737  001406  007412                  MOV     T10,TEMP1
032246  004737  032320                          JSR     PC,FRDY                 ;FIND CONTR READY
032252  000207                                  RTS     PC                      ;SET HERE IF NOT RDY
032254  062716  000002                          ADD     #2,(SP)                 ;ELSE SKIP OVER ERROR
032260  000207                                  RTS     PC
                                        ;
                                        ;THIS ROUTINE IS SIMILAR TO THE ABOVE BUT IS USED FOR DATA TRANSFERS
                                        ;& REQUIRES A LONGER TIMEOUT
                                        ;
032262  053737  001170  007354          DATCMD: BIS     $TMP4,HCS1              ;SET CDT IF RK07
032270  013765  007354  000000                  MOV     HCS1,RKCS1(R5)          ;DO CMD
032276  013737  001420  007412                  MOV     T50000,TEMP1
032304  004737  032320                          JSR     PC,FRDY                 ;FIND CONTR RDY
032310  000207                                  RTS     PC
032312  062716  000002                          ADD     #2,(SP)
032316  000207                                  RTS     PC

                                        ;
                                        ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
                                        ;ENTER WITH A COUNT IN TEMP1
                                        ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
                                        ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
                                        ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
                                        ;
032320  032765  000200  000000          FRDY:   BIT     #RDY,RKCS1(R5)
032326  001010                                  BNE     1$
032330  005337  007412                          DEC     TEMP1
032334  001371                                  BNE     FRDY
032336  004737  032454                          JSR     PC,HOLD                 ;STORE ALL RK611 REGS IN HOLDING REGS.
032342  004737  033602                          JSR     PC,CKCERR               ;CHECK FOR SPECIAL CERR
032346  000207                                  RTS     PC                      ;NO RDY, EXIT
032350  062716  000002                  1$:     ADD     #2,(SP)                 ;SKIP OVER ERROR
032354  004737  032454                          JSR     PC,HOLD
032360  004737  033602                          JSR     PC,CKCERR               ;CHECK FOR SPECIAL CERR
032364  000207                                  RTS     PC
                                        ;
                                        ;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
                                        ;
032366  032765  000200  000000          FRDY1:  BIT     #RDY,RKCS1(R5)
032374  001014                                  BNE     1$
032376  005337  007412                          DEC     TEMP1
032402  001371                                  BNE     FRDY1
032404  016537  000034  007402                  MOV     RKMR2(R5),HMR2
032412  016537  000036  007404                  MOV     RKMR3(R5),HMR3
032420  004737  033602                          JSR     PC,CKCERR               ;CHECK FOR SPECIAL CERR CONDITIONS
032424  000207                                  RTS     PC                      ;NO RDY, EXIT
032426  062716  000002                  1$:     ADD     #2,(SP)                 ;SKIP OVER ERROR
```

```
032432  016537  000034  007402          MOV     RKMR2(R5),HMR2
032440  016537  000036  007404          MOV     RKMR3(R5),HMR3
032446  004737  033602                  JSR     PC,CKCERR               ;CHECK FOR SPECIAL CERR CONDITIONS
032452  000207                          RTS     PC


                                ;
                                ;STORE ALL RK611 REGISTERS IN HOLDING REGS
                                ;

032454  016537  000000  007354  HOLD:   MOV     RKCS1(R5),HCS1
032462  016537  000010  007356          MOV     RKCS2(R5),HCS2
032470  016537  000002  007360          MOV     RKWC(R5),HWC
032476  016537  000004  007362          MOV     RKBA(R5),HBA
032504  016537  000006  007364          MOV     RKDA(R5),HDA
032512  016537  000012  007366          MOV     RKDS(R5),HDS
032520  016537  000014  007370          MOV     RKER(R5),HER
032526  016537  000016  007372          MOV     RKASOF(R5),HASOF
032534  016537  000020  007374          MOV     RKDC(R5),HDC
032542  016537  000026  007400          MOV     RKMR1(R5),HMR1
032550  016537  000034  007402          MOV     RKMR2(R5),HMR2
032556  016537  000036  007404          MOV     RKMR3(R5),HMR3
032564  016537  000030  007406          MOV     RKECPS(R5),HPOS
032572  016537  000032  007410          MOV     RKECPT(R5),HPAT
032600  000207                          RTS     PC


                                ;
                                ;ROUTINE TO CHECK FOR CORRECT ATTN
                                ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
                                ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
                                ;
032602  010446                  TSTATN: MOV     R4,-(SP)                ;SAV R4
032604  013704  001222                  MOV     $UNIT,R4
032610  136437  007344  007373          BITB    ATTN(R4),HASOF+1
032616  001404                          BEQ     1$                      ;BRANCH IF ATTN NOT PRESENT
032620  012604                          MOV     (SP)+,R4                ;RESTOR R4
032622  062716  000002                  ADD     #2,(SP)                 ;INCR RET ADDR TO JUMP OVER ERROR.
032626  000207                          RTS     PC
032630  012604                  1$:     MOV     (SP)+,R4                ;RESTOR R4
032632  000207                          RTS     PC


                                ;
                                ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
                                ;ENTER WITH TIME IN SECONDS IN TEMP2
                                ;RETURN IF NO ATTN (ERROR CONDITION)
                                ;RETURN +2 IF ATTN FOUND
                                ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
                                ;

032634  010446                  FATT1:  MOV     R4,-(SP)                ;SAV R4
032636  012737  177777  007412  3$:     MOV     #-1,TEMP1
032644  013704  001222                  MOV     $UNIT,R4
032650  136465  007344  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5)   ;FIND CORRECT ATTN
032656  001014                          BNE     2$
032660  005337  007412                  DEC     TEMP1
032664  001371                          BNE     1$
032666  005337  007414                  DEC     TEMP2
```

```
032672  001361                          BNE     3$
032674  005065  000026                  CLR     RKMR1(R5)               ;SELECT WORD 0
032700  004737  033664                  JSR     PC,GSTAT                ;GET LATEST STATUS
032704  012604                          MOV     (SP)+,R4                ;RESTOR R4
032706  000207                          RTS     PC
032710  005065  000026          2$:     CLR     RKMR1(R5)
032714  004737  033664                  JSR     PC,GSTAT                ;GET STATUS AFTER ATTN SEEN
032720  012604                          MOV     (SP)+,R4                ;RESTOR R4
032722  062716  000002                  ADD     #2,(SP)                 ;SKIP OVER ERROR
032726  000207                          RTS     PC

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

032730  010446                  FATT2:  MOV     R4,-(SP)                ;SAV R4
032732  013704  001222          2$:     MOV     $UNIT,R4
032736  136465  007344  000017          BITB    ATTN(R4),RKASOF+1(R5)   ;FIND CORRECT ATTN
032744  001011                          BNE     1$
032746  005337  007412                  DEC     TEMP1
032752  001367                          BNE     2$
032754  005065  000026                  CLR     RKMR1(R5)               ;SELECT WORD 0
032760  004737  033664                  JSR     PC,GSTAT                ;GET LATEST STATUS.
032764  012604                          MOV     (SP)+,R4                ;RESTOR R4
032766  000207                          RTS     PC
032770  005065  000026          1$:     CLR     RKMR1(R5)
032774  004737  033664                  JSR     PC,GSTAT
033000  012604                          MOV     (SP)+,R4                ;RESTOR R4
033002  062716  000002                  ADD     #2,(SP)                 ;SKIP OVER ERROR
033006  000207                          RTS     PC

                                ;ENTER WITH A COUNT IN TEMP1
                                ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
                                ;WHEN COUNT IS 0. BASED ON AN 11/05
                                ;
033010  005737  007412          DLY:    TST     TEMP1                   ;5.6 US
033014  001403                          BEQ     1$                      ;1.9 US
033016  005337  007412                  DEC     TEMP1                   ;6.8 US
033022  000772                          BR      DLY                     ;2.5 US
033024  000207                  1$:     RTS     PC                      ;3.8 US

                                ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
                                ;

033026  104401  045243          BYP:    TYPE    ,MSG14                  ;BYPASS DRIVE
033032  010046                          MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
                                                                        ;;TYPE DR#
033034  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
033036     001                          .BYTE   1                       ;;TYPE 1 DIGIT(S)
033037     000                          .BYTE   0                       ;;SUPPRESS LEADING ZEROS
033040  000207                          RTS     PC
                                ;
```

```
                                                ;THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
                                                ;
033042  017637  000000  001520  CHKMSG: MOV     @(SP),CHKFLG            ;PASS MSGS TO BE TESTED
033050  062716  000002                  ADD     #2,(SP)                 ;BUMP RETURN ADDR TO 1ST ERROR
033054  004737  033720                  JSR     PC,GSTAT1               ;GET ALL ACTUAL DRIVE & CONTR STATUS

033060  053737  001222  007444          BIS     $UNIT,E.A0              ;SET UNIT #
033066  053737  001222  007450          BIS     $UNIT,E.A1
033074  053737  001222  007454          BIS     $UNIT,E.A2
033102  053737  001222  007460          BIS     $UNIT,E.A3
033110  053737  015444  007444          BIS     E.DDT,E.A0              ;ADD DRIVE TYPE

033116  013746  007412                  MOV     TEMP1,-(SP)             ;SAVE TEMP1

033122  013737  007444  007412          MOV     E.A0,TEMP1
033130  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG A0
033134  013737  007412  007444          MOV     TEMP1,E.A0

033142  013737  007450  007412          MOV     E.A1,TEMP1
033150  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG A1
033154  013737  007412  007450          MOV     TEMP1,E.A1

033162  013737  007454  007412          MOV     E.A2,TEMP1
033170  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG A2
033174  013737  007412  007454          MOV     TEMP1,E.A2

033202  013737  007446  007412          MOV     E.B0,TEMP1
033210  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG B0
033214  013737  007412  007446          MOV     TEMP1,E.B0

033222  013737  007452  007412          MOV     E.B1,TEMP1
033230  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG B1
033234  013737  007412  007452          MOV     TEMP1,E.B1

033242  013737  007456  007412          MOV     E.B2,TEMP1
033250  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG B2
033254  013737  007412  007456          MOV     TEMP1,E.B2

033262  013737  007462  007412          MOV     E.B3,TEMP1
033270  004737  036156                  JSR     PC,SBPAR                ;GET PARITY FOR MSG B3
033274  013737  007412  007462          MOV     TEMP1,E.B3

033302  012637  007412                  MOV     (SP)+,TEMP1             ;RESTORE TEMP1
033306  013737  001176  001172          MOV     $ESCAPE,$TMP5           ;SAVE ESCAPE

033314  023737  007424  007444          CMP     H.A0,E.A0               ;TEST MSG A0
033322  001411                          BEQ     2$                      ;BR IF OK
033324  012737  033336  001176          MOV     #1$,$ESCAPE             ;ELSE SETUP ESCAPE
033332  011646                          MOV     (SP),-(SP)              ;COPY RET ADDR.
033334  000207                          RTS     PC                      ;& RETURN TO MAINLINE ERROR

033336  032777  001000  145574  1$:     BIT     #SW9,@SWR               ;RET HERE FROM MAINLINE ERROR
033344  001107                          BNE     20$                     ;& BR IF LOOP ON ERROR
033346  062716  000002          2$:     ADD     #2,(SP)                 ;BUMP RET ADDR TO NEXT ERROR

033352  023737  007426  007446          CMP     H.B0,E.B0               ;TEST MSG B0
```

```
033360  001411                                BEQ     5$                      ;BR IF OK
033362  012737  033374  001176                MOV     #4$,$ESCAPE             ;ELSE SETUP ESCAPE
033370  011646                                MOV     (SP),-(SP)              ;COPY RET ADDR
033372  000207                                RTS     PC                      ;& RETURN TO MAINLINE ERROR

033374  032777  001000  145536  4$:           BIT     #SW9,@SWR               ;RETURN HERE FROM MAINLINE ERROR
033402  001070                                BNE     20$                     ;& BR IF LOOP ON ERROR
033404  062716  000002          5$:           ADD     #2,(SP)                 ;BUMP RET ADDR TO NEXT ERROR

033410  023737  007430  007450                CMP     H.A1,E.A1               ;TEST MSG A1
033416  001411                                BEQ     8$                      ;BR IF OK
033420  012737  033432  001176                MOV     #7$,$ESCAPE
033426  011646                                MOV     (SP),-(SP)
033430  000207                                RTS     PC

033432  032777  001000  145500  7$:           BIT     #SW9,@SWR
033440  001051                                BNE     20$
033442  062716  000002          8$:           ADD     #2,(SP)

033446  023737  007432  007452                CMP     H.B1,E.B1               ;TEST MSG B1
033454  001411                                BEQ     11$                     ;BR IF OK
033456  012737  033470  001176                MOV     #10$,$ESCAPE
033464  011646                                MOV     (SP),-(SP)
033466  000207                                RTS     PC

033470  032777  001000  145442  10$:          BIT     #SW9,@SWR
033476  001032                                BNE     20$
033500  062716  000002          11$:          ADD     #2,(SP)

033504  032737  000001  001520  12$:          BIT     #T.A2,CHKFLG            ;TEST MSG A2?
033512  001402                                BEQ     13$                     ;BR IF NO
033514  004737  034604                        JSR     PC,RCYLD                ;PUT INFO CYLDIF, DO NOT CHECK
033520  032737  000002  001520  13$:          BIT     #T.B2,CHKFLG            ;TEST MSG B2?
033526  001402                                BEQ     14$                     ;BR IF NO
033530  004737  034656                        JSR     PC,RCYLA                ;PUT INFO IN CYLADD, DO NOT CHECK

033534  032737  000004  001520  14$:          BIT     #T.B3,CHKFLG            ;TEST MSG B3?
033542  001404                                BEQ     15$
033544  004737  034714                        JSR     PC,RSEC                 ;PUT INFO IN SECTOR, DO NOT CHECK
033550  004737  034752                        JSR     PC,RHEAD                ;PUT INFO IN HEADA, DO NOT CHECK

033554  013737  001172  001176  15$:          MOV     $TMP5,$ESCAPE           ;RESTORE ESCAPE
033562  000207                                RTS     PC

033564  012706  001100          20$:          MOV     #STACK,SP               ;RESET STACK PTR
033570  013737  001172  001176                MOV     $TMP5,$ESCAPE           ;RESTORE ESCAPE
033576  000177  145306                        JMP     @$LPERR

                                ;
                                ;THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
                                ;I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
                                ;
033602  005737  001516          CKCERR:       TST     BYPCERR
033606  001025                                BNE     4$
033610  032737  100000  007354                BIT     #CERR,HCS1
033616  001001                                BNE     1$                      ;BR IF CERR
033620  000207                                RTS     PC
```

```
033622  032737  004000  007354  1$:     BIT     #CTO,HCS1
033630  001402                          BEQ     2$                      ;BR IF NOT CTO
033632  104125                          ERROR   125                     ;CTO ERROR, MSG A & B INVALID
033634  000207                          RTS     PC
033636  032737  010000  007356  2$:     BIT     #NED,HCS2
033644  001401                          BEQ     3$                      ;BR IF NOT NED
033646  104126                          ERROR   126                     ;NED ERROR, MSG A & B INVALID
033650  032737  001000  007356  3$:     BIT     #MDS,HCS2
033656  001401                          BEQ     4$
033660  104127                          ERROR   127                     ;MDS ERROR, MSG A & B INVALID
033662  000207                  4$:     RTS     PC
                                ;
                                ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
                                ;IT THEN WAITS FOR CONTROLLER READY
                                ;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
                                ;
033664  013746  007412          GSTAT:  MOV     TEMP1,-(SP)             ;SAVE TEMP1
033670  013765  001222  000010          MOV     $UNIT,RKCS2(R5)         ;CURRENT DRIVE #
033676  012737  000001  007354          MOV     #SELDRV,HCS1
033704  004737  032224                  JSR     PC,DOCMD                ;DO SELDRV (STATUS) CMD & GET CONTR RDY
033710  104117                          ERROR   117                     ;RDY NOT SET BY END OF SELECT DRIVE CMD
033712  012637  007412                  MOV     (SP)+,TEMP1             ;RESTOR TEMP1.
033716  000207                          RTS     PC
                                ;
                                ;THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
                                ;& ALL CONTROLLER REGISTERS.
                                ;
033720  013746  007412          GSTAT1: MOV     TEMP1,-(SP)             ;SAVE TEMP1
033724  004737  032454                  JSR     PC,HOLD                 ;GET ALL CONTR REG
033730  012765  100000  000000          MOV     #CCLR,RKCS1(R5)         ;CLEAR CONTR
033736  013765  001222  000010          MOV     $UNIT,RKCS2(R5)         ;CURRENT DRIVE #
033744  012765  000003  000026          MOV     #3,RKMR1(R5)            ;SELECT WORD 3
033752  004737  034150                  JSR     PC,GSTAT2
033756  104117                          ERROR   117                     ;RDY NOT SET BY END OF SELECT DRV CMD
033760  013737  007402  007440          MOV     HMR2,H.A3               ;STORE MSG A3
033766  013737  007404  007442          MOV     HMR3,H.B3               ;STORE MSG B3
033774  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
034002  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
034010  012765  000002  000026          MOV     #2,RKMR1(R5)            ;SELECT WORD 2
034016  004737  034150                  JSR     PC,GSTAT2
034022  104117                          ERROR   117                     ;RDY NOT SET BY END OF SELECT DRV CMD
034024  013737  007402  007434          MOV     HMR2,H.A2               ;STORE MSG A2
034032  013737  007404  007436          MOV     HMR3,H.B2               ;STORE MSG B2
034040  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
034046  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
034054  012765  000001  000026          MOV     #1,RKMR1(R5)            ;SELECT WORD 1
034062  004737  034150                  JSR     PC,GSTAT2
034066  104117                          ERROR   117                     ;RDY NOT SET BY END OF SELECT DRV CMD
```

```
034070  013737  007402  007430            MOV     HMR2,H.A1          ;STORE MSG A1
034076  013737  007404  007432            MOV     HMR3,H.B1          ;STORE MSG B1

034104  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
034112  013765  001222  000010            MOV     $UNIT,RKCS2(R5)
034120  004737  034150                    JSR     PC,GSTAT2
034124  104117                            ERROR   117                ;RDY NOT SET BY END OF SEL DRV CMD
034126  013737  007402  007424            MOV     HMR2,H.A0          ;STORE MSG A0
034134  013737  007404  007426            MOV     HMR3,H.B0          ;STORE MSG B0

034142  012637  007412                    MOV     (SP)+,TEMP1        ;RESTORE TEMP1
034146  000207                            RTS     PC

034150  012737  000001  007354    GSTAT2: MOV     #SELDRV,HCS1
034156  053737  001170  007354            BIS     $TMP4,HCS1         ;RET CDT IF RK07
034164  013765  007354  000000            MOV     HCS1,RKCS1(R5)     ;GET STATUS
034172  013737  001406  007412            MOV     T10,TEMP1
034200  004737  032366                    JSR     PC,FRDY1           ;FIND CONTR RDY & STORE DRIVE REGS ONLY
034204  000207                            RTS     PC                 ;RET HERE IF NOT RDY
034206  062716  000002                    ADD     #2,(SP)            ;RET HERE IF OK
034212  000207                            RTS     PC


                                  ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
                                  ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
                                  ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
                                  ;RETURN IF CERR SET
                                  ;RETURN +2 IF CERR CLEAR

034214  012765  000040  000010    SUBCLR: MOV     #SCLR,RKCS2(R5)    ;SUBSYS CLEAR
034222  013737  001406  007412            MOV     T10,TEMP1
034230  004737  032320                    JSR     PC,FRDY            ;FIND RDY
034234  104120                            ERROR   120                ;RDY NOT SET BY END OF SCLR
034236  013765  001222  000010            MOV     $UNIT,RKCS2(R5)    ;CURRENT DRIVE #
034244  005065  000026                    CLR     RKMR1(R5)          ;SELECT WORD 0
034250  004737  033664                    JSR     PC,GSTAT           ;GET STATUS
034254  032737  100000  007354            BIT     #CERR,HCS1         ;CHECK FOR CONT ERROR
034262  001401                            BEQ     1$
034264  000207                            RTS     PC
034266  062716  000002            1$:     ADD     #2,(SP)            ;SKIP OVER ERROR
034272  000207                            RTS     PC


                                  ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'

034274  012765  000003  000026    RDSEC:  MOV     #3,RKMR1(R5)       ;WORD 3
034302  004737  033664                    JSR     PC,GSTAT
034306  013737  007404  001402            MOV     HMR3,SECTOR
034314  042737  177017  001402            BIC     #^C<M.SECT>,SECTOR
034322  006237  001402                    ASR     SECTOR             ;RIGHT JUSTIFY
034326  006237  001402                    ASR     SECTOR             ;SECTOR
034332  006237  001402                    ASR     SECTOR             ;INFO
034336  006237  001402                    ASR     SECTOR
034342  000207                            RTS     PC

                                  ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
```

```
                                        ;
034344  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
034352  004737  033664                  JSR     PC,GSTAT
034356  013737  007402  001356          MOV     HMR2,CYLDIF
034364  043737  015442  001356          BIC     MASK1,CYLDIF
034372  006237  001356                  ASR     CYLDIF          ;RIGHT JUSTIFY
034376  006237  001356                  ASR     CYLDIF          ;CYL DIFF/OFFSET
034402  006237  001356                  ASR     CYLDIF          ;INFO
034406  006237  001356                  ASR     CYLDIF
034412  023737  001356  015440          CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
034420  001002                          BNE     1$              ;BR IF NOT
034422  005037  001356                  CLR     CYLDIF          ;CLR IF YES
034426  000207                  1$:     RTS     PC

                                        ;
                                        ;QUICK SELECT DRIVE COMMAND TO OBTAIN CYL DIFF
                                        ;
034430  013746  007412          QKCYLD: MOV     TEMP1,-(SP)     ;SAVE TEMP1
034434  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
034442  012737  000001  007354          MOV     #SELDRV,HCS1    ;SELECT DRIVE CMD
034450  053737  001170  007354          BIS     $TMP4,HCS1
034456  013765  007354  000000          MOV     HCS1,RKCS1(R5)
034464  013737  001406  007412          MOV     T10,TEMP1
034472  032765  000200  000000  1$:     BIT     #RDY,RKCS1(R5)  ;TEST FOR CONT RDY
034500  001004                          BNE     2$              ;BR IF THERE
034502  005337  007412                  DEC     TEMP1
034506  001371                          BNE     1$
034510  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD

034512  016537  000034  001356  2$:     MOV     RKMR2(R5),CYLDIF
034520  043737  015442  001356          BIC     MASK1,CYLDIF    ;GET CYL DIFF ONLY (NO SHIFTING)
034526  012637  007412                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
034532  000207                          RTS     PC
                                        ;
                                        ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
                                        ;
034534  012765  000002  000026  RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
034542  004737  033664                  JSR     PC,GSTAT
034546  013737  007404  001360          MOV     HMR3,CYLADD
034554  043737  015442  001360          BIC     MASK1,CYLADD
034562  006237  001360                  ASR     CYLADD          ;RIGHT JUSTIFY
034566  006237  001360                  ASR     CYLADD          ;CYL ADDR
034572  006237  001360                  ASR     CYLADD          ;INFO
034576  006237  001360                  ASR     CYLADD
034602  000207                          RTS     PC
                                        ;
                                        ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                        ;
034604  013737  007434  001356  RCYLD:  MOV     H.A2,CYLDIF
034612  043737  015442  001356          BIC     MASK1,CYLDIF  ;CLEAR UNWANTED INFO
034620  006237  001356                  ASR     CYLDIF   ;RIGHT JUSTIFY
034624  006237  001356                  ASR     CYLDIF
034630  006237  001356                  ASR     CYLDIF
034634  006237  001356                  ASR     CYLDIF
034640  023737  001356  015440          CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
034646  001002                          BNE     1$              ;BR IF NO
```

```
034650  005037  001356                    CLR     CYLDIF            ;ELSE CLEAR
034654  000207                    1$:     RTS     PC
                                  ;
                                  ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
                                  ;
034656  013737  007436  001360    RCYLA:  MOV     H.B2,CYLADD
034664  043737  015442  001360            BIC     MASK1,CYLADD   ;CLEAR UNWANTED INFO
034672  006237  001360                    ASR     CYLADD            ;RIGHT JUSTIFY
034676  006237  001360                    ASR     CYLADD
034702  006237  001360                    ASR     CYLADD
034706  006237  001360                    ASR     CYLADD
034712  000207                            RTS     PC
                                  ;
                                  ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                  ;
034714  013737  007442  001402    RSEC:   MOV     H.B3,SECTOR
034722  042737  177017  001402            BIC     #↑C<M.SECT>,SECTOR   ;CLEAR UNWANTED INFO
034730  006237  001402                    ASR     SECTOR            ;RIGHT JUSTIFY
034734  006237  001402                    ASR     SECTOR
034740  006237  001402                    ASR     SECTOR
034744  006237  001402                    ASR     SECTOR
034750  000207                            RTS     PC
                                  ;
                                  ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
                                  ;
034752  013737  007442  001462    RHEAD:  MOV     H.B3,HEADA
034760  042737  170777  001462            BIC     #↑C<M.HEAD>,HEADA   ;CLEAR UNWANTED INFO
034766  006237  001462                    ASR     HEADA             ;RIGHT JUSTIFY IT
034772  000337  001462                    SWAB    HEADA
034776  000207                            RTS     PC
                                  ;
                                  ;FIND SECTOR 23
                                  ;RETURN IF NOT FOUND
                                  ;RETURN +4 IF FOUND
                                  ;
035000  013737  001416  007412    FSEC23: MOV     T5000,TEMP1       ;SETUP TIMEOUT
035006  004737  034274            1$:     JSR     PC,RDSEC          ;READ SECTOR
035012  023727  001402  000023            CMP     SECTOR,#23        ;TEST FOR SECTOR 23(8)
035020  001014                            BNE     2$                ;BR IF NOT 23(8)

035022  004737  034274                    JSR     PC,RDSEC
035026  023727  001402  000023            CMP     SECTOR,#23
035034  001412                            BEQ     3$                ;BR IF READ SAME TWICE
035036  004737  034274                    JSR     PC,RDSEC          ;ELSE TRY 1 MORE TIME
035042  023727  001402  000023            CMP     SECTOR,#23
035050  001404                            BEQ     3$                ;BR IF 23(8)

035052  005337  007412            2$:     DEC     TEMP1
035056  001353                            BNE     1$                ;TRY AGAIN
035060  000207                            RTS     PC

035062  062716  000004            3$:     ADD     #4,(SP)           ;SKIP OVER ERROR
035066  000207                            RTS     PC
                                  ;
                                  ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
```

```
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND - SKIP OVER ERROR
                                        ;
035070  012737  177777  007412  FHDHM:  MOV     #-1,TEMP1               ;ALL 1'S
035076  012765  000001  000026          MOV     #1,RKMR1(R5)            ;WORD 1
035104  004737  033664          1$:     JSR     PC,GSTAT
035110  032737  000040  007402          BIT     #D.HDHM,HMR2
035116  001007                          BNE     2$
035120  005337  007412                  DEC     TEMP1
035124  001367                          BNE     1$
035126  005337  007414                  DEC     TEMP2
035132  001356                          BNE     FHDHM
035134  000207                          RTS     PC
035136  062716  000002          2$:     ADD     #2,(SP)                 ;SKIP OVER ERROR
035142  000207                          RTS     PC

                                        ;
                                        ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND: SKIP OVER ERROR
                                        ;
035144  012737  000372  007412  FLOAD:  MOV     #250.,TEMP1
035152  012765  000001  000026          MOV     #1,RKMR1(R5)            ;WORD 1
035160  004737  033664          1$:     JSR     PC,GSTAT
035164  032737  010000  007402          BIT     #D.LOAD,HMR2
035172  001004                          BNE     2$
035174  005337  007412                  DEC     TEMP1
035200  001367                          BNE     1$
035202  000207                          RTS     PC
035204  062716  000002          2$:     ADD     #2,(SP)                 ;SKIP OVER ERROR
035210  000207                          RTS     PC
                                        ;
                                        ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
                                        ;ENTER WITH CYL # IN 'CALADD'
                                        ;ENTER WITH HEAD # IN 'HEAD'
                                        ;ENTER WITH FORMAT IN 'FORMAT'
                                        ;
035212  010046                  FHDTAB: MOV     R0,-(SP)                ;SAV R0
035214  010146                          MOV     R1,-(SP)                ;SAV R1
035216  012700  001522                  MOV     #HDTAB,R0               ;HEADER WORD TABLE ADDR
035222  005001                          CLR     R1                      ;SECTOR COUNTER
035224  013737  001460  001464          MOV     HEAD,HD1
035232  006337  001464                  ASL     HD1
035236  006337  001464                  ASL     HD1
035242  006337  001464                  ASL     HD1
035246  006337  001464                  ASL     HD1
035252  006337  001464                  ASL     HD1                     ;SETUP HEAD # FOR WORD 2 OF HEADER
035256  013737  001466  001470          MOV     FORMAT,FMT1
035264  000337  001470                  SWAB    FMT1
035270  006337  001470                  ASL     FMT1                    ;SETUP FORMAT FOR WORD 2 OF HEADER

035274  013720  001362          1$:     MOV     CALADD,(R0)+            ;HEADER WORD 1-CYL ADDR
035300  010110                          MOV     R1,(R0)                 ;HEADER WORD 2-SECTOR NO
035302  053710  001464                  BIS     HD1,(R0)                ;               -HEAD NO
035306  053710  001470                  BIS     FMT1,(R0)               ;               -FORMAT
035312  004737  035372                  JSR     PC,SECFLG
```

```
035316  013737  001362  007412          MOV     CALADD,TEMP1
035324  011037  007414                  MOV     (R0),TEMP2
035330  043737  001362  007414          BIC     CALADD,TEMP2
035336  042037  007412                  BIC     (R0)+,TEMP1
035342  053737  007412  007414          BIS     TEMP1,TEMP2
035350  013720  007414                  MOV     TEMP2,(R0)+             ;HEADER WORD 3-HEADER CHECK

035354  005201                          INC     R1                      ;SECTOR CTR
035356  020127  000026                  CMP     R1,#22.                 ;ALL 22 SECTORS DONE? (66 WORDS)
035362  001344                          BNE     1$                      ;BR IF NO

035364  012601                          MOV     (SP)+,R1                ;RESTOR R1
035366  012600                          MOV     (SP)+,R0                ;RESTOR R0
035370  000207                          RTS     PC

                                ;
                                ;THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
                                ;TEST & SETS BITS 14 & 15 APPROPRIATLY.
                                ;
035372  010246                  SECFLG: MOV     R2,-(SP)                ;SAVE R2
035374  005737  001466                  TST     FORMAT
035400  001016                          BNE     1$                      ;BR IF 20 SECTOR FORMAT
035402  012702  003346                  MOV     #BSE22H+8.,R2
035406  004737  035472                  JSR     PC,FLGTST               ;GET HARDWARE DETECTED FLAG
035412  052710  100000                  BIS     #BIT15,(R0)             ;RETURN HERE IF GOOD SECTOR

035416  012702  005346                  MOV     #BSE22S+8.,R2           ;ELSE RETURN HERE
035422  004737  035472                  JSR     PC,FLGTST               ;GET SOFTWARE DETECTED FLAG
035426  052710  040000                  BIS     #BIT14,(R0)             ;RETURN HERE IF GOOD SECTOR

035432  012602                          MOV     (SP)+,R2                ;ELSE RETURN HERE
035434  000207                          RTS     PC

035436  012702  002346          1$:     MOV     #BSE20H+8.,R2
035442  004737  035472                  JSR     PC,FLGTST               ;GET HARDWARE DETECTED FLAG
035446  052710  100000                  BIS     #BIT15,(R0)             ;RETURN HERE IF GOOD SECTOR

035452  012702  004346                  MOV     #BSE20S+8.,R2
035456  004737  035472                  JSR     PC,FLGTST               ;GET SOFTWARE DETECTED FLAG
035462  052710  040000                  BIS     #BIT14,(R0)             ;RETURN HERE IF GOOD SECTOR

035466  012602                          MOV     (SP)+,R2                ;RESTORE R2
035470  000207                          RTS     PC


                                ;
                                ;THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
                                ;ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
                                ;RETURN IF NO COMPARE
                                ;RETURN+4 IF COMPARE
                                ;
035472  010346                  FLGTST: MOV     R3,-(SP)                ;SAVE R3

035474  021227  177777          1$:     CMP     (R2),#-1                ;SEE IF ALL 1'S
035500  001002                          BNE     2$                      ;BR IF NO
035502  012603                          MOV     (SP)+,R3                ;RESTORE R3
035504  000207                          RTS     PC
```

```
035506  022237  001362          2$:     CMP     (R2)+,CALADD            ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
035512  001403                          BEQ     3$
035514  062702  000002                  ADD     #2,R2                   ;GO TO NEXT CYL WORD IN TABLE
035520  000765                          BR      1$

035522  013703  001460          3$:     MOV     HEAD,R3                 ;GET HEAD # FROM FHDTAB ROUTINE
035526  000303                          SWAB    R3
035530  050103                          BIS     R1,R3                   ;ADD SECTOR # FROM FHDTAB ROUTINE
035532  022203                          CMP     (R2)+,R3                ;SEE IF SECTOR/HEAD COMPARE
                                                                        ;& INCR PTR TO NEXT CYL WORD
035534  001401                          BEQ     4$                      ;BR IF COMPARE
035536  000756                          BR      1$                      ;ELSE TRY NEXT CYL

035540  012603                  4$:     MOV     (SP)+,R3                ;RESTORE R3
035542  062716  000004                  ADD     #4,(SP)                 ;INCREMENT RET ADDR
035546  000207                          RTS     PC


                                ;
                                ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
                                ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
                                ;
035550  010046                  SORT:   MOV     R0,-(SP)                ;SAVE R0
035552  010146                          MOV     R1,-(SP)                ;SAVE R1
035554  004737  034274                  JSR     PC,RDSEC
035560  062737  000001  001402          ADD     #1,SECTOR
035566  004737  035656                  JSR     PC,MULT6                ;MULT SECTOR BY 6

035572  012700  000204                  MOV     #132.,R0
035576  163700  001402                  SUB     SECTOR,R0               ;R0-SECTOR TO R0 = INDEX
035602  010037  001402                  MOV     R0,SECTOR
035606  062737  001726  001402          ADD     #RHTAB,SECTOR           ;SAVE INDEX

035614  062700  001726                  ADD     #RHTAB,R0               ;INDEX TO BOT HALF OF RHTAB
035620  012701  002132                  MOV     #SRTTAB,R1              ;INDEX TO TOP HALF OF SRTTAB

035624  012021                  1$:     MOV     (R0)+,(R1)+             ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
035626  020027  002132                  CMP     R0,#RHTAB+132.
035632  001374                          BNE     1$

035634  012700  001726                  MOV     #RHTAB,R0               ;PUT TOP OF RHTAB TO BOT OF SRTTAB
035640  012021                  2$:     MOV     (R0)+,(R1)+
035642  020037  001402                  CMP     R0,SECTOR
035646  001374                          BNE     2$

035650  012601                          MOV     (SP)+,R1                ;RESTOR R1
035652  012600                          MOV     (SP)+,R0                ;RESTOR R0
035654  000207                          RTS     PC

                                ;
                                ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
                                ;
035656  006337  001402          MULT6:  ASL     SECTOR                  ;2 X SECTOR
035662  013746  001402                  MOV     SECTOR,-(SP)
035666  006337  001402                  ASL     SECTOR                  ;4 X SECTOR
035672  062637  001402                  ADD     (SP)+,SECTOR            ;(4 X S)+(2 X S) = 6 X SECTOR
035676  000207                          RTS     PC
```

```
                                                ;
                                                ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
                                                ;CMD.  IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
                                                ;CYLINDER AT CYL 410, TRACK 2.
                                                ;
                                                ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
                                                ;RETURN+2 IF LISTED, SKIP OVER ERROR
                                                ;
035700  010446                                  TRUERR: MOV     R4,-(SP)                ;SAVE R4

035702  032737  010000  007354                          BIT     #CFMT,HCS1      ;CHECK FORMAT
035710  001014                                          BNE     2$              ;BR FOR 20 SECTOR FORMAT

035712  012704  003346                                  MOV     #BSE22H+8.,R4
035716  004737  036000                                  JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
035722  000422                                          BR      3$              ;RETURN HERE IF YES

035724  012704  005346                                  MOV     #BSE22S+8.,R4   ;ELSE RETURN HERE
035730  004737  036000                                  JSR     PC,TERR1        ;& SEE IF ON SOFTWARE DETECTED TABLE
035734  000415                                          BR      3$              ;RETURN HERE IF YES

035736  012604                                  1$:     MOV     (SP)+,R4        ;RESTORE R4
035740  000207                                          RTS     PC              ;RETURN WITHOUT JUMPING OVER ERROR

035742  012704  002346                          2$:     MOV     #BSE20H+8.,R4
035746  004737  036000                                  JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
035752  000406                                          BR      3$              ;RETURN HERE IF YES

035754  012704  004346                                  MOV     #BSE20S+8.,R4   ;ELSE RETURN HERE
035760  004737  036000                                  JSR     PC,TERR1        ;SEE IF ON SOFTWARE DETECTED TABLE
035764  000401                                          BR      3$              ;RETURN HERE IF YES
035766  000763                                          BR      1$              ;RETURN HERE IF NO

035770  012604                                  3$:     MOV     (SP)+,R4        ;RESTORE R4
035772  062716  000002                                  ADD     #2,(SP)         ;SKIP OVER ERROR ON RETURN
035776  000207                                          RTS     PC
```

```
6805                                      ;
6806                                      ;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
6807                                      ;THE BSE TABLE FOR THE ABOVE SUBROUTINE.
6808                                      ;RETURN IF FOUND ON TABLE
6809                                      ;RETURN+2 IF NOT FOUND
6810                                      ;
6811  036000  021427  177777              TERR1:  CMP     (R4),#-1        ;SEE IF ALL 1'S
```

```
036004  001405                          BEQ     1$              ;BR IF YES, NOT ON TABLE
036006  022437  007374                  CMP     (R4)+,HDC       ;SEE IF CYL MATCH
036012  001405                          BEQ     2$              ;BR IF YES
036014  005724                          TST     (R4)+           ;ELSE ADV TO NEXT CYL WORD
036016  000770                          BR      TERR1           ;& TRY AGAIN.

036020  062716  000002          1$:     ADD     #2,(SP)
036024  000207                          RTS     PC

036026  022437  007364          2$:     CMP     (R4)+,HDA       ;SEE IF SECTOR & TRACK MATCH
036032  001401                          BEQ     3$              ;BR IF YES
036034  000761                          BR      TERR1           ;OR TRY AGAIN

036036  000207                  3$:     RTS     PC
                                ;
                                ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
                                ;
036040  005037  001372          CLKON:  CLR     TIMUP
036044  005737  007520                  TST     PCLKF
```

```
036050  001004                          BNE     1$              ;BRANCH IF P-CLOCK PRESENT
036052  012777  000100  143246          MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
036060  000207                          RTS     PC
036062  012777  177777  143232  1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
036070  012777  000135  143222          MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
036076  000207                          RTS     PC              ;LINE FREQ & RUN

                                ;KW11-L & KW11-P INTERRUPT HANDLER

036100  005037  001372          CLOCK:  CLR     TIMUP
036104  005337  001366                  DEC     COUNT
036110  001010                          BNE     1$
036112  013737  001364  001366          MOV     HZ,COUNT
036120  005337  001370                  DEC     SEC
036124  001002                          BNE     1$
036126  005237  001372                  INC     TIMUP           ;SORRY, TIME IS UP
036132  000002                  1$:     RTI

                                ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF

036134  005737  007520          CLKOF:  TST     PCLKF
036140  001003                          BNE     1$              ;BRACH IF P-CLOCK PRESENT
036142  005077  143160                  CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
036146  000207                          RTS     PC
036150  005077  143144          1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
036154  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
                                ;
036156  010046                  SBPAR:  MOV     R0,-(SP)        ;SAVE R0
036160  010146                          MOV     R1,-(SP)        ;SAVE R1
036162  012700  000021                  MOV     #17.,R0         ;SHIFT COUNTER
036166  005001                          CLR     R1              ;COUNT # OF 1'S IN TEMP1
036170  000241                          CLC                     ;CLEAR CARRY

036172  006137  007412          1$:     ROL     TEMP1
036176  103001                          BCC     2$              ;BR IF CARRY CLEAR
036200  005201                          INC     R1              ;COUNT # OF 1'S
036202  005300                  2$:     DEC     R0              ;SHIFT COUNTER
036204  001372                          BNE     1$

036206  032701  000001                  BIT     #BIT0,R1
036212  001003                          BNE     3$              ;BR IF ODD # IN R0
036214  052737  100000  007412          BIS     #M.PAR,TEMP1    ;SET PARITY BIT
036222  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
036224  012600                          MOV     (SP)+,R0        ;RESTORE R0
036226  000207                          RTS     PC
```

```
                                        ;
                                        ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                        ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                        ; IE: MY LOOP MACRO
                                        ;
036230  032777  001000  142702  SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
036236  001406                          BEQ     1$              ;BR IF NO
036240  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
036244  001403                          BEQ     1$              ;BR IF NO
036246  013716  001110                  MOV     $LPERR,(SP)
036252  000002                          RTI

036254  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
036260  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.
                                        ;
036262  022626                  STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT

036264  004737  034214                  JSR     PC,SUBCLR
036270  104024                          ERROR   24              ;CERR AFTER

036272  005737  007336                  TST     UNLD            ;SEE IF HEADS UNLOADED
036276  001431                          BEQ     3$              ;BR IF NO
036300  005737  000042                  TST     42              ;SEE IF MANUAL OR AUTO MODE
036304  001403                          BEQ     1$              ;BR IF MANUAL MODE
036306  104401  045775                  TYPE    ,MSG74          ;PGM ABORT PENDING
036312  000402                          BR      2$
036314  104401  046043          1$:     TYPE    ,MSG75          ;HALT PENDING
036320                          2$:

036320  004737  034214                  JSR     PC,SUBCLR
036324  104024                          ERROR   24              ;CERR AFTER SCLR

036326  012737  000011  007354          MOV     #SRTSPL,HCS1
036334  004737  032224                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
036340  104121                          ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.

036342  013737  001414  007414          MOV     T100,TEMP2      ;SETUP TIMEOUT
036350  004737  032634                  JSR     PC,FATT1        ;FIND ATTN
036354  104074                          ERROR   74              ;NO ATTN AFTER ST SPIN CMD.

036356  005037  007336                  CLR     UNLD


036362  005737  007340          3$:     TST     BADHDR          ;SEE IF HEADERS VALID
036366  001460                          BEQ     4$              ;BR IF YES
036370  005237  007342                  INC     HPEND

036374  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
036402  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
036410  012737  000013  007354          MOV     #RECAL,HCS1
036416  004737  032224                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
```

```
036422  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD

036424  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
036432  004737  033664                  JSR     PC,GSTAT
036436  032737  020000  007402          BIT     #D.RTZ,HMR2
036444  001001                          BNE     64$
036446  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
036450  013737  001406  007414  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
036456  004737  032634                  JSR     PC,FATT1        ;FIND ATTN
036462  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD

036464  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
036472  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
036500  012737  000005  007354          MOV     #CLEAR,HCS1
036506  004737  032224                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
036512  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
036514  004737  032602                  JSR     PC,TSTATN       ;TEST FOR ATTN
036520  000401                          BR      65$
036522  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
036524                          65$:


036524  000137  031174                  JMP     FORM            ;WRITE VALID FORMATS

036530  005737  000042          4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
036534  001410                          BEQ     5$              ;BR IF MANUAL MODE
036536  104401  046100                  TYPE    ,MSG76          ;PGM ABORTED
036542  005037  031540                  CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
036546  005037  001176                  CLR     $ESCAPE
036552  000137  031512                  JMP     $EOP1           ;ABORT PGM

036556  104401  046122          5$:     TYPE    ,MSG77          ;CPU HALTED
036562  000000                          HALT
036564  000137  013356                  JMP     ST5             ;START OVER IF CONTINUE PRESSED



                                .SBTTL  UNEXPECTED TIMEOUT HANDLER

                                ;
                                ;THIS ROUTINE IS ENTERED IF THERE IS
                                ;       A.  NON EXISTANT MEMORY (NO SSYN)
                                ;       B.  BOUNDRY ERROR
                                ;       C.  STACK OVERFLOW
                                ;
                                ;
036570  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
036572  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE
036574  032777  020000  142336          BIT     #SW13,@SWR      ;INHIBIT ERR TYPOUT?
036602  001005                          BNE     1$              ;YES, DON'T TYPE
036604  104401  046303                  TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
036610  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE PC
036612  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
036614     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
036615     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
036616  032777  001000  142314  1$:     BIT     #SW9,@SWR               ;LOOP ON ERROR?
036624  001403                          BEQ     2$                      ;NO, BRANCH
036626  022626                          CMP     (SP)+,(SP)+             ;YES, RESTORE STACK
036630  000177  142252                  JMP     @$LPADR                 ;GO TO STARTING ADDR OF TEST
                                                                        ;THAT GAVE BAD TIMEOUT
036634  032777  040000  142276  2$:     BIT     #SW14,@SWR              ;LOOP ON TEST?
036642  001401                          BEQ     3$                      ;NO BRANCH
036644  000002                          RTI                             ;YES

036646  000000                  3$:     HALT                            ;UNEXPECTED TIME OUT OCCURRED
                                                                        ;AS INDICATED. YOU CAN LOOP ON
                                                                        ;ERROR, LOOP ON TEST OR INHIBIT
                                                                        ;ERROR TYPEOUT BY SETTING THOSE
                                                                        ;SWITCHES.
                                                                        ;
036650  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK
036652  000137  031512                  JMP     $EOP1                   ;ABORT TESTS

                                .SBTTL  MEMORY CHECK ENABLE TRAP

036656  012737  036672  001176  MEMERR: MOV     #1$,$ESCAPE
036664  011637  001334                  MOV     (SP),TRAPPC             ;STORE PC
036670  104041                          ERROR   41                      ;UNEXP MEM PARITY TRAP
036672  005037  001176          1$:     CLR     $ESCAPE
036676  032777  001000  142234          BIT     #SW9,@SWR               ;CHECK IF LOOP ON ERROR
036704  001001                          BNE     2$                      ;YES, FORCE STACK AND TRY AGAIN
036706  000002                          RTI                             ;ELSE RETURN

036710  012706  001100          2$:     MOV     #STACK,SP               ;INIT STACK
036714  000177  142170                  JMP     @$LPERR                 ;LOOP ON ERROR


                                .SBTTL  RK06 INTERRUPT HANDLER

036720  000240                  INTER:  NOP
036722  000240                          NOP
036724  000240                          NOP
036726  011600                          MOV     (SP),R0                 ;SAVE PC WHERE INT OCCURRED.
036730  005740                          TST     -(R0)                   ;GET PC BEFORE UPDATE.
036732  104401  044550                  TYPE    ,MSG6                   ;INT AT PC=
036736  010046                          MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
                                                                        ;;TYPE PC
036740  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
036742     006                          .BYTE   6                       ;;TYPE 6 DIGIT(S)
036743     000                          .BYTE   0                       ;;SUPPRESS LEADING ZEROS
036744  000000                          HALT
036746  000240                          NOP
036750  000240                          NOP
036752  000002                          RTI

                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;POWER DOWN ROUTINE

036754  012737  036766  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
036762  000000                          HALT
```

```
036764  000776                          BR      .-2                     ;HANG UP.

                                ;POWER UP ROUTINE

036766  005037  037040          $PWRUP: CLR     $PWRCT                  ;WAIT LOOP FOR TTY
036772  005237  037040          1$:     INC     $PWRCT                  ;WAIT FOR THE INCR
036776  001375                          BNE     1$                      ;OF WORD
037000  012737  036754  000024          MOV     #$PWRDN,PWRVEC          ;SET POWER DOWN VECTOR
037006  012737  000340  000026          MOV     #PR7,PWRVEC+2           ;PRIORITY 7
037014  012737  000340  000036          MOV     #PR7,TRAPVEC+2          ;LOCKOUT ALL INTERRUPTS FOR TRAPS
037022  012706  001100                  MOV     #STACK,SP               ;INITIALIZE STACK
037026  104401  044740                  TYPE    ,MSG11                  ;REPORT POWER FAIL
037032  000005                          RESET
037034  000137  015446                  JMP     PFSRT

037040  000000                  $PWRCT: 0                               ;WAIT COUNT FOR TTY


                                ;
                                ;DIVISION UTILITY ROUTINE
                                ;
                                ;R0-R1-R2-R3=DIVIDEND
                                ;R4-R5=DIVISOR
                                ;R0-R1=REMAINDER AFTER DIVISION
                                ;R2-R3=QUOTIENT AFTER DIVISION
                                ;ENTER WITH JSR PC,M.DPID
                                ;
037042  012746  000040          M.DPID: MOV     #40,-(SP)               ;COUNTER FOR DIVISION CYCLES
037046  010446                          MOV     R4,-(SP)                ;HI ORDER
037050  010546                          MOV     R5,-(SP)                ;LO ORDER TO THE STACK
037052  005466  000002                  NEG     2(SP)                   ;FORM NEGATIVE
037056  005416                          NEG     @SP                     ;VERSION OF DIVISOR
037060  005666  000002                  SBC     2(SP)
037064  061601                          ADD     @SP,R1
037066  005500                          ADC     R0                      ;PERFORM INIT SUBT.
037070  066600  000002                  ADD     2(SP),R0
037074  103445                          BCS     M.DP50                  ;IF CARRY THEN OVERFLOW HAS OCCURRED
037076  005046                          CLR     -(SP)                   ;THIS IS A LONGER LASTING CARRY BIT
037100  006103                  M.DP40: ROL     R3
037102  006102                          ROL     R2
037104  006101                          ROL     R1
037106  006100                          ROL     R0
037110  005716                          TST     @SP                     ;TEST CARRY INDICATOR
037112  001410                          BEQ     M.DP41                  ;IF TO CARRY THEN ADD, ELSE SUBT.
037114  005016                          CLR     @SP                     ;CLEAR UP FOR NEXT TIME
037116  066601  000002                  ADD     2(SP),R1
037122  005500                          ADC     R0                      ;ADD -(DIVISOR)
037124  005516                          ADC     @SP                     ;SET CARRY
037126  066600  000004                  ADD     4(SP),R0
037132  000404                          BR      M.DP42

037134  060501                  M.DP41: ADD     R5,R1
037136  005500                          ADC     R0                      ;ADD +(DIVISOR)
037140  005516                          ADC     @SP                     ;SET CARRY
037142  060400                          ADD     R4,R0
037144  005516                  M.DP42: ADC     @SP                     ;SET CARRY
037146  005716                          TST     @SP                     ;TEST THE UPDATE INDICATOR
```

```
037150  001401                          BEQ     .+4             ;IF 0,FORGET IT
037152  005203                          INC     R3              ;NO CARRY POSSIBLE HERE
037154  005366  000006                  DEC     6(SP)           ;DECREMENT CTR
037160  003347                          BGT     M.DP40          ;BR IF MORE TO DO
037162  006003                          ROR     R3
037164  103404                          BCS     M.DP44
037166  060501                          ADD     R5,R1
037170  005500                          ADC     R0
037172  060400                          ADD     R4,R0
037174  000241                          CLC

037176  006103                  M.DP44: ROL     R3
037200  062706  000010                  ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
037204  000242                          CLV
037206  000207                          RTS     PC

037210  062706  000006          M.DP50: ADD     #6,SP
037214  000262                          SEV
037216  000207                          RTS     PC
```

```
                                        .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

 037220                                 $SCOPE:
 037220  104407                                 CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
 037222  032777  040000  141710         1$:     BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
 037230  001114                                 BNE     $OVER                   ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
 037232  000416                         $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 037234  013746  000004                         MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 037240  012737  037260  000004                 MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
 037246  005737  177060                         TST     @#177060                ;;TIME OUT ON XOR?
 037252  012637  000004                         MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
 037256  000463                                 BR      $SVLAD                  ;;GO TO THE NEXT TEST
 037260  022626                         5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
 037262  012637  000004                         MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
 037266  000423                                 BR      7$                      ;;LOOP ON THE PRESENT TEST
 037270                                 6$:;#####END OF CODE FOR THE XOR TESTER#####
 037270  032777  000400  141642                 BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
 037276  001404                                 BEQ     2$                      ;;BR IF NO
 037300  127737  141634  001102                 CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
 037306  001465                                 BEQ     $OVER                   ;;BR IF YES
 037310  105737  001103                 2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
 037314  001421                                 BEQ     3$                      ;;BR IF NO
 037316  123737  001115  001103                 CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 037324  101015                                 BHI     3$                      ;;BR IF NO
 037326  032777  001000  141604                 BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
 037334  001404                                 BEQ     4$                      ;;BR IF NO
 037336  013737  001110  001106         7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
 037344  000446                                 BR      $OVER
 037346  105037  001103                 4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
 037352  005037  001174                         CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 037356  000415                                 BR      1$                      ;;ESCAPE TO THE NEXT TEST
 037360  032777  004000  141552         3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
 037366  001011                                 BNE     1$                      ;;BR IF YES
 037370  005737  001216                         TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
 037374  001406                                 BEQ     1$                      ;;      INHIBIT ITERATIONS
 037376  005237  001104                         INC     $ICNT                   ;;INCREMENT ITERATION COUNT
 037402  023737  001174  001104                 CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
 037410  002024                                 BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
 037412  012737  000001  001104         1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
 037420  013737  037476  001174                 MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
 037426  105237  001102                 $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
 037432  113737  001102  001214                 MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
```

```
037440  011637  001106                   MOV     (SP),$LPADR          ;;SAVE SCOPE LOOP ADDRESS
037444  011637  001110                   MOV     (SP),$LPERR          ;;SAVE ERROR LOOP ADDRESS
037450  005037  001176                   CLR     $ESCAPE              ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
037454  112737  000001  001115           MOVB    #1,$ERMAX            ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
037462  013777  001102  141452  $OVER:   MOV     $TSTNM,@DISPLAY      ;;DISPLAY TEST NUMBER
037470  013716  001106                   MOV     $LPADR,(SP)          ;;FUDGE RETURN ADDRESS
037474  000002                           RTI                          ;;FIXES PS
037476  003720                  $MXCNT:  2000.                        ;;MAX. NUMBER OF ITERATIONS
                                .SBTTL   ERROR HANDLER ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO TYPERR ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1          HALT ON ERROR
                                ;*SW13=1          INHIBIT ERROR TYPEOUTS
                                ;*SW10=1          BELL ON ERROR
                                ;*SW09=1          LOOP ON ERROR
                                ;*CALL
                                ;*       ERROR   N        ;;ERROR=EMT AND N=ERROR ITEM NUMBER

037500                          $ERROR:
037500  104407                           CKSWR                        ;;TEST FOR CHANGE IN SOFT-SWR
037502  105237  001103          7$:      INCB    $ERFLG               ;;SET THE ERROR FLAG
037506  001775                           BEQ     7$                   ;;DON'T LET THE FLAG GO TO ZERO
037510  013777  001102  141424           MOV     $TSTNM,@DISPLAY      ;;DISPLAY TEST NUMBER AND ERROR FLAG
037516  032777  002000  141414           BIT     #BIT10,@SWR          ;;BELL ON ERROR?
037524  001402                           BEQ     1$                   ;;NO - SKIP
037526  104401  001200                   TYPE    ,$BELL               ;;RING BELL
037532  005237  001112          1$:      INC     $ERTTL               ;;COUNT THE NUMBER OF ERRORS
037536  011637  001116                   MOV     (SP),$ERRPC          ;;GET ADDRESS OF ERROR INSTRUCTION
037542  162737  000002  001116           SUB     #2,$ERRPC
037550  117737  141342  001114           MOVB    @$ERRPC,$ITEMB       ;;STRIP AND SAVE THE ERROR ITEM CODE
037556  032777  020000  141354           BIT     #BIT13,@SWR          ;;SKIP TYPEOUT IF SET
037564  001004                           BNE     20$                  ;;SKIP TYPEOUTS
037566  004737  055140                   JSR     PC,TYPERR            ;;GO TO USER ERROR ROUTINE
037572  104401  001205                   TYPE    ,$CRLF
037576                          20$:
037576  122737  000001  001230           CMPB    #APTENV,$ENV         ;;RUNNING IN APT MODE
037604  001007                           BNE     2$                   ;;NO,SKIP APT ERROR REPORT
037606  113737  001114  037620           MOVB    $ITEMB,21$           ;;SET ITEM NUMBER AS ERROR NUMBER
037614  004737  040424                   JSR     PC,$ATY4             ;;REPORT FATAL ERROR TO APT
037620     000                  21$:     .BYTE   0
037621     000                           .BYTE   0
037622  000777                  22$:     BR      22$                  ;;APT ERROR LOOP
037624  005777  141310          2$:      TST     @SWR                 ;;HALT ON ERROR
037630  100002                           BPL     3$                   ;;SKIP IF CONTINUE
037632  000000                           HALT                         ;;HALT ON ERROR!
037634  104407                           CKSWR                        ;;TEST FOR CHANGE IN SOFT-SWR
037636  032777  001000  141274  3$:      BIT     #BIT09,@SWR          ;;LOOP ON ERROR SWITCH SET?.
037644  001402                           BEQ     4$                   ;;BR IF NO
037646  013716  001110                   MOV     $LPERR,(SP)          ;;FUDGE RETURN FOR LOOPING
037652  005737  001176          4$:      TST     $ESCAPE              ;;CHECK FOR AN ESCAPE ADDRESS
037656  001402                           BEQ     5$                   ;;BR IF NONE
037660  013716  001176                   MOV     $ESCAPE,(SP)         ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
037664                                    5$:
037664  022737  031600  000042                    CMP     #$ENDAD,@#42            ;;ACT-11 AUTO-ACCEPT?
037672  001001                                    BNE     6$                      ;;BRANCH IF NO
037674  000000                                    HALT                            ;;YES
037676                                    6$:
037676  000002                                    RTI                             ;;RETURN
                                          .SBTTL  TYPE ROUTINE

                                          ;;*************************************************************
                                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                          ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                          ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                          ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                          ;*
                                          ;*CALL:
                                          ;*1) USING A TRAP INSTRUCTION
                                          ;*       TYPE    ,MESADR                 ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                          ;*OR
                                          ;*       TYPE
                                          ;*       MESADR
                                          ;*

037700  105737  001157                    $TYPE:  TSTB    $TPFLG                  ;;IS THERE A TERMINAL?
037704  100002                                    BPL     1$                      ;;BR IF YES
037706  000000                                    HALT                            ;;HALT HERE IF NO TERMINAL
037710  000430                                    BR      3$                      ;;LEAVE
037712  010046                            1$:     MOV     R0,-(SP)                ;;SAVE R0
037714  017600  000002                            MOV     @2(SP),R0               ;;GET ADDRESS OF ASCIZ STRING
037720  122737  000001  001230                    CMPB    #APTENV,$ENV            ;;RUNNING IN APT MODE
037726  001011                                    BNE     62$                     ;;NO,GO CHECK FOR APT CONSOLE
037730  132737  000100  001231                    BITB    #APTSPOOL,$ENVM         ;;SPOOL MESSAGE TO APT
037736  001405                                    BEQ     62$                     ;;NO,GO CHECK FOR CONSOLE
037740  010037  037750                            MOV     R0,61$                  ;;SETUP MESSAGE ADDRESS FOR APT
037744  004737  040414                            JSR     PC,$ATY3                ;;SPOOL MESSAGE TO APT
037750  000000                            61$:    .WORD   0                       ;;MESSAGE ADDRESS
037752  132737  000040  001231            62$:    BITB    #APTCSUP,$ENVM          ;;APT CONSOLE SUPPRESSED
037760  001003                                    BNE     60$                     ;;YES,SKIP TYPE OUT
037762  112046                            2$:     MOVB    (R0)+,-(SP)             ;;PUSH CHARACTER TO BE TYPED ONTO STACK
037764  001005                                    BNE     4$                      ;;BR IF IT ISN'T THE TERMINATOR
037766  005726                                    TST     (SP)+                   ;;IF TERMINATOR POP IT OFF THE STACK
037770  012600                            60$:    MOV     (SP)+,R0                ;;RESTORE R0
037772  062716  000002                    3$:     ADD     #2,(SP)                 ;;ADJUST RETURN PC
037776  000002                                    RTI                             ;;RETURN
040000  122716  000011                    4$:     CMPB    #HT,(SP)                ;;BRANCH IF <HT>
040004  001430                                    BEQ     8$
040006  122716  000200                            CMPB    #CRLF,(SP)              ;;BRANCH IF NOT <CRLF>
040012  001006                                    BNE     5$
040014  005726                                    TST     (SP)+                   ;;POP  <CR><LF> EQUIV
040016  104401                                    TYPE                            ;;TYPE A CR AND LF
040020  001205                                    $CRLF
040022  105037  040156                            CLRB    $CHARCNT                ;;CLEAR CHARACTER COUNT
040026  000755                                    BR      2$                      ;;GET NEXT CHARACTER
040030  004737  040112                    5$:     JSR     PC,$TYPEC               ;;GO TYPE THIS CHARACTER
040034  123726  001156                    6$:     CMPB    $FILLC,(SP)+            ;;IS IT TIME FOR FILLER CHARS.?
040040  001350                                    BNE     2$                      ;;IF NO GO GET NEXT CHAR.
```

```
040042  013746  001154                  MOV     $NULL,-(SP)             ;;GET # OF FILLER CHARS. NEEDED
                                                                        ;;AND THE NULL CHAR.
040046  105366  000001          7$:     DECB    1(SP)                   ;;DOES A NULL NEED TO BE TYPED?
040052  002770                          BLT     6$                      ;;BR IF NO--GO POP THE NULL OFF OF STACK
040054  004737  040112                  JSR     PC,$TYPEC               ;;GO TYPE A NULL
040060  105337  040156                  DECB    $CHARCNT                ;;DO NOT COUNT AS A COUNT
040064  000770                          BR      7$                      ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

040066  112716  000040          8$:     MOVB    #' ,(SP)                ;;REPLACE TAB WITH SPACE
040072  004737  040112          9$:     JSR     PC,$TYPEC               ;;TYPE A SPACE
040076  132737  000007  040156          BITB    #7,$CHARCNT             ;;BRANCH IF NOT AT
040104  001372                          BNE     9$                      ;;TAB STOP
040106  005726                          TST     (SP)+                   ;;POP SPACE OFF STACK
040110  000724                          BR      2$                      ;;GET NEXT CHARACTER
040112  105777  141032          $TYPEC: TSTB    @$TPS                   ;;WAIT UNTIL PRINTER IS READY
040116  100375                          BPL     $TYPEC
040120  116677  000002  141024          MOVB    2(SP),@$TPB             ;;LOAD CHAR TO BE TYPED INTO DATA REG.
040126  122766  000015  000002          CMPB    #CR,2(SP)               ;;IS CHARACTER A CARRIAGE RETURN?
040134  001003                          BNE     1$                      ;;BRANCH IF NO
040136  105037  040156                  CLRB    $CHARCNT                ;;YES--CLEAR CHARACTER COUNT
040142  000406                          BR      $TYPEX                  ;;EXIT
040144  122766  000012  000002  1$:     CMPB    #LF,2(SP)               ;;IS CHARACTER A LINE FEED?
040152  001402                          BEQ     $TYPEX                  ;;BRANCH IF YES
040154  105227                          INCB    (PC)+                   ;;COUNT THE CHARACTER
040156  000000                  $CHARCNT: .WORD 0                       ;;CHARACTER COUNT STORAGE
040160  000207                  $TYPEX: RTS     PC

                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

040162                          $TYPDS:
040162  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
040164  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
040166  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
040170  010346                          MOV     R3,-(SP)                ;;PUSH R3 ON STACK
040172  010546                          MOV     R5,-(SP)                ;;PUSH R5 ON STACK
040174  012746  020200                  MOV     #20200,-(SP)            ;;SET BLANK SWITCH AND SIGN
040200  016605  000020                  MOV     20(SP),R5               ;;GET THE INPUT NUMBER
040204  100004                          BPL     1$                      ;;BR IF INPUT IS POS.
040206  005405                          NEG     R5                      ;;MAKE THE BINARY NUMBER POS.
040210  112766  000055  000001          MOVB    #'-,1(SP)               ;;MAKE THE ASCII NUMBER NEG.
040216  005000                  1$:     CLR     R0                      ;;ZERO THE CONSTANTS INDEX
040220  012703  040376                  MOV     #$DBLK,R3               ;;SETUP THE OUTPUT POINTER
040224  112723  000040                  MOVB    #' ,(R3)+               ;;SET THE FIRST CHARACTER TO A BLANK
040230  005002                  2$:     CLR     R2                      ;;CLEAR THE BCD NUMBER
```

```
040232  016001  040366                    MOV     $DTBL(R0),R1       ;;GET THE CONSTANT
040236  160105                    3$:     SUB     R1,R5              ;;FORM THIS BCD DIGIT
040240  002402                            BLT     4$                 ;;BR IF DONE
040242  005202                            INC     R2                 ;;INCREASE THE BCD DIGIT BY 1
040244  000774                            BR      3$
040246  060105                    4$:     ADD     R1,R5              ;;ADD BACK THE CONSTANT
040250  005702                            TST     R2                 ;;CHECK IF BCD DIGIT=0
040252  001002                            BNE     5$                 ;;FALL THROUGH IF 0
040254  105716                            TSTB    (SP)               ;;STILL DOING LEADING 0'S?
040256  100407                            BMI     7$                 ;;BR IF YES
040260  106316                    5$:     ASLB    (SP)               ;;MSD?
040262  103003                            BCC     6$                 ;;BR IF NO
040264  116663  000001  177777            MOVB    1(SP),-1(R3)       ;;YES--SET THE SIGN
040272  052702  000060            6$:     BIS     #'0,R2             ;;MAKE THE BCD DIGIT ASCII
040276  052702  000040            7$:     BIS     #' ,R2             ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
040302  110223                            MOVB    R2,(R3)+           ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
040304  005720                            TST     (R0)+              ;;JUST INCREMENTING
040306  020027  000010                    CMP     R0,#10             ;;CHECK THE TABLE INDEX
040312  002746                            BLT     2$                 ;;GO DO THE NEXT DIGIT
040314  003002                            BGT     8$                 ;;GO TO EXIT
040316  010502                            MOV     R5,R2              ;;GET THE LSD
040320  000764                            BR      6$                 ;;GO CHANGE TO ASCII
040322  105726                    8$:     TSTB    (SP)+              ;;WAS THE LSD THE FIRST NON-ZERO?
040324  100003                            BPL     9$                 ;;BR IF NO
040326  116663  177777  177776            MOVB    -1(SP),-2(R3)      ;;YES--SET THE SIGN FOR TYPING
040334  105013                    9$:     CLRB    (R3)               ;;SET THE TERMINATOR
040336  012605                            MOV     (SP)+,R5           ;;POP STACK INTO R5
040340  012603                            MOV     (SP)+,R3           ;;POP STACK INTO R3
040342  012602                            MOV     (SP)+,R2           ;;POP STACK INTO R2
040344  012601                            MOV     (SP)+,R1           ;;POP STACK INTO R1
040346  012600                            MOV     (SP)+,R0           ;;POP STACK INTO R0
040350  104401  040376                    TYPE    ,$DBLK             ;;NOW TYPE THE NUMBER
040354  016666  000002  000004            MOV     2(SP),4(SP)        ;;ADJUST THE STACK
040362  012616                            MOV     (SP)+,(SP)
040364  000002                            RTI                        ;;RETURN TO USER
040366  023420                    $DTBL:  10000.
040370  001750                            1000.
040372  000144                            100.
040374  000012                            10.
040376  000004                    $DBLK:  .BLKW   4
                                  .SBTTL  APT COMMUNICATIONS ROUTINE


                                  ;;*********************************************************
040406  112737  000001  040652    $ATY1:  MOVB    #1,$FFLG           ;;TO REPORT FATAL ERROR
040414  112737  000001  040650    $ATY3:  MOVB    #1,$MFLG           ;;TO TYPE A MESSAGE
040422  000403                            BR      $ATYC
040424  112737  000001  040652    $ATY4:  MOVB    #1,$FFLG           ;;TO ONLY REPORT FATAL ERROR
040432                            $ATYC:
040432  010046                            MOV     R0,-(SP)           ;;PUSH R0 ON STACK
040434  010146                            MOV     R1,-(SP)           ;;PUSH R1 ON STACK
040436  105737  040650                    TSTB    $MFLG              ;;SHOULD TYPE A MESSAGE?
040442  001450                            BEQ     5$                 ;;IF NOT:  BR
040444  122737  000001  001230            CMPB    #APTENV,$ENV       ;;OPERATING UNDER APT?
040452  001031                            BNE     3$                 ;;IF NOT:  BR
040454  132737  000100  001231            BITB    #APTSPOOL,$ENVM    ;;SHOULD SPOOL MESSAGES?
040462  001425                            BEQ     3$                 ;;IF NOT:  BR
```

```
040464  017600  000004                          MOV     @4(SP),R0               ;;GET MESSAGE ADDR.
040470  062766  000002  000004                  ADD     #2,4(SP)                        ;;BUMP RETURN ADDR.
040476  005737  001210                  1$:     TST     $MSGTYPE                ;;SEE IF DONE W/ LAST XMISSION?
040502  001375                                  BNE     1$                      ;;IF NOT:  WAIT
040504  010037  001224                          MOV     R0,$MSGAD               ;;PUT ADDR IN MAILBOX
040510  105720                          2$:     TSTB    (R0)+                   ;;FIND END OF MESSAGE
040512  001376                                  BNE     2$
040514  163700  001224                          SUB     $MSGAD,R0               ;;SUB START OF MESSAGE
040520  006200                                  ASR     R0                      ;;GET MESSAGE LNGTH IN WORDS
040522  010037  001226                          MOV     R0,$MSGLGT              ;;PUT LENGTH IN MAILBOX
040526  012737  000004  001210                  MOV     #4,$MSGTYPE             ;;TELL APT TO TAKE MSG.
040534  000413                                  BR      5$
040536  017637  000004  040562          3$:     MOV     @4(SP),4$               ;;PUT MSG ADDR IN JSR LINKAGE
040544  062766  000002  000004                  ADD     #2,4(SP)                        ;;BUMP RETURN ADDRESS
040552  013746  177776                          MOV     177776,-(SP)            ;;PUSH 177776 ON STACK
040556  004737  037700                          JSR     PC,$TYPE                ;;CALL TYPE MACRO
040562  000000                          4$:     .WORD   0
040564                                  5$:
040564  105737  040652                  10$:    TSTB    $FFLG                   ;;SHOULD REPORT FATAL ERROR?
040570  001416                                  BEQ     12$                     ;;IF NOT:  BR
040572  005737  001230                          TST     $ENV                    ;;RUNNING UNDER APT?
040576  001413                                  BEQ     12$                     ;;IF NOT:  BR
040600  005737  001210                  11$:    TST     $MSGTYPE                ;;FINISHED LAST MESSAGE?
040604  001375                                  BNE     11$                     ;;IF NOT:  WAIT
040606  017637  000004  001212                  MOV     @4(SP),$FATAL           ;;GET ERROR #
040614  062766  000002  000004                  ADD     #2,4(SP)                        ;;BUMP RETURN ADDR.
040622  005237  001210                          INC     $MSGTYPE                ;;TELL APT TO TAKE ERROR
040626  105037  040652                  12$:    CLRB    $FFLG                   ;;CLEAR FATAL FLAG
040632  105037  040651                          CLRB    $LFLG                   ;;CLEAR LOG FLAG
040636  105037  040650                          CLRB    $MFLG                   ;;CLEAR MESSAGE FLAG
040642  012601                                  MOV     (SP)+,R1                ;;POP STACK INTO R1
040644  012600                                  MOV     (SP)+,R0                ;;POP STACK INTO R0
040646  000207                                  RTS     PC                      ;;RETURN
040650     000                          $MFLG:  .BYTE   0                       ;;MESSG. FLAG
040651     000                          $LFLG:  .BYTE   0                       ;;LOG FLAG
040652     000                          $FFLG:  .BYTE   0                       ;;FATAL FLAG
        040654                                  .EVEN
        000200                          APTSIZE=200
        000001                          APTENV=001
        000100                          APTSPOOL=100
        000040                          APTCSUP=040
                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                        ;;***********************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)               ;;NUMBER TO BE TYPED
                                        ;*      TYPOS                           ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N                       ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M                       ;;M=1 OR 0
                                        ;*                                                      ;;1=TYPE LEADING ZEROS
                                        ;*                                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$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

040654  017646  000000                  $TYPOS: MOV     @(SP),-(SP)             ;;PICKUP THE MODE
040660  116637  000001  041077                  MOVB    1(SP),$OFILL            ;;LOAD ZERO FILL SWITCH
040666  112637  041101                          MOVB    (SP)+,$OMODE+1          ;;NUMBER OF DIGITS TO TYPE
040672  062716  000002                          ADD     #2,(SP)                 ;;ADJUST RETURN ADDRESS
040676  000406                                  BR      $TYPON
040700  112737  000001  041077  $TYPOC: MOVB    #1,$OFILL               ;;SET THE ZERO FILL SWITCH
040706  112737  000006  041101                  MOVB    #6,$OMODE+1             ;;SET FOR SIX(6) DIGITS
040714  112737  000005  041076  $TYPON: MOVB    #5,$OCNT                ;;SET THE ITERATION COUNT
040722  010346                                  MOV     R3,-(SP)                ;;SAVE R3
040724  010446                                  MOV     R4,-(SP)                ;;SAVE R4
040726  010546                                  MOV     R5,-(SP)                ;;SAVE R5
040730  113704  041101                          MOVB    $OMODE+1,R4             ;;GET THE NUMBER OF DIGITS TO TYPE
040734  005404                                  NEG     R4
040736  062704  000006                          ADD     #6,R4                   ;;SUBTRACT IT FOR MAX. ALLOWED
040742  110437  041100                          MOVB    R4,$OMODE               ;;SAVE IT FOR USE
040746  113704  041077                          MOVB    $OFILL,R4               ;;GET THE ZERO FILL SWITCH
040752  016605  000012                          MOV     12(SP),R5               ;;PICKUP THE INPUT NUMBER
040756  005003                                  CLR     R3                      ;;CLEAR THE OUTPUT WORD
040760  006105                          1$:     ROL     R5                      ;;ROTATE MSB INTO "C"
040762  000404                                  BR      3$                      ;;GO DO MSB
040764  006105                          2$:     ROL     R5                      ;;FORM THIS DIGIT
040766  006105                                  ROL     R5
040770  006105                                  ROL     R5
040772  010503                                  MOV     R5,R3
040774  006103                          3$:     ROL     R3                      ;;GET LSB OF THIS DIGIT
040776  105337  041100                          DECB    $OMODE                  ;;TYPE THIS DIGIT?
041002  100016                                  BPL     7$                      ;;BR IF NO
041004  042703  177770                          BIC     #177770,R3              ;;GET RID OF JUNK
041010  001002                                  BNE     4$                      ;;TEST FOR 0
041012  005704                                  TST     R4                      ;;SUPPRESS THIS 0?
041014  001403                                  BEQ     5$                      ;;BR IF YES
041016  005204                          4$:     INC     R4                      ;;DON'T SUPPRESS ANYMORE 0'S
041020  052703  000060                          BIS     #'0,R3                  ;;MAKE THIS DIGIT ASCII
041024  052703  000040                  5$:     BIS     #' ,R3                  ;;MAKE ASCII IF NOT ALREADY
041030  110337  041074                          MOVB    R3,8$                   ;;SAVE FOR TYPING
041034  104401  041074                          TYPE    ,8$                     ;;GO TYPE THIS DIGIT
041040  105337  041076                  7$:     DECB    $OCNT                   ;;COUNT BY 1
041044  003347                                  BGT     2$                      ;;BR IF MORE TO DO
041046  002402                                  BLT     6$                      ;;BR IF DONE
041050  005204                                  INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
041052  000744                                  BR      2$                      ;;GO DO THE LAST DIGIT
041054  012605                          6$:     MOV     (SP)+,R5                ;;RESTORE R5
041056  012604                                  MOV     (SP)+,R4                ;;RESTORE R4
041060  012603                                  MOV     (SP)+,R3                ;;RESTORE R3
041062  016666  000002  000004                  MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
041070  012616                                  MOV     (SP)+,(SP)
```

```
041072  000002                          RTI                                 ;;RETURN
041074     000                  8$:     .BYTE   0                           ;;STORAGE FOR ASCII DIGIT
041075     000                          .BYTE   0                           ;;TERMINATOR FOR TYPE ROUTINE
041076     000                  $OCNT:  .BYTE   0                           ;;OCTAL DIGIT COUNTER
041077     000                  $OFILL: .BYTE   0                           ;;ZERO FILL SWITCH
041100  000000                  $OMODE: .WORD   0                           ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*******************************************************************
                                .ENABL  LSB
041102  000000                  $TKCNT: .WORD   0                           ;;NUMBER OF ITEMS IN QUEUE
041104  000000                  $TKQIN: .WORD   0                           ;;INPUT POINTER
041106  000000                  $TKQOUT: .WORD  0                           ;;OUTPUT POINTER
041110  000001                  $TKQSRT: .BLKB  1                           ;;TTY KEYBOARD QUEUE
        041111                  $TKQEND=.
        041112                  .EVEN

                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN

041112  005037  041102          $TKINT: CLR     $TKCNT              ;;CLEAR COUNT OF ITEMS IN QUEUE
041116  012737  041110  041104          MOV     #$TKQSRT,$TKQIN     ;;MOVE THE STARTING ADDRESS OF THE
041124  013737  041104  041106          MOV     $TKQIN,$TKQOUT      ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
041132  012737  041162  000060          MOV     #$TKSRV,@#TKVEC     ;;INITIALIZE THE KEYBOARD VECTOR
041140  012737  000200  000062          MOV     #200,@#TKVEC+2      ;;"BR" LEVEL 4
041146  005777  137774                  TST     @$TKB               ;;CLEAR DONE FLAG
041152  012777  000100  137764          MOV     #100,@$TKS          ;;ENABLE TTY KEYBOARD INTERRUPT
041160  000207                          RTS     PC                  ;;RETURN TO CALLER

                                ;*TK SERVICE ROUTINE
                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                ;*IT IN THE QUEUE.
                                ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
                                ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
                                ;
041162  117746  137760          $TKSRV: MOVB    @$TKB,-(SP)         ;;PICKUP THE CHARACTER
041166  042716  177600                  BIC     #↑C177,(SP)         ;;STRIP THE JUNK
041172  021627  000003                  CMP     (SP),#3             ;;IS IT A CONTROL C?
041176  001007                          BNE     1$                  ;;BRANCH IF NO
041200  104401  042310                  TYPE    ,$CNTLC             ;;TYPE A CONTROL-C (↑C)
041204  004737  041112                  JSR     PC,$TKINT           ;;INIT THE KEYBOARD
041210  005726                          TST     (SP)+               ;;CLEAN UP STACK
041212  000137  036262                  JMP     STOP                ;;CONTROL C RESTART
041216  021627  000007          1$:     CMP     (SP),#7             ;;IS IT A CONTROL G?
041222  001004                          BNE     2$                  ;;BRANCH IF NO
041224  022737  000176  001140          CMP     #SWREG,SWR          ;;IS SOFT-SWR SELECTED?
041232  001500                          BEQ     6$                  ;;GO TO SWR CHANGE

041234                          2$:
041234  022737  000001  041102          CMP     #1,$TKCNT           ;;IS THE QUEUE FULL?
```

```
 041242  001004                           BNE     3$                      ;;BRANCH IF NO
 041244  104401  001200                   TYPE    ,$BELL                  ;;RING THE TTY BELL
 041250  005726                           TST     (SP)+                   ;;CLEAN CHARACTER OFF OF STACK
 041252  000451                           BR      5$                      ;;EXIT
 041254  021627  000023           3$:     CMP     (SP),#23                ;;IS IT A CONTROL-S?
 041260  001021                           BNE     32$                     ;;BRANCH IF NO
 041262  005077  137656                   CLR     @$TKS                   ;;DISABLE TTY KEYBOARD INTERRUPTS
 041266  005726                           TST     (SP)+                   ;;CLEAN CHAR OFF STACK
 041270  105777  137650           31$:    TSTB    @$TKS                   ;;WAIT FOR A CHAR
 041274  100375                           BPL     31$                     ;;LOOP UNTIL ITS THERE
 041276  117746  137644                   MOVB    @$TKB,-(SP)             ;;GET THE CHARACTER
 041302  042716  177600                   BIC     #↑C177,(SP)             ;;MAKE IT 7-BIT ASCII
 041306  022627  000021                   CMP     (SP)+,#21               ;;IS IT A CONTROL-Q?
 041312  001366                           BNE     31$                     ;;BRANCH IF NO
 041314  012777  000100  137622           MOV     #100,@$TKS              ;;REENABLE TTY KEYBOARD INTERRUPTS
 041322  000002                           RTI                             ;;RETURN
 041324  005237  041102           32$:    INC     $TKCNT                  ;;COUNT THIS CHARACTER
 041330  021627  000140                   CMP     (SP),#140               ;;IS IT UPPER CASE?
 041334  002405                           BLT     4$                      ;;BRANCH IF YES
 041336  021627  000175                   CMP     (SP),#175               ;;IS IT A SPECIAL CHAR?
 041342  003002                           BGT     4$                      ;;BRANCH IF YES
 041344  042716  000040                   BIC     #40,(SP)                ;;MAKE IT UPPER CASE
 041350  112677  177530           4$:     MOVB    (SP)+,@$TKQIN           ;;AND PUT IT IN QUEUE
 041354  005237  041104                   INC     $TKQIN                  ;;UPDATE THE POINTER
 041360  023727  041104  041111           CMP     $TKQIN,#$TKQEND         ;;GO OFF THE END?
 041366  001003                           BNE     5$                      ;;BRANCH IF NO
 041370  012737  041110  041104           MOV     #$TKQSRT,$TKQIN         ;;RESET THE POINTER
 041376  000002                   5$:     RTI                             ;;RETURN

                                  ;;*****************************************************************
                                  ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                  ;;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                                  ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                                  ;;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
 041400  022737  000176  001140   $CKSWR: CMP     #SWREG,SWR              ;;IS THE SOFT-SWR SELECTED
 041406  001124                           BNE     15$                     ;;EXIT IF NOT
 041410  105777  137530                   TSTB    @$TKS                   ;;IS A CHAR WAITING?
 041414  100121                           BPL     15$                     ;;IF NOT, EXIT
 041416  117746  137524                   MOVB    @$TKB,-(SP)             ;;YES
 041422  042716  177600                   BIC     #↑C177,(SP)             ;;MAKE IT  7-BIT ASCII
 041426  021627  000007                   CMP     (SP),#7                 ;;IS IT A CONTROL-G?
 041432  001300                           BNE     2$                      ;;IF NOT, PUT IT IN THE TTY QUEUE
                                                                          ;;AND EXIT

                                  ;;*****************************************************************
                                  ;;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                                  ;;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                                  ;;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 041434  123727  001134  000001   6$:     CMPB    $AUTOB,#1               ;;ARE WE RUNNING IN AUTO-MODE?
 041442  001674                           BEQ     2$                      ;;BRANCH IF YES
 041444  005726                           TST     (SP)+                   ;;CLEAR CONTROL-G OFF STACK
 041446  004737  041112                   JSR     PC,$TKINT               ;;FLUSH THE TTY INPUT QUEUE
 041452  005077  137466                   CLR     @$TKS                   ;;DISABLE TTY KEYBOARD INTERRUPTS
 041456  112737  000001  001135           MOVB    #1,$INTAG               ;;SET INTERRUPT MODE INDICATOR

 041464  104401  042322                   TYPE    ,$CNTLG                 ;;ECHO THE CONTROL-G (↑G)
```

```
041470  104401  042327          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
041474  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
041500  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
041502  104401  042340                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
041506  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
041510  005046                          CLR     -(SP)           ;;THE NEW SWR
041512  105777  137426          7$:     TSTB    @$TKS           ;;CHAR THERE?
041516  100375                          BPL     7$              ;;IF NOT TRY AGAIN

041520  117746  137422                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
041524  042716  177600                  BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII

041530  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
041534  001015                          BNE     9$              ;;BRANCH IF NOT
041536  104401  042310                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (↑C)
041542  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
041546  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
041554  001003                          BNE     8$              ;;BRANCH IF NO
041556  012777  000100  137360          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
041564  000137  036262          8$:     JMP     STOP            ;;CONTROL-C RESTART


041570  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
041574  001005                          BNE     10$             ;;BRANCH IF NOT
041576  104401  042315                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (↑U)
041602  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
041606  000737                          BR      19$             ;;LET'S TRY IT AGAIN


041610  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
041614  001022                          BNE     16$             ;;BRANCH IF NO
041616  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
041622  001403                          BEQ     11$             ;;BRANCH IF YES
041624  016677  000002  137306          MOV     2(SP),@SWR      ;;SAVE NEW SWR
041632  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
041636  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
041642  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
041650  001003                          BNE     15$             ;;BRANCH IF NOT
041652  012777  000100  137264          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
041660  000002                  15$:    RTI                     ;;RETURN
041662  004737  040112          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
041666  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
041672  002420                          BLT     18$             ;;BRANCH IF YES
041674  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
041700  003015                          BGT     18$             ;;BRANCH IF YES
041702  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
041706  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
041712  001403                          BEQ     17$             ;;BRANCH IF YES
041714  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
041716  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
041720  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
041722  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
041726  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
041732  000667                          BR      7$              ;;GET THE NEXT ONE
041734  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
041740  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
                                        ;
041742  011646                          $RDCHR: MOV     (SP),-(SP)              ;;PUSH DOWN THE PC AND
041744  016666  000004  000002                  MOV     4(SP),2(SP)             ;;THE PS
041752  005066  000004                          CLR     4(SP)                   ;;GET READY FOR A CHARACTER
041756  005046                                  CLR     -(SP)                   ;;PUT NEW PS ON STACK
041760  012746  041766                          MOV     #64$,-(SP)              ;;PUT NEW PC ON  STACK
041764  000002                                  RTI                             ;;POP NEW PC AND PS
041766                                  64$:
041766  005737  041102                  1$:     TST     $TKCNT                  ;;WAIT ON A CHARACTER
041772  001775                                  BEQ     1$
041774  005337  041102                          DEC     $TKCNT                  ;;DECREMENT THE COUNTER
042000  117766  177102  000004                  MOVB    @$TKQOUT,4(SP)          ;;GET ONE CHARACTER
042006  005237  041106                          INC     $TKQOUT                 ;;UPDATE THE POINTER
042012  023727  041106  041111                  CMP     $TKQOUT,#$TKQEND        ;;DID IT GO OFF OF THE END?
042020  001003                                  BNE     2$                      ;;BRANCH IF NO
042022  012737  041110  041106                  MOV     #$TKQSRT,$TKQOUT        ;;RESET THE POINTER
042030  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

042032  010346                          $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
042034  005046                                  CLR     -(SP)                   ;;CLEAR THE RUBOUT KEY
042036  012703  042266                  1$:     MOV     #$TTYIN,R3              ;;GET ADDRESS
042042  022703  042310                  2$:     CMP     #$TTYIN+22,R3           ;;BUFFER FULL?
042046  101456                                  BLOS    4$                      ;;BR IF YES
042050  104410                                  RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
042052  112613                                  MOVB    (SP)+,(R3)              ;;GET CHARACTER
042054  122713  000177                  10$:    CMPB    #177,(R3)               ;;IS IT A RUBOUT
042060  001022                                  BNE     5$                      ;;BR IF NO
042062  005716                                  TST     (SP)                    ;;IS THIS THE FIRST RUBOUT?
042064  001007                                  BNE     6$                      ;;BR IF NO
042066  112737  000134  042264                  MOVB    #'\,9$                  ;;TYPE A BACK SLASH
042074  104401  042264                          TYPE    ,9$
042100  012716  177777                          MOV     #-1,(SP)                ;;SET THE RUBOUT KEY
042104  005303                          6$:     DEC     R3                      ;;BACKUP BY ONE
042106  020327  042266                          CMP     R3,#$TTYIN              ;;STACK EMPTY?
042112  103434                                  BLO     4$                      ;;BR IF YES
042114  111337  042264                          MOVB    (R3),9$                 ;;SETUP TO TYPEOUT THE DELETED CHAR.
042120  104401  042264                          TYPE    ,9$                     ;;GO TYPE
042124  000746                                  BR      2$                      ;;GO READ ANOTHER CHAR.
042126  005716                          5$:     TST     (SP)                    ;;RUBOUT KEY SET?
042130  001406                                  BEQ     7$                      ;;BR IF NO
```

```
042132  112737  000134  042264           MOVB    #'\,9$              ;;TYPE A BACK SLASH
042140  104401  042264                   TYPE    ,9$
042144  005016                           CLR     (SP)                ;;CLEAR THE RUBOUT KEY
042146  122713  000025           7$:     CMPB    #25,(R3)            ;;IS CHARACTER A CTRL U?
042152  001003                           BNE     8$                  ;;BR IF NO
042154  104401  042315                   TYPE    ,$CNTLU             ;;TYPE A CONTROL "U"
042160  000726                           BR      1$                  ;;GO START OVER
042162  122713  000022           8$:     CMPB    #22,(R3)            ;;IS CHARACTER A "↑R"?
042166  001011                           BNE     3$                  ;;BRANCH IF NO
042170  105013                           CLRB    (R3)                ;;CLEAR THE CHARACTER
042172  104401  001205                   TYPE    ,$CRLF              ;;TYPE A "CR" & "LF"
042176  104401  042266                   TYPE    ,$TTYIN             ;;TYPE THE INPUT STRING
042202  000717                           BR      2$                  ;;GO PICKUP ANOTHER CHACTER
042204  104401  001204           4$:     TYPE    ,$QUES              ;;TYPE A '?'
042210  000712                           BR      1$                  ;;CLEAR THE BUFFER AND LOOP
042212  111337  042264           3$:     MOVB    (R3),9$             ;;ECHO THE CHARACTER
042216  104401  042264                   TYPE    ,9$
042222  122723  000015                   CMPB    #15,(R3)+           ;;CHECK FOR RETURN
042226  001305                           BNE     2$                  ;;LOOP IF NOT RETURN
042230  105063  177777                   CLRB    -1(R3)              ;;CLEAR RETURN (THE 15)
042234  104401  001206                   TYPE    ,$LF                ;;TYPE A LINE FEED
042240  005726                           TST     (SP)+               ;;CLEAN RUBOUT KEY FROM THE STACK
042242  012603                           MOV     (SP)+,R3            ;;RESTORE R3
042244  011646                           MOV     (SP),-(SP)          ;;ADJUST THE STACK AND PUT ADDRESS OF THE
042246  016666  000004  000002           MOV     4(SP),2(SP)         ;;       FIRST ASCII CHARACTER ON IT
042254  012766  042266  000004           MOV     #$TTYIN,4(SP)
042262  000002                           RTI                         ;;RETURN
042264     000                   9$:     .BYTE   0                   ;;STORAGE FOR ASCII CHAR. TO TYPE
042265     000                           .BYTE   0                   ;;TERMINATOR
042266  000022                   $TTYIN: .BLKB   22                  ;;RESERVE 22 BYTES FOR TTY INPUT
042310  041536  005015     000   $CNTLC: .ASCIZ  /↑C/<15><12>        ;;CONTROL "C"
042315     136  006525  000012   $CNTLU: .ASCIZ  /↑U/<15><12>        ;;CONTROL "U"
042322  043536  005015     000   $CNTLG: .ASCIZ  /↑G/<15><12>        ;;CONTROL "G"
042327     015  051412  051127   $MSWR:  .ASCIZ  <15><12>/SWR = /
042334  036440  000040
042340  020040  042516  020127   $MNEW:  .ASCIZ  /  NEW = /
042346  020075     000
        042352                   .EVEN
                                 .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                 ;;*************************************************************
                                 ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                 ;*CHANGE IT TO BINARY.
                                 ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                 ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                 ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                 ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                 ;*CALL:
                                 ;*       RDOCT                        ;;READ AN OCTAL NUMBER
                                 ;*       RETURN HERE                  ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                 ;*                                    ;;HIGH ORDER BITS ARE IN $HIOCT

042352  011646                   $RDOCT: MOV     (SP),-(SP)          ;;PROVIDE SPACE FOR THE
042354  016666  000004  000002           MOV     4(SP),2(SP)         ;;INPUT NUMBER
042362  010046                           MOV     R0,-(SP)            ;;PUSH R0 ON STACK
042364  010146                           MOV     R1,-(SP)            ;;PUSH R1 ON STACK
```

```
042366  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042370  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
042372  012600                          MOV     (SP)+,RC        ;;GET ADDRESS OF 1ST CHARACTER
042374  010037  042500                  MOV     R0,5$           ;;AND SAVE IT
042400  005001                          CLR     R1              ;;CLEAR DATA WORD
042402  005002                          CLR     R2
042404  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
042406  001420                          BEQ     3$              ;;IF ZERO GET OUT
042410  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
042414  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
042416  122716  000067                  CMPB    #'7,(SP)
042422  002423                          BLT     4$
042424  006301                          ASL     R1              ;;*2
042426  006102                          ROL     R2
042430  006301                          ASL     R1              ;;*4
042432  006102                          ROL     R2
042434  006301                          ASL     R1              ;;*8
042436  006102                          ROL     R2
042440  042716  177770                  BIC     #↑C7,(SP)       ;;STRIP THE ASCII JUNK
042444  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
042446  000756                          BR      2$              ;;LOOP
042450  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
042452  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
042456  010237  042510                  MOV     R2,$HIOCT
042462  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
042464  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
042466  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
042470  000002                          RTI                     ;;RETURN
042472  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
042474  105010                          CLRB    (R0)            ;;SET A TERMINATOR
042476  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
042500  000000                  5$:     .WORD   0
042502  104401  001204                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
042506  000730                          BR      1$              ;;TRY AGAIN
042510  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
                                .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE


                                ;;************************************************************
                                ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
                                ;*UNSIGNED OCTAL ASCIZ NUMBER.
                                ;*CALL
                                ;*       MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                                ;*       JSR     PC,@#$DB20      ;;CALL THE ROUTINE
                                ;*       RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK


042512  104413                  $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
042514  016601  000002                  MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
042520  012705  042631                  MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
042524  012704  000014                  MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
042530  012703  177770                  MOV     #↑C7,R3         ;;MASK
042534  012100                          MOV     (R1)+,R0        ;;LOWER WORD
042536  012101                          MOV     (R1)+,R1        ;;HIGH WORD
042540  005002                          CLR     R2              ;;TERMINATOR
042542  110245                  1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
042544  010002                          MOV     R0,R2           ;;GET THIS DIGIT
```

```
042546  005304                          DEC     R4              ;;COUNT THIS CHARACTER
042550  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
042552  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
042554  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
042556  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
042562  104414                          RESREG                  ;;RESTORE ALL REGISTERS
042564  000207                          RTS     PC              ;;RETURN TO USER
042566  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
042570  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
042572  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
042574  006001                          ROR     R1              ;;
042576  006000                          ROR     R0
042600  006001                          ROR     R1
042602  006000                          ROR     R0
042604  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
042606  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
042612  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
042614  000016                  $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
                                .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE

                                ;;*******************************************************************
                                ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
                                ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
                                ;*POSITIVE.
                                ;*CALL
                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                                ;*      JSR     PC,@#$DB2D
                                ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
                                                                ;;IS ON THE STACK


042632  104413                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
042634  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
042640  012700  043012                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
042644  010066  000002                  MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
042650  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
042652  012202                          MOV     (R2)+,R2
042654  012737  000012  042730          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
042662  012704  042742                  MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
042666  012705  042744                  MOV     #$TNPWR+2,R5
042672  005003                  1$:     CLR     R3              ;;CLEAR PARTIAL
042674  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
042676  005602                          SBC     R2
042700  161502                          SUB     (R5),R2
042702  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
042704  005203                          INC     R3              ;;ADD 1 TO PARTIAL
042706  000772                          BR      2$              ;;LOOP
042710  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
042712  005502                          ADC     R2
042714  062402                          ADD     (R4)+,R2
042716  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
042720  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
042724  110320                          MOVB    R3,(R0)+        ;;SAVE IT
042726  005327                          DEC     (PC)+           ;;DONE?
042730  000000                  4$:     .WORD   0
042732  001357                          BNE     1$              ;;BR IF NO
```

```
042734  105020                            CLRB    (R0)+             ;;TERMINATOR
042736  104414                            RESREG                    ;;RESTORE REGISTERS
042740  000207                            RTS     PC                ;;RETURN
042742  145000                   $TNPWR:  145000                    ;;1.0E09
042744  035632                            35632
042746  160400                            160400                    ;;1.0E08
042750  002765                            2765
042752  113200                            113200                    ;;1.0E07
042754  000230                            230
042756  041100                            041100                    ;;1.0E06
042760  000017                            17
042762  103240                            103240                    ;;1.0E05
042764  000001                            1
042766  023420                            23420                     ;;1.0E04
042770  000000                            0
042772  001750                            1750                      ;;1.0E03
042774  000000                            0
042776  000144                            144                       ;;1.0E02
043000  000000                            0
043002  000012                            12                        ;;1.0E01
043004  000000                            0
043006  000001                            1                         ;;1.0E00
043010  000000                            0
043012  000014                   $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


043026  016637  000002  043056   $SB2D:   MOV     2(SP),1$          ;;SAVE BINARY NUMBER
043034  012746  043056                    MOV     #1$,-(SP)         ;;SET POINTER
043040  004737  042632                    JSR     PC,@#$DB2D        ;;CALL DOUBLE LENGTH CONVERT
043044  062716  000005                    ADD     #5,(SP)           ;;ONLY ALLOW FIVE CHARACTERS
043050  012666  000002                    MOV     (SP)+,2(SP)       ;;PICKUP POINTER
043054  000207                            RTS     PC                ;;RETURN
043056  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


043062  010046                   $SUPRS:  MOV     R0,-(SP)          ;;SAVE R0
043064  016600  000004                    MOV     4(SP),R0          ;;PICKUP THE POINTER
043070  105710                   1$:      TSTB    (R0)              ;;TERMINATEOR?
043072  001403                            BEQ     2$                ;;BR IF YES
```

```
043074  122720  000060          CMPB    #'0,(R0)+               ;;IS THIS AN ASCII "0" ?
043100  001773                  BEQ     1$                      ;;BR IF YES
043102  005300          2$:     DEC     R0                      ;;BACKUP BY "1"
043104  010037  043112          MOV     R0,3$                   ;;SAVE FOR TYPING
043110  104401                  TYPE                            ;;GO TYPE
043112  000000          3$:     .WORD   0                       ;;ASCIZ POINTER GOES HERE
043114  012600                  MOV     (SP)+,R0                ;;RESTORE R0
043116  012616                  MOV     (SP)+,(SP)              ;;RESTORE THE STACK
043120  000207                  RTS     PC                      ;;RETURN
                        .SBTTL  INTEGER MULTIPLY ROUTINE

                        ;;*************************************************************
                        ;*CALL
                        ;*      MOV     MULTIPLER,-(SP)
                        ;*      MOV     MULTIPLICAND,-(SP)
                        ;*      JSR     PC,@#$MULT
                        ;*      RETURN  ;;PRODUCT IS ON THE STACK
                        ;*
                        ;*      STACK   PRODUCT
                        ;*      -----   -------
                        ;*      TOP     LSB'S
                        ;*      +2      MSB'S

043122                  $MULT:
043122  010046                  MOV     R0,-(SP)                ;;PUSH R0 ON STACK
043124  010146                  MOV     R1,-(SP)                ;;PUSH R1 ON STACK
043126  010246                  MOV     R2,-(SP)                ;;PUSH R2 ON STACK
043130  005046                  CLR     -(SP)                   ;;CLEAR THE SIGN KEY
043132  016601  000012          MOV     12(SP),R1               ;;GET THE MULTIPLICAND
043136  100002                  BPL     1$                      ;;BR IF PLUS
043140  005216                  INC     (SP)                    ;;SET THE SIGN KEY
043142  005401                  NEG     R1                      ;;MAKE THE MULTIPLICAND POSTIVE
043144  016602  000014  1$:     MOV     14(SP),R2               ;;GET THE MULTIPLIER
043150  100002                  BPL     2$                      ;;BR IF PLUS
043152  005316                  DEC     (SP)                    ;;UPDATE THE SIGN KEY
043154  005402                  NEG     R2                      ;;MAKE THE MULTIPLIER POSTIVE
043156  012746  000021  2$:     MOV     #17.,-(SP)              ;;SET THE LOOP COUNT
043162  005000                  CLR     R0                      ;;SETUP FOR THE MULTIPLY LOOP
043164  103001          3$:     BCC     4$                      ;;DON'T ADD IF MULTIPLICAND = 0
043166  060200                  ADD     R2,R0
043170  006000          4$:     ROR     R0                      ;;POSITION THE PARITIAL PRODUCT AND
043172  006001                  ROR     R1                      ;;THE MULTIPLICAND
043174  005316                  DEC     (SP)                    ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
043176  001372                  BNE     3$                      ;;BR IF NO
043200  022616                  CMP     (SP)+,(SP)              ;;SHOULD PRODUCT BE NEGATIVE?
043202  001403                  BEQ     5$                      ;;GO TO EXIT IF NO
043204  005400                  NEG     R0                      ;;YES--SO MAKE IT SO
043206  005401                  NEG     R1
043210  005600                  SBC     R0
043212  005726          5$:     TST     (SP)+                   ;;CLEAR SIGN INFO. OFF OF STACK
043214  010066  000012          MOV     R0,12(SP)               ;;PUT THE PRODUCT ON THE STACK (MSB'S)
043220  010166  000010          MOV     R1,10(SP)               ;;LSB'S
043224  012602                  MOV     (SP)+,R2                ;;POP STACK INTO R2
043226  012601                  MOV     (SP)+,R1                ;;POP STACK INTO R1
043230  012600                  MOV     (SP)+,R0                ;;POP STACK INTO R0
043232  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

043234                                  $SAVREG:
043234  010046                                  MOV     R0,-(SP)                ;;PUSH R0 ON STACK
043236  010146                                  MOV     R1,-(SP)                ;;PUSH R1 ON STACK
043240  010246                                  MOV     R2,-(SP)                ;;PUSH R2 ON STACK
043242  010346                                  MOV     R3,-(SP)                ;;PUSH R3 ON STACK
043244  010446                                  MOV     R4,-(SP)                ;;PUSH R4 ON STACK
043246  010546                                  MOV     R5,-(SP)                ;;PUSH R5 ON STACK
043250  016646  000022                          MOV     22(SP),-(SP)            ;;SAVE PS OF MAIN FLOW
043254  016646  000022                          MOV     22(SP),-(SP)            ;;SAVE PC OF MAIN FLOW
043260  016646  000022                          MOV     22(SP),-(SP)            ;;SAVE PS OF CALL
043264  016646  000022                          MOV     22(SP),-(SP)            ;;SAVE PC OF CALL
043270  000002                                  RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
043272                                  $RESREG:
043272  012666  000022                          MOV     (SP)+,22(SP)            ;;RESTORE PC OF CALL
043276  012666  000022                          MOV     (SP)+,22(SP)            ;;RESTORE PS OF CALL
043302  012666  000022                          MOV     (SP)+,22(SP)            ;;RESTORE PC OF MAIN FLOW
043306  012666  000022                          MOV     (SP)+,22(SP)            ;;RESTORE PS OF MAIN FLOW
043312  012605                                  MOV     (SP)+,R5                ;;POP STACK INTO R5
043314  012604                                  MOV     (SP)+,R4                ;;POP STACK INTO R4
043316  012603                                  MOV     (SP)+,R3                ;;POP STACK INTO R3
043320  012602                                  MOV     (SP)+,R2                ;;POP STACK INTO R2
043322  012601                                  MOV     (SP)+,R1                ;;POP STACK INTO R1
043324  012600                                  MOV     (SP)+,R0                ;;POP STACK INTO R0
043326  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.

043330  010046                          $TRAP:  MOV     R0,-(SP)                ;;SAVE R0
043332  016600  000002                          MOV     2(SP),R0                ;;GET TRAP ADDRESS
043336  005740                                  TST     -(R0)                   ;;BACKUP BY 2
```

```
043340  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
043342  006300                          ASL     R0              ;;POSITION FOR INDEXING
043344  016000  043364                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
043350  000200                          RTS     R0              ;;GO TO ROUTINE


                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

043352  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
043354  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
043362  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION

                                ;       ROUTINE
                                        -------
043364  043352                  $TRPAD: .WORD   $TRAP2
043366  037700                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
043370  040700                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
043372  040654                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
043374  040714                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
043376  040162                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

043400  041470                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

043402  041400                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
043404  041742                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
043406  042032                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
043410  042352                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
043412  043234                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
043414  043272                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
043416  036230                          SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
```

```

                                                  .SBTTL  SERVICE MESSAGES


        043420  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06-RK07 DRIVE DIAGNOSTIC--PART 2/
        043426  051525  051040  030113
        043434  026466  045522  033460
        043442  042040  044522  042526
        043450  042040  040511  047107
        043456  051517  044524  026503
        043464  050055  051101  020124
        043472     062
        043473     015  041412  051132          .ASCII  <CR><LF>/CZR6IED/<CR><LF>
        043500  044466  030105  005015
        043506  005015  025011  025052          .ASCII  <CR><LF>/        ***** CAUTION *****/<CR><LF>
        043514  025052  041440  052501
        043522  044524  047117  025040
        043530  025052  025052  005015
        043536  005015  044124  051511          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY BY TYPING CONTROL-C/
        043544  050040  047522  051107
        043552  046501  051440  047510
        043560  046125  020104  041040
        043566  020105  040510  052114
        043574  042105  047440  046116
        043602  020131  054502  052040
        043610  050131  047111  020107
        043616  047503  052116  047522
        043624  026514     103
        043627     015  047412  044124          .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND, OR THE DRIVE/
        043634  051105  044527  042523
        043642  020054  040503  052122
        043650  044522  043504  020105
        043656  047506  046522  052101
        043664  044524  043516  040440
        043672  042116  020054  051117
        043700  052040  042510  042040
        043706  044522  042526
        043712  005015  040515  020131          .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/
        043720  042502  046040  043105
        043726  020124  047111  040440
        043734  020116  047125  042504
        043742  042524  046522  047111
        043750  042105  051440  040524
        043756  042524
        043760  005015  047111  052111          .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
        043766  040511  046114  026131
        043774  042040  044522  042526
        044002  020123  047524  041040
        044010  020105  042524  052123
        044016  042105  051440  047510
        044024  046125  020104  040510
        044032  042526  006472     012
        044037     015  040412  020056          .ASCII  <CR><LF>/A. HEADS MANUALLY LOADED/
        044044  042510  042101  020123
        044052  040515  052516  046101
        044060  054514  046040  040517
```

```
044066  042504     104
044071     015  041012  020056             .ASCII  <CR><LF>/B. CORRECT PORT SELECTED/
044076  047503  051122  041505
044104  020124  047520  052122
044112  051440  046105  041505
044120  042524     104
044123     015  041412  020056             .ASCII  <CR><LF>/C. WRITE LOCK DISABLED/
044130  051127  052111  020105
044136  047514  045503  042040
044144  051511  041101  042514
044152     104
044153     015  042012  020056             .ASCII  <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
044160  051104  053111  020105
044166  042522  042101  020131
044174  047111  044504  040503
044202  047524  020122  047117
044210  005015
044212  005015  051104  053111             .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
044220  051505  047040  052117
044226  052040  020117  042502
044234  052040  051505  042524
044242  020104  052515  052123
044250  044040  053101     105
044255     015  041012  052117             .ASCIZ  <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
044262  020110  047520  052122
044270  020123  042504  042523
044276  042514  052103  042105
044304  005015     000

044307     015  041412  040510     MSG2:   .ASCII  <CR><LF>/CHANGE XXDP PACK/
044314  043516  020105  054130
044322  050104  050040  041501
044330     113
044331     015  041412  042514             .ASCIZ  <CR><LF>/CLEAR LOC 40,RESTART PROGRAM/
044336  051101  046040  041517
044344  032040  026060  042522
044352  052123  051101  020124
044360  051120  043517  040522
044366  000115
044370  005015  051104  053111     MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
044376  024105  024523  052040
044404  020117  042502  052040
044412  051505  042524  035104
044420  000040
044422  005015  054524  042520     MSG4:   .ASCIZ  <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440:  /
044430  041040  051525  020123
044436  042101  051104  051505
044444  020123  043111  047040
044452  052117  030440  033467
044460  032064  035060  020040
044466     000
044467     015  052012  050131     MSG5:   .ASCIZ  <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210:  /
044474  020105  047503  052116
044502  047522  046114  051105
044510  044440  052116  051105
044516  052522  052120  053040
```

```
044524  041505  047524  020122
044532  043111  047040  052117
044540  031040  030061  020072
044546  000040
044550  005015  047111  042524   MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
044556  051122  050125  020124
044564  041517  052503  051122
044572  042105  040440  020124
044600  041520  000075
044604  005015  051104  053111   MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
044612  020105  020060  044527
044620  046114  047040  052117
044626  041040  020105  042524
044634  052123  042105     000
044641     015  051012  040505   MSG8:   .ASCIZ  <CR><LF>/READ DATA WITH OFFSET TEST/<CR><LF>
044646  020104  040504  040524
044654  053440  052111  020110
044662  043117  051506  052105
044670  052040  051505  006524
044676  000012
044700  005015  042510  042101   MSG9:   .ASCIZ  <CR><LF>/HEAD NO./
044706  047040  027117     000
044713     015  005012  044527   MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
044720  046114  052040  051505
044726  020124  051104  053111
044734  051505  000072
044740  005015  050012  053517   MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
044746  051105  052440  020120
044754  042522  052123  051101
044762  020124  047524  052040
044770  051505  020124  006461
044776  000012
045000  005015  052012  042510   MSG12:  .ASCII  <CR><LF><LF>/THE ABOVE OFFSET FAILURES ARE NOT ERRORS/
045006  040440  047502  042526
045014  047440  043106  042523
045022  020124  040506  046111
045030  051125  051505  040440
045036  042522  047040  052117
045044  042440  051122  051117
045052     123
045053     015  041012  052125           .ASCIZ  <CR><LF>/BUT INDICATORS OF SURFACE,HEAD, & ELECTRONICS QUALITY/<CR><LF>
045060  044440  042116  041511
045066  052101  051117  020123
045074  043117  051440  051125
045102  040506  042503  044054
045110  040505  026104  023040
045116  042440  042514  052103
045124  047522  044516  051503
045132  050440  040525  044514
045140  054524  005015     000
045145     015  047012  020117   MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
045152  020114  051117  050040
045160  041440  047514  045503
045166  020123  051120  051505
045174  047105     124
045177     015  044012  040505           .ASCIZ  <CR><LF>/HEAD SWITCHING TIME TEST BYPASSED/
```

```
045204  020104  053523  052111
045212  044103  047111  020107
045220  044524  042515  052040
045226  051505  020124  054502
045234  040520  051523  042105
045242     000
045243     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
045250  051501  044523  043516
045256  042040  044522  042526
045264  000040
045266  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
045274  042526  000040
045300  005015  051104  053111  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL NO. /
045306  020105  042523  044522
045314  046101  047040  027117
045322  000040
045324  005015  040503  052122  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
045332  044522  043504  020105
045340  042523  044522  046101
045346  047040  027117  000040
045354  005015  040412  047502  MSG26:  .ASCIZ  <CR><LF><LF>/ABORTING BALANCE OF TESTS ON THIS DRIVE/<CR><LF><LF>
045362  052122  047111  020107
045370  040502  040514  041516
045376  020105  043117  052040
045404  051505  051524  047440
045412  020116  044124  051511
045420  042040  044522  042526
045426  005015  000012
045432  005015  040412  046114  MSG27:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
045440  042040  044522  042526
045446  020123  042524  052123
045454  042105  005015  000012
045462  005015  047516  053440  MSG37:  .ASCIZ  <CR><LF>/NO WRITE CHECK ERROR AT MAX POSITIVE OFFSET/
045470  044522  042524  041440
045476  042510  045503  042440
045504  051122  051117  040440
045512  020124  040515  020130
045520  047520  044523  044524
045526  042526  047440  043106
045534  042523  000124
045540  005015  047516  053440  MSG38:  .ASCIZ  <CR><LF>/NO WRITE CHECK ERROR AT MAX NEGATIVE OFFSET/<CR><LF>
045546  044522  042524  041440
045554  042510  045503  042440
045562  051122  051117  040440
045570  020124  040515  020130
045576  042516  040507  044524
045604  042526  047440  043106
045612  042523  006524  000012
045620  005015  051127  052111  MSG39:  .ASCIZ  <CR><LF>/WRITE CHECK FAILURE AT OFFSET =/
045626  020105  044103  041505
045634  020113  040506  046111
045642  051125  020105  052101
045650  047440  043106  042523
045656  020124  000075
045662  005015  047503  046125  MSG40:  .ASCII  <CR><LF>/COULD NOT READ BAD SECTOR INFO ON CYL 410/
045670  020104  047516  020124
```

```
 045676  042522  042101  041040
 045704  042101  051440  041505
 045712  047524  020122  047111
 045720  047506  047440  020116
 045726  054503  020114  030464
 045734     060
 045735     015  047412  020122          .ASCIZ  <CR><LF>/OR ALIGNMENT CARTRIDGE USED/<CR><LF>
 045742  046101  043511  046516
 045750  047105  020124  040503
 045756  052122  044522  043504
 045764  020105  051525  042105
 045772  005015     000
 045775     015  050012  047522  MSG74:  .ASCIZ  <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
 046002  051107  046501  040440
 046010  047502  052122  050040
 046016  047105  044504  043516
 046024  027056  050056  042514
 046032  051501  020105  040527
 046040  052111     000
 046043     015  044012  046101  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING...PLEASE WAIT/
 046050  020124  042520  042116
 046056  047111  027107  027056
 046064  046120  040505  042523
 046072  053440  044501  000124
 046100  005015  051120  043517  MSG76:  .ASCIZ  <CR><LF>/PROGRAM ABORTED/
 046106  040522  020115  041101
 046114  051117  042524  000104
 046122  005015  050103  020125  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
 046130  040510  052114  042105
 046136     000

                                  .SBTTL  ERROR MESSAGES

 046137     015  042412  051122  EM1:    .ASCIZ  <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
 046144  051117  020054  047117
 046152  054514  030040  052040
 046160  051110  020125  020067
 046166  046101  047514  042527
 046174  026104  052040  054522
 046202  040440  040507  047111
 046210  005015     000
 046213     104  044522  042526  EM2:    .ASCIZ  /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
 046220  021440  044440  020116
 046226  045522  051503  020062
 046234  040503  047116  052117
 046242  041040  020105  042522
 046250  042101  041040  041501
 046256  020113  047503  051122
 046264  041505  046124  020131
 046272  047111  051040  046513
 046300  031122     000
 046303     015  040412  047502  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
 046310  052122  052040  051505
 046316  051524  027056  052456
 046324  042516  050130  041505
 046332  042524  020104  044524
```

```
046340  042515  047440  052125
046346  040440  020124  041520
046354  000075
046356  005015  041101  051117  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
046364  020124  042524  052123
046372  027123  027056  047125
046400  054105  042520  052103
046406  042105  044440  052116
046414  051105  052522  052120
046422  040440  020124  041520
046430  000075
046432  042115  020123  042523  EM5:    .ASCIZ  /MDS SET IN RKCS2/
046440  020124  047111  051040
046446  041513  031123     000
046453     125  042506  051440  EM6:    .ASCIZ  /UFE SET IN RKCS2/
046460  052105  044440  020116
046466  045522  051503  000062
046474  051104  020101  047111  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
046502  051040  042113  020123
046510  020046  042516  020104
046516  047111  051040  041513
046524  031123  051040  051505
046532  052105  020073  051127
046540  047117  020107  047520
046546  052122  051440  046105
046554  041505  042524  037504
046562     000
046563     104  044522  042526  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
046570  050040  042522  042523
046576  052116  041040  052125
046604  047040  052117  051440
046612  042520  044503  044506
046620  042105  041040  020131
046626  050117  051105  052101
046634  051117     000
046637     104  044522  042526  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
046644  047040  052117  050040
046652  042522  042523  052116
046660  041040  052125  051440
046666  042520  044503  044506
046674  042105  041040  020131
046702  050117  051105  052101
046710  051117     000
046713     101  047502  052122  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
046720  052040  051505  051524
046726  027056  041456  047101
046734  047516  020124  042522
046742  042506  042522  041516
046750  020105  047503  052116
046756  047522  046114  051105
046764  051040  043505  051511
046772  042524  000122
046776  051104  020101  047111  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
047004  051040  042113  020123
047012  020046  042516  020104
047020  047111  051040  041513
```

```
047026  031123  041040  052117
047034  020110  042523  000124
047042  047503  052116  047522   EM12:   .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
047050  046114  051105  047040
047056  052117  051040  040505
047064  054504  044440  020116
047072  045522  051503  000061
047100  047516  040440  052124   EM13:   .ASCIZ  /NO ATTN IN RKASOF/
047106  020116  047111  051040
047114  040513  047523  000106
047122  047125  054105  042520   EM14:   .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
047130  052103  042105  046440
047136  046505  051117  020131
047144  040520  044522  054524
047152  052040  040522  000120
047160  045522  041504  023040   EM15:   .ASCII  /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
047166  051040  042113  020101
047174  047111  044504  040503
047202  042524  052040  040510
047210  020124  041527  020105
047216  041517  052503  051122
047224  042105  040440     124
047231     015  041412  046131           .ASCIZ  <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
047236  032040  030461  020054
047244  051124  041501  020113
047252  026062  051440  041505
047260  047524  020122  030462
047266     000
047267     103  047101  047516   EM16:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
047274  020124  042522  042101
047302  041040  042101  051440
047310  041505  047524  020122
047316  047111  047506  046522
047324  052101  047511  000116
047332  042515  051523  043501   EM17:   .ASCIZ  /MESSAGE A0 ERROR/
047340  020105  030101  042440
047346  051122  051117     000
047353     115  051505  040523   EM18:   .ASCIZ  /MESSAGE B0 ERROR/
047360  042507  041040  020060
047366  051105  047522  000122
047374  042515  051523  043501   EM19:   .ASCIZ  /MESSAGE A1 ERROR/
047402  020105  030501  042440
047410  051122  051117     000
047415     115  051505  040523   EM20:   .ASCIZ  /MESSAGE B1 ERROR/
047422  042507  041040  020061
047430  051105  047522  000122
047436  042503  051122  051440   EM21:   .ASCIZ  /CERR SET IN RKCS1/
047444  052105  044440  020116
047452  045522  051503  000061
047460  047516  042040  044522   EM22:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/<CR><LF>
047466  042526  020123  047506
047474  047125  020104  047111
047502  042040  053105  041511
047510  020105  040515  020120
047516  022050  042504  046526
047524  006451     012
```

```
047527     123  052105  050125           .ASCIZ  /SETUP CORRECTLY AND RESTART/<CR><LF>
047534  041440  051117  042522
047542  052103  054514  040440
047550  042116  051040  051505
047556  040524  052122  005015
047564     000
047565     116  020117  051104  EM23:    .ASCII  /NO DRIVES FOUND ON BUSS/<CR><LF>
047572  053111  051505  043040
047600  052517  042116  047440
047606  020116  052502  051523
047614  005015
047616  042523  052524  020120           .ASCIZ  /SETUP CORRECTLY AND PRESS 'CONTINUE'/<CR><LF>
047624  047503  051122  041505
047632  046124  020131  047101
047640  020104  051120  051505
047646  020123  041447  047117
047654  044524  052516  023505
047662  005015     000
047665     126  046117  053040  EM24:    .ASCIZ  /VOL VALID NOT SET IN RKMR2/
047672  046101  042111  047040
047700  052117  051440  052105
047706  044440  020116  045522
047714  051115  000062
047720  005015  042504  042524  EM25:    .ASCIZ  <CR><LF>/DETECTED 10 BAD SECTORS...ABORTING TEST/
047726  052103  042105  030440
047734  020060  040502  020104
047742  042523  052103  051117
047750  027123  027056  041101
047756  051117  044524  043516
047764  052040  051505  000124
047772  042504  042524  052103  EM26:    .ASCIZ  /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
050000  042105  041040  042523
050006  041040  052125  047040
050014  052117  046040  051511
050022  042524  020104  047111
050030  041040  042101  051440
050036  041505  047524  020122
050044  044506  042514     000
050051     104  052105  041505  EM27:    .ASCII  /DETECTED BSE IN READ COMMAND/
050056  042524  020104  051502
050064  020105  047111  051040
050072  040505  020104  047503
050100  046515  047101     104
050105     015  041012  052125           .ASCIZ  <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
050112  047040  052117  044440
050120  020116  051120  053105
050126  047511  051525  053440
050134  044522  042524  041440
050142  046517  040515  042116
050150  052040  020117  040523
050156  042515  051440  041505
050164  047524  000122
050170  054503  020114  042101  EM36:    .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
050176  051104  044440  020116
050204  045522  051115  020063
050212  047516  020124  040523
```

```
050220 042515 040440 020123
050226 045522 041504    000
050233    103 046131 042040  EM39:  .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
050240 043111 020106 020046
050246 043117 051506 052105
050254 044440 020116 045522
050262 051115 020062 047516
050270 020124 046103 040505
050276 042522 000104
050302 054503 020114 042101  EM40:  .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
050310 051104 044440 020116
050316 045522 051115 020063
050324 047516 020124 046103
050332 040505 042522 000104
050340 054503 020114 042101  EM41:  .ASCIZ  /CYL ADDR IN B2 DID NOT REMAIN CLEARED/
050346 051104 044440 020116
050354 031102 042040 042111
050362 047040 052117 051040
050370 046505 044501 020116
050376 046103 040505 042522
050404 000104
050406 052101 047124 047040  EM55:  .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
050414 052117 041440 042514
050422 051101 042105 044440
050430 020116 045522 051501
050436 043117    000
050441    104 052114 051440  EM63:  .ASCIZ  /DLT SET IN RKCS2/
050446 052105 044440 020116
050454 045522 051503 000062
050462 042522 042101 044040  EM65:  .ASCIZ  /READ HEADER ERROR/
050470 040505 042504 020122
050476 051105 047522 000122
050504 046101 043511 046516  EM69:  .ASCIZ  /ALIGNMENT CARTRIDGE USED/
050512 047105 020124 040503
050520 052122 044522 043504
050526 020105 051525 042105
050534    000
050535    103 047524 051440  EM73:  .ASCIZ  /CTO SET IN RKCS1/
050542 052105 044440 020116
050550 045522 051503 000061
050556 052122 020132 047516  EM74:  .ASCIZ  /RTZ NOT SET IN RKMR2/
050564 020124 042523 020124
050572 047111 051040 046513
050600 031122    000
050603    116 042105 051440  EM79:  .ASCIZ  /NED SET IN RKCS2/
050610 052105 044440 020116
050616 045522 051503 000062
050624 051127 052111 020105  EM80:  .ASCIZ  /WRITE CHECK ERROR SET IN RKCS2/
050632 044103 041505 020113
050640 051105 047522 020122
050646 042523 020124 047111
050654 051040 041513 031123
050662    000
050663    127 044522 042524  EM81:  .ASCIZ  /WRITE CHECK COMMAND NOT FUNCTIONING/
050670 041440 042510 045503
050676 041440 046517 040515
```

```
050704 042116 047040 052117
050712 043040 047125 052103
050720 047511 044516 043516
050726    000
050727    122 040505 020104   EM82:  .ASCIZ  /READ DATA DID NOT COMPARE WITH WRITE DATA/
050734 040504 040524 042040
050742 042111 047040 052117
050750 041440 046517 040520
050756 042522 053440 052111
050764 020110 051127 052111
050772 020105 040504 040524
051000    000
051001    104 052101 020101   EM83:  .ASCIZ  /DATA CHECK ERROR SET IN RKER/
051006 044103 041505 020113
051014 051105 047522 020122
051022 042523 020124 047111
051030 051040 042513 000122
051036 044127 046111 020105   EM84:  .ASCIZ  /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
051044 040527 052111 047111
051052 020107 047506 020122
051060 047503 052116 020122
051066 042522 042101 020131
051074 051117 040440 052106
051102 051105 041440 047117
051110 051124 051040 040505
051116 054504 051040 041505
051124 042047    000
051127    117 043106 042523   EM85:  .ASCIZ  /OFFSET STATUS BIT IN RKMR2 CLEARED/
051134 020124 052123 052101
051142 051525 041040 052111
051150 044440 020116 045522
051156 051115 020062 046103
051164 040505 042522 000104
051172 043117 051506 052105   EM86:  .ASCIZ  /OFFSET REG IN A2 NOT = RKASOF/
051200 051040 043505 044440
051206 020116 031101 047040
051214 052117 036440 051040
051222 040513 047523 000106
051230 044504 020104 047516   EM88:  .ASCIZ  /DID NOT FIND SECTOR 0 FROM INDEX/
051236 020124 044506 042116
051244 051440 041505 047524
051252 020122 020060 051106
051260 046517 044440 042116
051266 054105    000
051271    122 040505 044504   EM93:  .ASCIZ  /READING WRONG CYLINDER # IN HEADER...MISPOSITION/
051276 043516 053440 047522
051304 043516 041440 046131
051312 047111 042504 020122
051320 020043 047111 044040
051326 040505 042504 027122
051334 027056 044515 050123
051342 051517 052111 047511
051350 000116
051352 043117 051506 052105   EM94:  .ASCIZ  /OFFSET IT IN A2 NOT CLEARED/
051360 044440 020124 047111
051366 040440 020062 047516
```

```
051374  020124  046103  040505
051402  042522  000104
051406  047506  046522  052101   EM95:   .ASCIZ  /FORMAT BIT NOT SET IN RKMR2/
051414  041040  052111  047040
051422  052117  051440  052105
051430  044440  020116  045522
051436  051115  000062
051442  040503  047116  052117   EM96:   .ASCIZ  /CANNOT FIND SECTOR 23(8)/
051450  043040  047111  020104
051456  042523  052103  051117
051464  031040  024063  024470
051472     000
051473     110  040505  020104   EM97:   .ASCIZ  /HEAD SWITCHING REQ'D ANOTHER FULL REVOLUTION OF DISK/
051500  053523  052111  044103
051506  047111  020107  042522
051514  023521  020104  047101
051522  052117  042510  020122
051530  052506  046114  051040
051536  053105  046117  052125
051544  047511  020116  043117
051552  042040  051511  000113
051560  051104  053111  020105   EM100:  .ASCIZ  /DRIVE OFF TRACK SET IN RKMR3/
051566  043117  020106  051124
051574  041501  020113  042523
051602  020124  047111  051040
051610  046513  031522     000
051615     104  042111  047040   EM101:  .ASCIZ  /DID NOT GO TO CYLINDER 10/
051622  052117  043440  020117
051630  047524  041440  046131
051636  047111  042504  020122
051644  030061     000

                                 .SBTTL  DATA HEADERS

051647     124  051505  020124   DH1:    .ASCIZ  /TEST NO.  PC/
051654  047516  020056  050040
051662  000103
051664  045522  051115  004461   DH2:    .ASCIZ  /RKMR1  RKMR2  RKMR3  RKER   RKDS   RKCS1  RKCS2/
051672  045522  051115  004462
051700  045522  051115  004463
051706  045522  051105  051011
051714  042113  004523  045522
051722  051503  004461  045522
051730  051503  000062
051734  045522  041527  051011   DH3:    .ASCIZ  /RKWC   RKBA   RKDA   RKASOF RKDC   RKECPS RKECPT/
051742  041113  004501  045522
051750  040504  051011  040513
051756  047523  004506  045522
051764  041504  051011  042513
051772  050103  004523  045522
052000  041505  052120     000
052005     106  047522  020115   DH6:    .ASCIZ  /FROM CYL  TO CYL  CYL DIFF/
052012  054503  020114  052040
052020  020117  054503  020114
052026  041440  046131  042040
052034  043111  000106
```

```
052040  042524  052123  047040  DH8:    .ASCIZ  /TEST NO.       TRAP PC/
052046  027117  052011  040522
052054  020120  041520     000
052061     101  052106  051105  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
052066  051440  040524  052122
052074  051440  044520  042116
052102  042514  041440  046517
052110  040515  042116  051040
052116  041505  042047  041040
052124  020131  051104  053111
052132  000105
052134  052101  042440  042116  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
052142  047440  020106  042510
052150  042101  046040  040517
052156  044504  043516     000
052163     105  050130  041505  DH11:   .ASCIZ  /EXPECTED       WAS/
052170  042524  004504  040527
052176  000123
052200  047117  051440  041505  DH13:   .ASCIZ  /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
052206  047524  051522  030440
052214  026060  030440  026062
052222  030440  026064  030440
052230  026066  030440  020070
052236  051117  031040  020060
052244  054503  020114  030464
052252  020060  051124  041501
052260  020113  000062
052264  047117  051440  041505  DH14:   .ASCIZ  /ON SECTORS 11, 13, 15, 17, 19 OR 21 CYL 410 TRACK 2/
052272  047524  051522  030440
052300  026061  030440  026063
052306  030440  026065  030440
052314  026067  030440  020071
052322  051117  031040  020061
052330  054503  020114  030464
052336  020060  051124  041501
052344  020113  000062
052350  043101  042524  020122  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
052356  042522  040503  020114
052364  047503  046515  047101
052372  000104
052374  043101  042524  020122  DH19:   .ASCIZ  /AFTER PACK COMMAND/
052402  040520  045503  041440
052410  046517  040515  042116
052416     000
052417     101  052106  051105  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
052424  051440  046105  041505
052432  020124  051104  053111
052440  020105  047503  046515
052446  047101  000104
052452  043101  042524  020122  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
052460  052523  051502  051531
052466  042524  020115  046103
052474  040505  000122
052500  043101  042524  020122  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
052506  051104  053111  020105
052514  046103  040505  020122
```

```
052522 047503 046515 047101
052530 000104
052532 043101 042524 020122  DH24:   .ASCIZ  /AFTER OFFSET COMMAND/
052540 043117 051506 052105
052546 041440 046517 040515
052554 042116    000
052557    101 052106 051105  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
052564 051440 042505 020113
052572 047503 046515 047101
052600 000104
052602 043101 042524 020122  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
052610 042522 042101 042040
052616 052101 020101 047503
052624 046515 047101 000104
052632 043101 042524 020122  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
052640 051127 052111 020105
052646 040504 040524 041440
052654 046517 040515 042116
052662    000
052663    011 054105 042520  DH28:   .ASCIZ  /       EXPECT/
052670 052103    000
052673    011 041501 052524  DH29:   .ASCIZ  /       ACTUAL/
052700 046101    000
052703    101 052106 051105  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
052710 051040 040505 020104
052716 042510 042101 051105
052724 041440 046517 040515
052732 042116    000
052735    101 004460 030102  DH31:   .ASCIZ  /A0     B0      A1      B1      A2      B2      B3/
052742 040411 004461 030502
052750 040411 004462 031102
052756 041011 000063
052762 043101 042524 020122  DH32:   .ASCIZ  /AFTER WRITE CHECK COMMAND/
052770 051127 052111 020105
052776 044103 041505 020113
053004 047503 046515 047101
053012 000104
053014 043101 042524 020122  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
053022 051127 052111 020105
053030 042510 042101 051105
053036 041440 046517 040515
053044 042116    000
053047    127 051117 021504  DH40:   .ASCIZ  /WORD#  HEADER WAS  SHOULD BE/
053054 044011 040505 042504
053062 020122 040527 020123
053070 051440 047510 046125
053076 020104 042502    000
053103    104 051125 047111  DH41:   .ASCIZ  /DURING RECAL COMMAND/
053110 020107 042522 040503
053116 020114 047503 046515
053124 047101 000104
053130 047117 051440 041505  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
053136 047524 051522 030040
053144 031054 032054 033054
053152 047440 020122 020070
053160 041440 046131 032040
```

```
053166  030061  052040  040522
053174  045503  031040     000
053201     117  020116  042523   DH43:   .ASCIZ  /ON SECTORS 1,3,5,7 OR 9  CYL 410 TRACK 2/
053206  052103  051117  020123
053214  026061  026063  026065
053222  020067  051117  034440
053230  020040  054503  020114
053236  030464  020060  051124
053244  041501  020113  000062
053252  047506  046522  052101   DH44:   .ASCIZ  /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
053260  023040  040440  046114
053266  051040  040505  026504
053274  051127  052111  020105
053302  042524  052123  020123
053310  044527  046114  041040
053316  020105  054502  040520
053324  051523  042105     000
053331     101  052106  051105   DH47:   .ASCIZ  /AFTER READ HEADER COMMAND WITH MOVEMENT/
053336  051040  040505  020104
053344  042510  042101  051105
053352  041440  046517  040515
053360  042116  053440  052111
053366  020110  047515  042526
053374  042515  052116     000
053401     115  043523  040440   DH49:   .ASCIZ  /MSG A & B IN RKMR2 & RKMR3 RESP. ARE INVALID/
053406  023040  041040  044440
053414  020116  045522  051115
053422  020062  020046  045522
053430  051115  020063  042522
053436  050123  020056  051101
053444  020105  047111  040526
053452  044514  000104
053456  043101  042524  020122   DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
053464  042523  045505  052040
053472  020117  042523  043114
053500  041440  046517  040515
053506  042116     000
053511     127  052111  020110   DH52:   .ASCIZ  /WITH INTENTIONAL MISCOMPARE/
053516  047111  042524  052116
053524  047511  040516  020114
053532  044515  041523  046517
053540  040520  042522     000
053545     104  051125  047111   DH53:   .ASCIZ  /DURING OFFSET COMMAND/
053552  020107  043117  051506
053560  052105  041440  046517
053566  040515  042116     000
053573     101  052106  051105   DH54:   .ASCIZ  /AFTER FORMAT CHANGE AND CONTR READY REC'D/
053600  043040  051117  040515
053606  020124  044103  047101
053614  042507  040440  042116
053622  041440  047117  051124
053630  051040  040505  054504
053636  051040  041505  042047
053644     000
053645     103  046131  021440   DH56:   .ASCIZ  /CYL #  HEADER WORD 0/
053652  044011  040505  042504
```

```
053660  020122  047527  042122
053666  030040     000
053671     101  052106  051105   DH57:   .ASCIZ  /AFTER WRITE COMMAND WITH OFFSET/
053676  053440  044522  042524
053704  041440  046517  040515
053712  042116  053440  052111
053720  020110  043117  051506
053726  052105     000

                                 .SBTTL  ERROR OUTPUT DATA

        053732                           .EVEN
053732  001214  001116           DT1:    $TESTN,$ERRPC
053736  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
053744  007370  007366  007354
053752  007356
053754  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
053762  007372  007374  007406
053770  007410
053772  001214  001334           DT3:    $TESTN,TRAPPC
053776  001214  001116  001344   DT4:    $TESTN,$ERRPC,FRCYL,TOCYL,CALDIF
054004  001346  001354
054010  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054016  007370  007366  007354
054024  007356
054026  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054034  007372  007374  007406
054042  007410
054044  001214  001116  001454   DT6:    $TESTN,$ERRPC,WD2,WD1
054052  001452
054054  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054062  007370  007366  007354
054070  007356
054072  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054100  007372  007374  007406
054106  007410
054110  001214  001116  001472   DT7:    $TESTN,$ERRPC,WDCNT,HDWD,TEMP1
054116  001510  007412
054122  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054130  007370  007366  007354
054136  007356
054140  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054146  007372  007374  007406
054154  007410
054156  001214  001116  001346   DT8:    $TESTN,$ERRPC,TOCYL,FRCYL,CALDIF
054164  001344  001354
054170  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054176  007370  007366  007354
054204  007356
054206  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054214  007372  007374  007406
054222  007410
054224  001214  001116  001346   DT9:    $TESTN,$ERRPC,TOCYL,RHTAB
054232  001726
054234  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054242  007370  007366  007354
```

```
054250  007356
054252  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054260  007372  007374  007406
054266  007410
054270  001214  001116  007444   DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
054276  007446  007450  007452
054304  007424  007426  007430
054312  007432
054314  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054322  007370  007366  007354
054330  007356
054332  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054340  007372  007374  007406
054346  007410
054350  001214  001116  007444   DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
054356  007446  007450  007452
054364  007454  007456
054370  007424  007426  007430           H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
054376  007432  007434  007436
054404  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054412  007370  007366  007354
054420  007356
054422  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054430  007372  007374  007406
054436  007410
054440  001214  001116  007444   DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
054446  007446  007450  007452
054454  007454  007456  007462
054462  007424  007426  007430           H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
054470  007432  007434  007436
054476  007442
054500  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
054506  007370  007366  007354
054514  007356
054516  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
054524  007372  007374  007406
054532  007410


                                 .SBTTL  ERROR DATA FORMATS

054534  000003                   DF1:    3
054536     002     000                   .BYTE 2,0
054540  051664                           DH2
054542     007     000                   .BYTE 7,0
054544  051734                           DH3
054546     007     000                   .BYTE   7,0

054550  000001                   DF2:    1
054552     002     000                   .BYTE   2,0

054554  000005                   DF3:    5
054556     000     000                   .BYTE   0,0
054560  051647                           DH1
054562     002     000                   .BYTE   2,0
054564  052163                           DH11
054566     002     000                   .BYTE   2,0
```

```
054570  051664                          DH2
054572     007      000                 .BYTE   7,0
054574  051734                          DH3
054576     007      000                 .BYTE   7,0

054600  000003                 DF4:     3
054602     002      000                 .BYTE   2,0
054604  051664                          DH2
054606     007      000                 .BYTE   7,0
054610  051734                          DH3
054612     007      000                 .BYTE   7,0

054614  000005                 DF5:     5
054616     000      000                 .BYTE   0,0
054620  053401                          DH49
054622     000      000                 .BYTE   0,0
054624  051647                          DH1
054626     002      000                 .BYTE   2,0
054630  051664                          DH2
054632     007      000                 .BYTE   7,0
054634  051734                          DH3
054636     007      000                 .BYTE   7,0

054640  000005                 DF6:     5
054642     000      000                 .BYTE 0,0
054644  051647                          DH1
054646     002      000                 .BYTE 2,0
054650  052005                          DH6
054652     003      000                 .BYTE 3,0
054654  051664                          DH2
054656     007      000                 .BYTE   7,0
054660  051734                          DH3
054662     007      000                 .BYTE   7,0


054664  000004                 DF10:    4
054666     000      000                 .BYTE 0,0
054670  051647                          DH1
054672     002      000                 .BYTE 2,0
054674  051664                          DH2
054676     007      000                 .BYTE 7,0
054700  051734                          DH3
054702     007      000                 .BYTE   7,0

054704  000004                 DF14:    4
054706     002      000                 .BYTE   2,0
054710  053047                          DH40
054712     003      000                 .BYTE   3,0
054714  051664                          DH2
054716     007      000                 .BYTE   7,0
054720  051734                          DH3
054722     007      000                 .BYTE   7,0


054724  000004                 DF15:    4
054726     000      000                 .BYTE   0,0
```

```
054730  051647                          DH1
054732     002     000                  .BYTE   2,0
054734  051664                          DH2
054736     007     000                  .BYTE   7,0
054740  051734                          DH3
054742     007     000                  .BYTE   7,0

054744  000005                 DF17:    5
054746     000     000                  .BYTE   0,0
054750  053252                          DH44
054752     000     000                  .BYTE   0,0
054754  051647                          DH1
054756     002     000                  .BYTE   2,0
054760  051664                          DH2
054762     007     000                  .BYTE   7,0
054764  051734                          DH3
054766     007     000                  .BYTE   7,0
054770  000005                 DF20:    5
054772     000     000                  .BYTE   0,0
054774  051647                          DH1
054776     002     000                  .BYTE   2,0
055000  053645                          DH56
055002     002     000                  .BYTE   2,0
055004  051664                          DH2
055006     007     000                  .BYTE   7,0
055010  051734                          DH3
055012     007     000                  .BYTE   7,0

055014  000007                 DF21:    7
055016     000     000                  .BYTE   0,0
055020  051647                          DH1
055022     002     000                  .BYTE   2,0
055024  052663                          DH28
055026     000     000                  .BYTE   0,0
055030  052735                          DH31
055032     004     000                  .BYTE   4,0
055034  052673                          DH29
055036     004     000                  .BYTE   4,0
055040  051664                          DH2
055042     007     000                  .BYTE   7,0
055044  051734                          DH3
055046     007     000                  .BYTE   7,0

055050  000007                 DF22:    7
055052     000     000                  .BYTE   0,0
055054  051647                          DH1
055056     002     000                  .BYTE   2,0
055060  052663                          DH28
055062     000     000                  .BYTE   0,0
055064  052735                          DH31
055066     006     000                  .BYTE   6,0
055070  052673                          DH29
055072     006     000                  .BYTE   6,0
055074  051664                          DH2
055076     007     000                  .BYTE   7,0
055100  051734                          DH3
```

```
      055102     007     000              .BYTE   7,0

      055104  000007            DF23:     7
      055106     000     000              .BYTE   0,0
      055110  051647                      DH1
      055112     002     000              .BYTE   2,0
      055114  052663                      DH28
      055116     000     000              .BYTE   0,0
      055120  052735                      DH31
      055122     007     000              .BYTE   7,0
      055124  052673                      DH29
      055126     007     000              .BYTE   7,0
      055130  051664                      DH2
      055132     007     000              .BYTE   7,0
      055134  051734                      DH3
      055136     007     000              .BYTE   7,0
                                .EVEN
                                ;;***************************************************************
                                .SBTTL  TYPE ERROR ROUTINE
                                ;*ENTRY JSR PC,TYP ERR
                                ;*RETURN RTS PC
                                ;*
                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
                                ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
                                ;*THE ERROR.
                                ;;***************************************************************
      055140  104413            TYPERR:   SAVREG
      055142  113700  001114              MOVB    $ITEMB,R0          ;ENTER ERROR NUMBER
      055146  042700  177400              BIC     #177400,R0         ;CLEAR SIGN EXTENSION
      055152  005300                      DEC     R0                 ;FORM INDEX FOR ERROR TABLE
      055154  006300                      ASL     R0
      055156  006300                      ASL     R0
      055160  006300                      ASL     R0
      055162  062700  007526    1$:       ADD     #$ERRTB,R0         ;FORM ADDRESS OF ERROR ENTRY
      055166  012037  055202              MOV     (R0)+,2$           ;GET EM POINTER
      055172  001404                      BEQ     3$                 ;BRANCH IF THERE ISN'T ONE
      055174  104401  001205              TYPE    ,$CRLF             ;TYPE CARRIAGE RETURN LINE FEED
      055200  104401                      TYPE                       ;TYPE ERROR MESSAGE (EM)
      055202  000000            2$:       .WORD   0                  ;EM POINTER GOES HERE
      055204  012037  055220    3$:       MOV     (R0)+,4$           ;GET DH POINTER
      055210  001404                      BEQ     5$                 ;BRANCH IF THERE ISN'T ONE
      055212  104401  001205              TYPE    ,$CRLF             ;TYPE CR-LF
      055216  104401                      TYPE                       ;TYPE DATA HEADER
      055220  000000            4$:       .WORD   0                  ;DH POINTER GOES HERE
      055222  012001            5$:       MOV     (R0)+,R1           ;GET DT POINTER
      055224  001455                      BEQ     20$                ;BRANCH IF THERE ARE NONE
      055226  005004                      CLR     R4                 ;SET INDENT SWITCH
      055230  012000                      MOV     (R0)+,R0           ;GET DF POINTER
      055232  012002                      MOV     (R0)+,R2           ;STORE NUMBER OF DH'S
      055234  001446                      BEQ     17$                ;DH NUM IS 0-BRANCH
      055236  005104                      COM     R4                 ;NO INDENT
      055240  104401  001205              TYPE    ,$CRLF
      055244  112003            10$:      MOVB    (R0)+,R3           ;GET & STORE NUMBER OF DATA WORDS
      055246  105720                      TSTB    (R0)+              ;BUMP PAST FORMAT WORD
      055250  005703                      TST     R3                 ;TEST IF ANY DATA FOR THIS HEADER
```

```
055252  001407                                  BEQ     14$                     ;NO - SKIP DATA PRINT
055254  013146                          11$:    MOV     @(R1)+,-(SP)            ;PUT FIRST DATA WORD ON STACK
055256  104402                                  TYPOC                           ;TYPE IT
055260  005303                                  DEC     R3                      ;MORE DATA WORDS
055262  001403                                  BEQ     14$                     ;NO-BRANCH
055264  104401  055414                          TYPE    ,SPACE2                 ;TYPE SEPARATORS
055270  000771                                  BR      11$                     ;LOOP
055272  005302                          14$:    DEC     R2                      ;MORE DH'S?
055274  003431                                  BLE     20$                     ;NO-BRANCH
055276  104401  001205                          TYPE    ,$CRLF
055302  005760  000002                          TST     2(R0)                   ;ONLY A DH IN THIS REQUEST?
055306  001404                                  BEQ     15$                     ;YES-BRANCH BYPASS INDENT
055310  005104                                  COM     R4                      ;INDENT?
055312  001002                                  BNE     15$                     ;NO-BRANCH
055314  104401  055414                          TYPE    ,SPACE2                 ;YES-TYPE SPACES
055320  012037  055326                  15$:    MOV     (R0)+,16$               ;GET NEXT DH POINTER
055324  104401                                  TYPE                            ;TYPE DH
055326  000000                          16$:    .WORD   0                       ;DH POINTER GOES HERE
055330  105710                                  TSTB    (R0)                    ;TYPE A DT?
055332  001003                                  BNE     21$                     ;YES-BRANCH
055334  062700  000002                          ADD     #2,R0                   ;INCREMENT DF POINTER
055340  000754                                  BR      14$                     ;SEE IF END OF DF BLOCK
055342  104401  001205                  21$:    TYPE    ,$CRLF
055346  005704                                  TST     R4                      ;INDENT?
055350  001335                                  BNE     10$                     ;NO-BRANCH
055352  104401  055414                  17$:    TYPE    ,SPACE2                 ;YES-TYPE SPACES
055356  000732                                  BR      10$                     ;LOOP
055360  104414                          20$:    RESREG

055362  032777  010000  123550                  BIT     #SW12,@SWR              ;SEE IF ABORT DRV AFTER 20 ERRORS
055370  001410                                  BEQ     25$                     ;BR IF NO
055372  023727  001103  000024                  CMP     $ERFLG,#20.             ;ELSE SEE IF HAVE 20 ERRORS
055400  001004                                  BNE     25$                     ;BR IF NO
055402  012706  001100                          MOV     #STACK,SP               ;ELSE RESTORE STACK PTR
055406  000137  031452                          JMP     $EOP                    ;AND GO TO NEXT DRV

055412  000207                          25$:    RTS     PC
055414  020040     000                  SPACE2: .ASCIZ/  /                      ;2 SPACES
                                        ;  ODT-11 -- V005A

                                        ; DEC-11-UODPA-A-LA

                                        ; COPYRIGHT 1969,1970,1972
                                        ; DIGITAL EQUIPMENT CORPORATION
                                        ; MAYNARD, MASSACHUSETTS 01754
                                                .ENABL  ABS,AMA
        055420                                  .EVEN
        055500                                  .=.+60
        000000                          R0      =       %0      ; REGISTER
        000001                          R1      =       %1      ;  NAMING
        000002                          R2      =       %2      ;   CONVENTIONS
        000003                          R3      =       %3
        000004                          R4      =       %4
        000005                          R5      =       %5
        000006                          SP      =       %6
        000007                          PC      =       %7
```

```
        177776                          ST      =       177776          ;STATUS REGISTER
                                        ;
        000014                          O.TVEC  =       14              ;TRT VECTOR LOCATION
        000340                          O.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
        000020                          O.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
        000003                          TRT     =       000003          ;TRT INSTRUCTION
        000006                          RTT     =       000006          ;RTT INSTRUCTION
                                        ;
                                        ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
                                        ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
                                        ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
                                        ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
                                        ;
        177562                          O.RDB   =       177562  ;R DATA BUFFER
        177560                          O.RCSR  =       177560  ;R C/SR
        177566                          O.TDB   =       177566  ;T DATA BUFFER
        177564                          O.TCSR  =       177564  ;T C/SR
                                        ;
                                        ;
                                        ; INITIALIZE ODT
                                        ;  USE O.ODT FOR A NORMAL ENTRY
                                        ;  USE O.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
                                        ;  USE O.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
                                        ;
055500  000413                          O.ODT:  BR      O.STRT          ;NORMAL ENTRY
055502  000417                                  BR      O.RST           ;RESTART
055504  013737  177776  055460          O.ENTR: MOV     ST,O.UST        ;RE-ENTER  --  SAVE STATUS
055512  013737  000016  177776                  MOV     O.TVEC+2,ST     ;SET UP LOCAL STATUS
055520  010737  055456                          MOV     PC,O.UPC        ;FAKE THE PC
055524  000137  056656                          JMP     O.BK1
                                        ;
055530  012706  055440                  O.STRT: MOV     #O.URO,SP       ;SET UP STACK
055534  010637  055454                          MOV     SP,O.USP        ;FAKE THE SAVED STACK
055540  000414                                  BR      O.RST1          ;CLEAR BREAKPOINT TABLES
055542  004037  057064                  O.RST:  JSR     0,O.SVR         ;SAVE REGISTERS
055546  013777  055476  177716                  MOV     O.UIN,@O.ADR1   ;REMOVE THE BREAKPOINT
055554  113704  055462                          MOVB    O.PRI,R4        ;GET ODT PRIORITY
055560  106004                                  RORB    R4              ;SHIFT
055562  106004                                  RORB    R4              ; INTO
055564  106004                                  RORB    R4              ;  POSITION
055566  110437  177776                          MOVB    R4,ST           ;STORE IN STATUS
055572  000127                          O.RST1: JMP     (PC)+
055574  000403                                  BR      O.45
055576  012737  000002  056566                  MOV     #RTI,O.RTIT     ;SET TO RTI IF 11/20 OR /05
055604  105037  057505                  O.45:   CLRB    O.P             ;DISALLOW PROCEED
055610  012737  000340  000016                  MOV     #O.STM,O.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
055616  012737  056646  000014                  MOV     #O.BRK,O.TVEC   ;PC TO TRT VECTOR
055624  000447                                  BR      O.RALL          ;CLEAR BREAKPOINT TABLES
                                        ;
                                        ; SPECIAL NAME HANDLER
                                        ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URO
                                        ;
055626  004537  057306                  O.REGT: JSR     5,O.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
055632  012704  057531                          MOV     #O.TL,R4        ;TABLE START ADDRESS
055636  120024                          O.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
055640  001413                                  BEQ     O.SP            ;JUMP IF YES
```

```
055642  022704  057537              CMP     #O.TL+O.LG,R4   ;IS THE SEARCH DONE?
055646  101373                      BHI     O.RSP           ;BRANCH IF NOT
055650  042700  177770              BIC     #177770,R0      ;MASK OFF OCTAL
055654  010004                      MOV     R0,R4
055656  006304              O.SP1:  ASL     R4
055660  062704  055440              ADD     #O.UR0,R4       ;GENERATE ADDRESS
055664  005202                      INC     R2              ;SET FOUND FLAG
055666  000444                      BR      O.SCAN          ;GO FIND NEXT CHARACTER
055670  162704  057522      O.SP:   SUB     #O.TL-7,R4      ;CORRECT CONSTANT
055674  000770                      BR      O.SP1
                            ;
                            ; ← HANDLER - OPEN INDEXED ON THE PC
                            ;
055676  004737  057432      O.ORPC: JSR     PC,O.TCLS
055702  010502                      MOV     R5,R2           ;CURRENT ADDRESS IN R2
055704  061202                      ADD     @R2,R2          ;COMPUTE
055706  006202                      ASR     R2              ;MOVE ONE BIT TO CARRY
055710  103421                      BCS     O.ERR           ;ERROR IF ODD NUMBER
055712  006302                      ASL     R2              ;RESTORE WORD
055714  005722                      TST     (R2)+           ; AND INCREMENT BY TWO
055716  010205                      MOV     R2,R5           ;UPDATE CAD
055720  000137  056172              JMP     O.OP2           ;GO FINISH UP
                            ;
                            ; B HANDLER - SET AND REMOVE BREAKPOINTS
                            ;
055724  005702              O.BKPT: TST     R2              ;IF NO NUMBER TYPED
055726  001406                      BEQ     O.RALL          ;  REMOVE BREAKPOINT
055730  006204                      ASR     R4              ;CHECK IF ODD
055732  103410                      BCS     O.ERR           ;JUMP IF ODD
055734  006304                      ASL     R4              ;RESTORE ONE BIT
055736  010437  055472              MOV     R4,O.ADR1       ;SET A BREAKPOINT
055742  000412                      BR      O.DCD
055744  012737  057546  055472  O.RALL: MOV #O.TRTC,O.ADR1  ;CLEAR BREAKPOINT
055752  000406                      BR      O.DCD
                            ;
                            ; COMMAND DECODER - ODT11
                            ;
                            ;    REGISTERS R0-R4 MAY BE USED,
                            ;     REGISTER R5 WILL BE CONSIDERED SAFE
                            ;
055754  052705  000001      O.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
055760  012700  000077              MOV     #'?,R0          ;  ? TO BE TYPED
055764  004537  057364              JSR     5,O.FTYP        ; OUTPUT ?
055770  004537  057464      O.DCD:  JSR     5,O.CRLS        ;TYPE  <CR><LF>*
055774  005004              O.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
055776  005002                      CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
056000  004537  057306      O.SCAN: JSR     5,O.GET         ;GET A CHAR, RETURN IN R0
056004  022700  000060              CMP     #'0,R0          ;COMPARE WITH ASCII 0
056010  101013                      BHI     O.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
056012  022700  000067              CMP     #'7,R0          ;COMPARE WITH ASCII 7
056016  103410                      BLO     O.CLGL          ;CHECK LEGALITY IF NOT OCTAL
056020  042700  177770              BIC     #177770,R0      ;CONVERT TO BCD
056024  006304                      ASL     R4              ; MAKE ROOM
056026  006304                      ASL     R4              ;  IN
056030  006304                      ASL     R4              ;    R4
056032  060004                      ADD     R0,R4           ;PACK THREE BITS IN R4
```

```
056034  005202                  INC     R2              ;R2 HAS NUMERIC FLAG
056036  000760                  BR      O.SCAN          ;    AND TRY AGAIN
056040  005001          O.CLGL: CLR     R1              ;CLEAR INDEX
056042  120061  057515  O.LGL1: CMPB    R0,O.LGCH(R1)   ;DO THE CODES MATCH?
056046  001405                  BEQ     O.LGL2          ;JUMP IF YES
056050  005201                  INC     R1              ; SET INDEX FOR NEXT SEARCH
056052  020127  000014          CMP     R1,#O.CLGT      ;IS THE SEARCH DONE?
056056  103336                  BHIS    O.ERR           ;      OOPS!
056060  000770                  BR      O.LGL1          ;RE-LOOP
056062  006301          O.LGL2: ASL     R1              ;MULTIPLY BY TWO
056064  000171  056070          JMP     @O.LGDR(R1)     ;GO TO PROPER ROUTINE
                        ;
056070  056120          O.LGDR: O.WRD   ;  /    OPEN WORD
056072  056152                  O.CRET  ;  CARRIAGE RETURN    CLOSE
056074  055626                  O.REGT  ;  $  REGISTER OPS
056076  056462                  O.GO    ;  G  GO TO ADDRESS K
056100  056164                  O.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
056102  055676                  O.ORPC  ;  ←  OPEN RELATED, INDEX - PC
056104  056216                  O.BACK  ;  ↑  OPEN PREVIOUS
056106  056226                  O.OFST  ;  O  OFFSET
056110  056304                  O.WSCH  ;  W  SEARCH WORD
056112  056300                  O.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
056114  055724                  O.BKPT  ;  B  BREAKPOINTS
056116  056570                  O.PROC  ;  P  PROCEED
        000030          O.LGL   =       .-O.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS

                        ;
                        ; PROCESS / - OPEN WORD
                        ;
056120  005702          O.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
056122  001410                  BEQ     O.WRDA          ;SKIP OTHERWISE
056124  010405                  MOV     R4,R5           ;    PUT VALUE IN CAD
056126  006205          O.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
056130  103711          O.ERR2: BCS     O.ERR           ;JUMP IF ODD ADDRESS
056132  006305                  ASL     R5              ;RESTORE THE CARRY BIT
056134  011500                  MOV     @R5,R0          ;GET CONTENTS OF WORD
056136  004537  057222          JSR     5,O.CADV        ;GO GET AND TYPE OUT @CAD
056142  000714                  BR      O.DCD1          ;GO BACK TO DECODER
056144  042705  000001  O.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
056150  000766                  BR      O.WRD1          ;GO BACK TO MAIN-LINE

                        ;
                        ; PROCESS CARRIAGE RETURN
                        ;
056152  004737  057432  O.CRET: JSR     PC,O.TCLS       ;CLOSE LOCATION
056156  052705  000001          BIS     #1,R5           ;CLOSE EVERYTHING
056162  000702                  BR      O.DCD           ;RETURN TO DECODER

                        ;
                        ; PROCESS <LF>, OPEN NEXT WORD
                        ;
056164  004737  057432  O.OP1:  JSR     PC,O.TCLS       ;CLOSE PRESENT CELL
056170  005725                  TST     (R5)+           ;GENERATE NEW ADDRESS
056172  004537  057456  O.OP2:  JSR     5,O.CRLF        ;<CR><LF>
056176  010500                  MOV     R5,R0           ;NUMBER TO TYPE
056200  004537  057222          JSR     5,O.CADV        ; TYPE OUT ADDRESS
056204  012700  000057          MOV     #'/,R0          ;TYPE A /
056210  004537  057364          JSR     5,O.FTYP
056214  000744                  BR      O.WRD1          ;GO PROCESS IT
```

```
                                ;
                                ;       PROCESS ↑, OPEN PREVIOUS WORD
                                ;
056216  004737  057432          O.BACK: JSR     PC,O.TCLS
056222  005745                          TST     -(R5)           ;GENERATE NEW ADDRESS
056224  000762                          BR      O.OP2           ;GO DO THE REST
                                ;
                                ;       PROCESS O, COMPUTE OFFSET
                                ;
056226  006205                  O.OFST: ASR     R5              ;GET LOW ORDER BIT
056230  103737                          BCS     O.ERR2          ;ERROR IF CLOSED
056232  006305                          ASL     R5              ;RESTORE WORD
056234  012700  000040                  MOV     #' ,R0          ;TYPE ONE BLANK
056240  004537  057364                  JSR     5,O.FTYP        ; AS A SEPARATOR
056244  160504                          SUB     R5,R4           ;COMPUTE
056246  005304                          DEC     R4
056250  005304                          DEC     R4              ;       16 BIT OFFSET
056252  010400                          MOV     R4,R0           ;TYPE A
056254  010402                          MOV     R4,R2           ;SAVE R4
056256  004537  057222                  JSR     5,O.CADV        ;NUMBER IN R0 - WORD MODE
056262  010200                          MOV     R2,R0
056264  006200                          ASR     R0              ;DIVIDE BY TWO
056266  103402                          BCS     O.OF1           ;BRANCH IF ODD
056270  004537  057222                  JSR     5,O.CADV        ;NUMBER IN R0 - BYTE MODE
056274  000137  055774          O.OF1:  JMP     O.DCD1          ;ALL DONE
                                ;
                                ;       SEARCHES - $MSK   HAS THE MASK
                                ;                  $MSK+2 HAS THE FWA
                                ;                  $MSK+4 HAS THE LWA
```

```
                                  ;
056300  005201            O.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
056302  000401                    BR      O.WDS
056304  005001            O.WSCH: CLR     R1              ;SET WORD SEARCH
056306  005702            O.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
056310  001621            O.ERR1: BEQ     O.ERR           ;ERROR IF NO OBJECT
056312  013702  055466            MOV     O.MSK+2,R2      ;SET ORIGIN
056316  013705  055464            MOV     O.MSK,R5        ;SET MASK
056322  005105                    COM     R5              ;AND COMPLEMENT IT
056324  020237  055470    O.WDS2: CMP     R2,O.MSK+4      ; IS THE SEARCH ALL DONE?
056330  101217                    BHI     O.DCD           ;  YES
056332  011200                    MOV     @R2,R0          ; GET OBJECT
056334  005701                    TST     R1              ;NO
056336  001027                    BNE     O.EFF1          ;BRANCH IF EFFECTIVE SEARCH
056340  010046                    MOV     R0,-(SP)
056342  010403                    MOV     R4,R3           ;EXCLUSIVE OR
056344  040400                    BIC     R4,R0           ; IS DONE
056346  042603                    BIC     (SP)+,R3        ;       IN A VERY
056350  050003                    BIS     R0,R3           ;     FANCY MANNER HERE
056352  040503                    BIC     R5,R3           ;AND RESULT WITH MASK
056354  001016            O.WDS3: BNE     O.WDS4          ;RE-LOOP IF NO MATCH
056356  010446                    MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
056360  004537  057456            JSR     5,O.CRLF        ;TYPE <CR,LF>
056364  010200                    MOV     R2,R0           ;GET READY TO TYPE
056366  004537  057222            JSR     5,O.CADV        ;  TYPE ADDRESS
056372  012700  000057            MOV     #'/,R0          ;SLASH TO R0
056376  004537  057364            JSR     5,O.FTYP        ;TYPE IT
056402  011200                    MOV     @R2,R0          ;GET CONTENTS
056404  004537  057222            JSR     5,O.CADV        ;TYPE CONTENTS
056410  012604                    MOV     (SP)+,R4        ; RESTORE R4
056412  005722            O.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
056414  000743                    BR      O.WDS2          ;      RETURN
056416  020004            O.EFF1: CMP     R0,R4           ; IS (X)=K?
056420  001755                    BEQ     O.WDS3          ;TYPE IF EQUAL
056422  010003                    MOV     R0,R3           ;(X) TO R3
056424  060203                    ADD     R2,R3           ;(X)+X
056426  005203                    INC     R3
056430  005203                    INC     R3              ;(X)+X+2
056432  020304                    CMP     R3,R4           ;IS (X)+X+2=K?
056434  001747                    BEQ     O.WDS3          ;BRANCH IF EQUAL
056436  042700  177400            BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
056442  110000                    MOVB    R0,R0           ;EXTEND SIGN
056444  000257                    CCC
056446  006300                    ASL     R0              ;MULTIPLY BY TWO
056450  005200                    INC     R0              ;ADD TWO
056452  005200                    INC     R0
056454  060200                    ADD     R2,R0           ;ADD PC
056456  020004                    CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
056460  000735                    BR      O.WDS3

                                  ;
                                  ; PROCESS G - GO
                                  ;
056462  105037  057505    O.GO:   CLRB    O.P             ;DISALLOW PROCEED
056466  006204                    ASR     R4              ;CHECK LOW ORDER BIT
056470  103617                    BCS     O.ERR2          ;ERROR IF ODD NUMBER
056472  006304                    ASL     R4              ;RESTORE WORD
```

```
056474  010437  055456                  MOV     R4,O.UPC                ;SET UP NEW PC
056500  112737  000340  177776          MOVB    #O.STM,ST               ;SET HIGH PRIORITY
056506  004537  057154                  JSR     5,O.RSTT                ;RESTORE TELETYPE
056512  105037  057504          O.TBIT: CLRB    O.T                     ;CLEAR BOTH
056516  042737  000020  055460          BIC     #O.TBT,O.UST            ;  T-BIT FLAGS
056524  017737  176742  055476          MOV     @O.ADR1,O.UIN           ;SAVE INSTRUCTION
056532  013777  057546  176732          MOV     O.TRTC,@O.ADR1          ;REPLACE WITH TRAP
056540  012600                  O.GO2:  MOV     (SP)+,RO                ;RESTORE
056542  012601                          MOV     (SP)+,R1                ; RO
056544  012602                          MOV     (SP)+,R2                ;   THRU
056546  012603                          MOV     (SP)+,R3                ;
056550  012604                          MOV     (SP)+,R4                ;
056552  012605                          MOV     (SP)+,R5                ;    R5
056554  012606                          MOV     (SP)+,SP                ; AND SP
056556  013746  055460                  MOV     O.UST,-(SP)             ;    AND STATUS
056562  013746  055456                  MOV     O.UPC,-(SP)             ;    AND PC
056566  000006                  O.RTIT: RTT                             ;CHANGED TO RTI FOR 11/20 AND /05

                                ; PROCESS P - PROCEED
                                ;   ONLY ALLOWED AFTER A BREAKPOINT
                                ;
056570  105737  057505          O.PROC: TSTB    O.P                     ;CHECK LEGALITY OF PROCEED
056574  001645                          BEQ     O.ERR1                  ;NOT LEGAL
056576  105037  057505                  CLRB    O.P                     ;CLEAR PROCEED FLAG
056602  005702                          TST     R2                      ;WAS COUNT SPECIFIED?
056604  001402                          BEQ     O.PR1                   ;NO
056606  010437  055474                  MOV     R4,O.CT                 ;YES, PUT AWAY COUNT
056612  112737  000340  177776  O.PR1:  MOVB    #O.STM,ST               ;FORCE HIGH PRIORITY
056620  004537  057154                  JSR     5,O.RSTT                ;RESTORE TTY
056624  112737  000340  177776  O.C1:   MOVB    #O.STM,ST               ;SET HIGH PRIORITY
056632  105237  057504                  INCB    O.T                     ;SET T-BIT FLAG
056636  052737  000020  055460          BIS     #O.TBT,O.UST            ;SET T-BIT
056644  000735                          BR      O.GO2

                                ; BREAKPOINT HANDLER
                                ;  A TRT BREAKPOINT CAUSES O.BRK TO BE ENTERED, WHICH SAVES
                                ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
                                ;  AND GIVES CONTROL TO THE COMMAND DECODER
                                ;
056646  012637  055456          O.BRK:  MOV     (SP)+,O.UPC             ;PRIORITY IS 7 UPON ENTRY
056652  012637  055460                  MOV     (SP)+,O.UST             ;SAVE STATUS AND PC
056656  004037  057064          O.BK1:  JSR     0,O.SVR                 ;SAVE VARIOUS REGISTERS
056662  105737  057504                  TSTB    O.T                     ;CHECK FOR T-BIT SET
056666  001311                          BNE     O.TBIT                  ;JUMP IF SET
056670  013777  055476  176574          MOV     O.UIN,@O.ADR1           ;REMOVE BREAKPOINTS
056676  105737  055462                  TSTB    O.PRI                   ;CHECK IF PRIORITY
056702  100003                          BPL     O.BK2                   ; IS AS SAME AS USER PGM
056704  113705  055460                  MOVB    O.UST,R5                ;PICK UP USER UST IF SO
056710  000407                          BR      O.BK3                   ;AND DON'T COMPUTE THE PRIORITY
056712  113705  055462          O.BK2:  MOVB    O.PRI,R5                ;OTHERWISE PICK UP ACTUAL PRIORITY
056716  000257                          CCC                             ;CLEAR CARRY
056720  106005                          RORB    R5                      ;SHIFT LOW ORDER BITS
056722  106005                          RORB    R5                      ;  INTO
056724  106005                          RORB    R5                      ;    HIGH ORDER
056726  106005                          RORB    R5                      ;      POSITION
056730  110537  177776          O.BK3:  MOVB    R5,ST                   ;PUT THE STATUS AWAY WHERE IT BELONGS
```

```
056734  013705  055456                MOV     0.UPC,R5                ;GET PC, IT POINTS TO THE TRT
056740  005745                        TST     -(R5)                   ;SUBTRACT TWO
056742  010537  055456                MOV     R5,0.UPC                ;FROM THE USER'S PC
056746  020537  055472                CMP     R5,0.ADR1               ;COMPARE WITH LIST
056752  001417                        BEQ     0.B2                    ;JUMP IF FOUND
056754  004537  057122                JSR     5,0.SVTT                ;SAVE TELETYPE STATUS
056760  004537  057456                JSR     5,0.CRLF
056764  012704  057510                MOV     #0.BD,R4                ;ERROR, NOTHING FOUND
056770  012703  057511                MOV     #0.BD+1,R3
056774  004537  057350                JSR     5,0.TYPE                ;OUTPUT "BE" FOR BAD ENTRY
057000  010500                        MOV     R5,R0
057002  042737  000020  055460        BIC     #0.TBT,0.UST            ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
057010  000420                        BR      0.B3                    ; AND CONTINUE
057012  005337  055474        0.B2:   DEC     0.CT
057016  003302                        BGT     0.C1                    ;JUMP IF REPEAT
057020  012737  000001  055474        MOV     #1,0.CT                 ;RESET COUNT TO 1
057026  105237  057505                INCB    0.P                     ;ALLOW PROCEED
057032  004537  057122                JSR     5,0.SVTT                ;SAVE TELETYPE STATUS, R4 IS SAFE
057036  012700  000102                MOV     #'B,R0
057042  004537  057364                JSR     5,0.FTYP                ;TYPE "B"
057046  013700  055472                MOV     0.ADR1,R0               ;GET ADDRESS OF BREAK
057052  004537  057222        0.B3:   JSR     5,0.CADV                ;TYPE ADDRESS
057056  005005                        CLR     R5                      ;CLEAR CAD
057060  000137  055770                JMP     0.DCD                   ;GO TO DECODER

                              ;
                              ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
                              ;
057064  012637  057502        0.SVR:  MOV     (SP)+,0.XXX             ;PICK REGISTER FROM STACK AND SAVE
057070  010637  055454                MOV     SP,0.USP                ;SAVE USER STACK ADDRESS
057074  012706  055454                MOV     #0.USP,SP               ;SET TO INTERNAL STACK
057100  010546                        MOV     R5,-(SP)                ;SAVE
057102  010446                        MOV     R4,-(SP)                ; REGISTERS
057104  010346                        MOV     R3,-(SP)                ;1
057106  010246                        MOV     R2,-(SP)                ; THRU
057110  010146                        MOV     R1,-(SP)                ;    5
057112  013746  057502                MOV     0.XXX,-(SP)             ;PUT SAVED REGISTER ON STACK
057116  005746                        TST     -(SP)
057120  000200                        RTS     R0

                              ;
                              ; SAVE TELETYPE STATUS
                              ;
057122  113737  177560  057506 0.SVTT: MOVB   0.RCSR,0.CSR1           ;SAVE R C/SR
057130  113737  177564  057507        MOVB    0.TCSR,0.CSR2           ;SAVE T C/SR
057136  105037  177560                CLRB    0.RCSR                  ;CLEAR ENABLE AND MAINTENANCE
057142  105037  177564                CLRB    0.TCSR                  ;  BITS IN BOTH C/SR
057146  004537  057456                JSR     5,0.CRLF                ;TYPE <CR,LF>
057152  000205                        RTS     R5

                              ;
                              ; RESTORE TELETYPE STATUS
                              ;
057154  004537  057456        0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
057160  105737  177564                TSTB    0.TCSR                  ;WAIT READY ON PRINTER
057164  100375                        BPL     .-4
057166  032737  004000  177560        BIT     #4000,0.RCSR            ;CHECK BUSY FLAG ON READER
057174  001403                        BEQ     0.RSE1                  ;SKIP READY LOOP IF NOT BUSY
057176  105737  177560                TSTB    0.RCSR                  ;WAIT READY
```

```
057202  100375                          BPL     .-4                     ;       ON READER
057204  113737  057506  177560  O.RSE1: MOVB    O.CSR1,O.RCSR           ;RESTORE
057212  113737  057507  177564          MOVB    O.CSR2,O.TCSR           ;  THE STATUS REGISTERS
057220  000205                          RTS     R5

                                ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
                                ; WORD IS IN R0

057222  010246                  O.CADV: MOV     R2,-(SP)                ;SAVE R2
057224  012704  057545                  MOV     #O.BUF+6,R4             ;BUFFER START ADDRESS
057230  012746  000060                  MOV     #'0,-(SP)                       ;CONSTANT ASCII 0
057234  010002                  O.SPC:  MOV     R0,R2                   ; GET
057236  042702  177770                  BIC     #177770,R2              ;       OCTAL CHARACTER
057242  061602                          ADD     @SP,R2                  ;CONVERT TO ASCII
057244  110244                          MOVB    R2,-(R4)                ;STORE IN BUFFER
057246  006200                          ASR     R0                      ;SHIFT THIS MESS
057250  006200                          ASR     R0                      ; RIGHT
057252  006200                          ASR     R0                      ;    THREE WHOLE PLACES
057254  020427  057540                  CMP     R4,#O.BUF+1             ;DONE?
057260  101365                          BHI     O.SPC                   ;       NO
057262  042700  177776                  BIC     #177776,R0              ;GET LAST BIT
057266  062600                          ADD     (SP)+,R0                        ;CONVERT TO ASCII
057270  110044                          MOVB    R0,-(R4)                ;AND PUT IT AWAY
057272  012703  057545                  MOV     #O.BUF+6,R3             ;LWA
057276  004537  057350                  JSR     5,O.TYPE                ;TYPE WHOLE STRING OF CHARACTERS
057302  012602                          MOV     (SP)+,R2                ;RESTORE R2
057304  000205                          RTS     R5

                                ; GENERAL CHARACTER INPUT ROUTINE
                                ; CHARACTER INPUT GOES TO R0

057306  105737  177560          O.GET:  TSTB    O.RCSR                  ;WAIT FOR
057312  100375                          BPL     .-4                     ;     INPUT FROM KEYBOARD
057314  113700  177562                  MOVB    O.RDB,R0                ;GET A CHARACTER
057320  004537  057364                  JSR     5,O.FTYP                ;ECHO CHARACTER
057324  042700  177600                  BIC     #177600,R0              ;STRIP OFF PARITY FROM CHARACTER
057330  001766                          BEQ     O.GET                   ;IGNORE NULLS
057332  122700  000040                  CMPB    #40,R0                  ;CHECK FOR SPACES
057336  001763                          BEQ     O.GET                   ;IGNORE NULLS
057340  122700  000073                  CMPB    #';,R0                  ;CHECK FOR SEMI-COLON
057344  001760                          BEQ     O.GET                   ;IGNORE THEM IF FOUND
057346  000205                          RTS     R5

                                ; GENERAL CHARACTER OUTPUT ROUTINE
                                ;  ADDRESS OF FIRST BYTE IN R4,
                                ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)

057350  020304                  O.TYPE: CMP     R3,R4                   ;CHECK FOR COMPLETION
057352  103426                          BLO     O.TYP1                  ; EXIT WHEN DONE
057354  112400                          MOVB    (R4)+,R0                ;GET A CHARACTER
057356  004537  057364                  JSR     5,O.FTYP                ;TYPE ONE CHARACTER
057362  000772                          BR      O.TYPE                  ;LOOP UNTIL DONE

                                ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)

057364  105737  177564          O.FTYP: TSTB    O.TCSR                  ;CHECK STATUS
```

```
057370  100375                          BPL     .-4                     ;WAIT UNTIL READY
057372  110037  177566                  MOVB    R0,O.TDB                ;TYPE ONE CHARACTER
057376  120037  000045                  CMPB    R0,@#45                 ;IS CHAR TO BE FILLED?
057402  001012                          BNE     O.TYP1                  ;NO
057404  113746  000044                  MOVB    @#44,-(SP)              ;YES, INIT THE COUNT
057410  105737  177564          O.TYP2: TSTB    O.TCSR
057414  100375                          BPL     O.TYP2
057416  105037  177566                  CLRB    O.TDB                   ;GENERATE NULL FILLER
057422  105316                          DECB    @SP
057424  003371                          BGT     O.TYP2
057426  005726                          TST     (SP)+                   ;POP STACK
057430  000205                  O.TYP1: RTS     R5
                                ;
                                ;       CLOSE WORD OR BYTE AND EXIT,
                                ;       UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
                                ;
057432  006205                  O.TCLS: ASR     R5                      ;GET LOW ORDER BIT
057434  103405                          BCS     O.TC                    ;JUMP IF ALREADY CLOSED
057436  006305                          ASL     R5
057440  005702                          TST     R2                      ;IF NO NUMBER WAS TYPED THERE IS
057442  001401                          BEQ     O.CLS1                  ;NO CHANGE TO THE OPEN CELL
057444  010415                          MOV     R4,@R5                  ;STORE WORD
057446  000207                  O.CLS1: RTS     PC
057450  005746                  O.TC:   TST     -(SP)                   ;POP EXTRA CELL FROM STACK
057452  000137  055754                  JMP     O.ERR                   ;AND SCREAM BLOODY MURDER
                                ;
                                ;       O.CRLF - TYPE <CR,LF>
                                ;       O.CRLS - TYPE <CR,LF>*
                                ;
057456  012703  057513          O.CRLF: MOV     #O.CR+1,R3              ;LWA <CR,LF>
057462  000402                          BR      O.CRS
057464  012703  057514          O.CRLS: MOV     #O.CR+2,R3              ;LWA <CR,LF>*
057470  012704  057512          O.CRS:  MOV     #O.CR,R4                ;FWA
057474  004537  057350                  JSR     5,O.TYPE                ;TYPE SOMETHING
057500  000205                          RTS     R5
                                ;
057502  000000                  O.XXX:  .WORD   0                       ;TEMPORARY STORAGE
057504     000                  O.T:    .BYTE   0                       ;  T-BIT FLAG
057505     000                  O.P:    .BYTE   0                       ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
                                                                        ;             = 1 IF PROCEED ALLOWED
057506     000                  O.CSR1: .BYTE   0                       ;SAVE CELL - R C/SR
057507     000                  O.CSR2: .BYTE   0                       ;SAVE CELL - T C/SR
                                ;
                                        .EVEN
057510  042502                  O.BD:   .WORD   "BE
                                ;
057512     015                  O.CR:   .BYTE   015             ;       <CR>
057513     012                          .BYTE   012             ;       <LF>
057514     052                          .BYTE   '*              ;       *
                                ;
057515     057                  O.LGCH: .BYTE   '/              ;       /
057516     015                          .BYTE   015             ;       CARRIAGE RETURN
057517     044                          .BYTE   '$              ;       $
057520     107                          .BYTE   'G              ;       G
057521     012                          .BYTE   012             ;       <LF>
057522     137                          .BYTE   '←              ;       ←
```

```
057523     136                 .BYTE    '↑      ;  ↑
057524     117                 .BYTE    'O      ;  O
057525     127                 .BYTE    'W      ;  W
057526     105                 .BYTE    'E      ;  E
057527     102                 .BYTE    'B      ;  B
057530     120                 .BYTE    'P      ;  P
        000014       O.CLGT  =        .-O.LGCH                      ;TABLE LENGTH

057531     123       O.TL:     .BYTE    'S      ;DO        1
057532     120                 .BYTE    'P      ;NOT       2
057533     115                 .BYTE    'M      ;CHANGE    3
057534     000                 .BYTE    0       ;THE       4
057535     000                 .BYTE    0       ;ORDER     5
057536     102                 .BYTE    'B      ;HERE      6
        000006       O.LG      =        .-O.TL
                     ;
057537               O.BUF:                             ;6 CHAR. BUFFER WITH
        057545       .         =        .+6
057545     040                 .BYTE    ;                ;TRAILING BLANK
                               .EVEN
                     ;
057546  000003       O.TRTC: TRT                        ;TRACE TRAP PROTOTYPE
                     ;
                     ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
                     ;
        055440       .         =        O.ODT-40
055440  000000       O.URO:    0        ;USER R0
055442  000000                 0        ;     R1
055444  000000                 0        ;     R2
055446  000000                 0        ;     R3
055450  000000                 0        ;     R4
055452  000000                 0        ;     R5
055454  000000       O.USP:    0        ;USER SP
055456  000000       O.UPC:    0        ;USER PC
055460  000000       O.UST:    0        ;USER ST
055462  000007       O.PRI:    7        ;ODT PRIORITY
055464  000000       O.MSK:    0        ;MASK
055466  000000                 0        ;LOW LIMIT
055470  000000                 0        ;HIGH LIMIT
                     ;
                     ;  BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
                     ;     UIN = CONTENTS
                     ;
055472  000000       O.ADR1: 0
055474  000000       O.CT:   0
055476  000000       O.UIN:  0
        000001                 .END
```

```
ABASE = 177440      1889   1930   1944#
ACDW1 = 000000      1889   1932
ACDW2 = 000000      1889   1933
ACLO  = 000010      1245#
ACPUOP= 000000      1889   1904
ACT11   007470      2122#  3298*
ADDW0 = 000000      1889   1934
ADDW1 = 000000      1889   1935
ADDW10= 000000      1889
ADDW11= 000000      1889
ADDW12= 000000      1889
ADDW13= 000000      1889
ADDW14= 000000      1889
ADDW15= 000000      1889
ADDW2 = 000000      1889   1936
ADDW3 = 000000      1889   1937
ADDW4 = 000000      1889   1938
ADDW5 = 000000      1889   1939
ADDW6 = 000000      1889   1940
ADDW7 = 000000      1889   1941
ADDW8 = 000000      1889
ADDW9 = 000000      1889
ADEVCT= 000000      1889   1895
ADEVM = 000000      1889   1931
AENV  = 000000      1889   1900
AENVM = 000000      1889   1901
AFATAL= 000000      1889   1892
AMADR1= 000000      1889   1917
AMADR2= 000000      1889   1921
AMADR3= 000000      1889   1924
AMADR4= 000000      1889   1927
AMAMS1= 000000      1889   1911
AMAMS2= 000000      1889   1919
AMAMS3= 000000      1889   1922
AMAMS4= 000000      1889   1925
AMSGAD= 000000      1889   1897
AMSGLG= 000000      1889   1898
AMSGTY= 000000      1889   1891
AMTYP1= 000000      1889   1912
AMTYP2= 000000      1889   1920
AMTYP3= 000000      1889   1923
AMTYP4= 000000      1889   1926
APASS = 000000      1889   1894
APRIOR= 000000      1889
APTCSU= 000040      7279   7451#
APTENV= 000001      5976   7226   7272   7407   7449#
APTSIZ= 000200      3226   7448#
APTSPO= 000100      7274   7409   7450#
ASWREG= 000000      1889   1902
ATESTN= 000000      1889   1893
ATTN    007344      2055#  6183   6202   6228
AUNIT = 000000      1889   1896
AUSWR = 000000      1889   1903
AVECT1= 000000      1889   1928
AVECT2= 000000      1889   1929
BADHDR  007340      2045#  3306*  4020*  4151*  4418*  4608*  5708*  5923*  5926*  6935
```

```
BADTMO  036570         3380    6989#
BAI   = 000020         1208#   4162    4311    4449    4519    4921    4981    5106    5268    5344    5469    5522    5642
                       5776    5821
BA16  = 000400         1194#
BA17  = 001000         1195#
BIT0  = 000001         1137#   1191    1223    1242    2112    3421    6880
BIT00 = 000001         1127#   1137
BIT01 = 000002         1126#   1136
BIT02 = 000004         1125#   1135
BIT03 = 000010         1124#   1134
BIT04 = 000020         1123#   1133
BIT05 = 000040         1122#   1132
BIT06 = 000100         1121#   1131
BIT07 = 000200         1120#   1130
BIT08 = 000400         1119#   1129    7161
BIT09 = 001000         1118#   1128    7169    7237
BIT1  = 000002         1136#   1224    2113
BIT10 = 002000         1117#   1196    1214    1233    1265    1279    1293    1306    1320    7214
BIT11 = 004000         1116#   1197    1215    1234    1251    1266    1280    1294    1307    1321    7176
BIT12 = 010000         1115#   1198    1216    1235    1267    1281    1295    1308    1322
BIT13 = 020000         1114#   1199    1217    1236    1252    1268    1282    1296    1309    1323    7221
BIT14 = 040000         1113#   1200    1218    1237    1253    1269    1283    1297    1310    1324    1334    6682    6693
                       7147
BIT15 = 100000         1112#   1201    1202    1219    1238    1254    1270    1337    5710    5870    6678    6689
BIT2  = 000004         1135#   1225    1244    2114
BIT3  = 000010         1134#   1207    1226    1245
BIT4  = 000020         1133#   1208    1227    1246    1259    1287    1314
BIT5  = 000040         1132#   1209    1228    1247    1260    1274    1288    1301    1315
BIT6  = 000100         1131#   1192    1210    1229    1248    1261    1275    1289    1302    1316
BIT7  = 000200         1130#   1193    1211    1230    1249    1262    1276    1290    1303    1317    3411    4679    4723
                       4775    4872    5130
BIT8  = 000400         1129#   1194    1212    1231    1250    1263    1277    1291    1304    1318
BIT9  = 001000         1128#   1195    1213    1232    1264    1278    1292    1305    1319
BPTVEC= 000014         1144#
BSE   = 000200         1230#   4065    4172    4236    4462    4933    5278    5479
BSERR   001512         2028#   3929*   3934*   4011
BSE20H  002336         2037#   3950    6687    6792
BSE20S  004336         2039#   3959    6691    6796
BSE22H  003336         2038#   3870    3931    3968    6676    6781
BSE22S  005336         2040#   3941    6680    6785
BYP     033026         3497    3525    3529    3544    3603    3630    3634    3638    3642    6255#
BYPCER  001516         2030#   3408*   3655*   6370
CALADD  001362         1971#   4134*   4421*   4591*   5879*   6649    6655    6657    6711
CALDIF  001354         1968#   5768*   8923    8947
CCLR  = 100000         1202#   3835    3850    4075    4182    4472    4693    4703    4943    5095    5163    5178    5254
                       5288    5404    5419    5579    5594    6410    6418    6426    6434    6939    6954
CCYL    001350         1966#
CDT   = 002000         1196#   3512    3613    3694    3696
CERR  = 100000         1201#   3476    3587    3881    3986    4038    4062    4144    4169    4234    4320    4433    4459
                       4528    4601    4745    4802    4865    4930    4989    5275    5352    5476    5529    5661    5733
                       5755    5788    5833    5889    5913    6372    6467
CFMT  = 010000         1198#   4427    4430    4453    4456    4494    4522    4525    4557    6778
CHKFLG  001520         2031#   6265*   6348    6351    6355
CHKMSG  033042         3893    4087    4106    4194    4213    4260    4277    4339    4358    4392    4484    4511    4547
                       4574    4644    4665    4714    4766    4829    4955    4974    5008    5027    5066    5087    5225
                       5246    5300    5319    5371    5390    5494    5513    5548    5567    5673    5802    5847    6265#
```

```
CKCERR   033602    6133   6137   6148   6153   6370#
CKSWR  = 104407    7146   7210   7236   8111#
CLEAR  = 000005    1179#  3852   4695   5097   5180   5256   5421   5596   6956
CLKOF    036134    6851#
CLKON    036040    6829#
CLOCK    036100    3329   6840#
CLRFLG   031666    3247   5988#
COE    = 001000    1232#
COUNT    001366    1975#  6841*  6843*
CR     = 000015    1052#  7318   7328   8123   8131   8133   8137   8147   8156   8163   8171   8176   8181
                   8186   8192   8198   8204   8208   8214   8219   8226   8235   8240   8245   8251   8253
                   8257   8263   8271   8281   8286   8293   8297   8299   8303   8307   8315   8319   8327
                   8335   8341   8349   8355   8362   8367   8370   8376   8394   8402   8477   8504   8511
                   8517   8522   8534   8554
CRLF   = 000200    1053#  7289   7328
CTO    = 004000    1197#  6376
CYL      001422    1992#  5462
CYLADD   001360    1970#  4672   4846   5331   5808   5853   6525*  6526*  6527*  6528*  6529*  6530*  6548*
                   6549*  6550*  6551*  6552*  6553*
CYLDIF   001356    1969#  4676*  4682   4687   4720*  4726   4731   4772*  4778   4783   4850   5328   6490*
                   6491*  6492*  6493*  6494*  6495*  6496   6498*  6516*  6517*  6535*  6536*  6537*  6538*
                   6539*  6540*  6541   6543*
CYL7     001436    1999#  5460
DATA0    001474    2019#  4160   4161   4295   4371   5267   5343   5357
DATA01   001476    2020#  5641
DATA1    001500    2021#  4054   4299   4300   4312   4325   4448   4518   4533   4920   4980   4994   5105
                   5775   5820
DATCMD   032262    3878   4035   4059   4141   4166   4231   4317   4375   4428   4454   4523   4598   4800
                   4927   4986   5110   5272   5349   5473   5526   5730   5780   5825   5886   6113#
DCK    = 100000    1238#  4243
DCLO   = 000020    1246#
DCPAR  = 020000    1199#
DDISP  = 177570    1059#  1864   3214
DDPCH    007466    2121#  3294*  3484   3687
DDT    = 000400    1250#
DDUMP    007464    2120#  3260*  5988
DF1      054534    2162   2168   2174   2180   2191   2197   2203   2429   2469   2474   2514   8999#
DF10     054664    2208   2213   2218   2223   2228   2238   2248   2253   2263   2279   2299   2304   2339
                   2349   2354   2389   2419   2424   2464   2494   2499   2554   2559   2564   2569   2584
                   2609   2614   2639   2689   2704   2790   2810   2815   2865   2935   2970   3006   9050#
DF14     054704    2835   9059#
DF15     054724    2519   2800   2805   2825   2830   2940   9069#
DF17     054744    2945   2950   2960   2965   9078#
DF2      054550    2329   9006#
DF20     054770    2369   9088#
DF21     055014    2258   2268   2273   2284   2289   2294   2309   2314   2319   2324   2374   2379   2384
                   2394   2399   2404   2409   2414   2644   2649   2654   2659   2910   2915   3066   3071
                   3091   3096   3101   3106   3111   3116   3121   3126   3131   3136   3141   3146   3151
                   3156   9099#
DF22     055050    2334   2359   2364   2489   2504   2509   2544   2549   2574   2579   9114#
DF23     055104    9129#
DF3      054554    2233   2243   9009#
DF4      054600    2344   9020#
DF5      054614    2589   2594   2599   9027#
DF6      054640    2534   2539   2860   3001   9038#
DH1      051647    2160   2166   2172   2178   2189   2195   2201   2342   2427   2467   2472   2512   2833
```

```
                        8713#   9011    9031    9040    9052    9071    9082    9090    9101    9116    9131
DH10      052134        2462    2808    2813    8746#
DH11      052163        8750#   9013
DH13      052200        2943    8753#
DH14      052264        2948    8762#
DH17      052350        2357    2362    2387    2582    2908    2913    3129    3134    8771#
DH19      052374        2277    2552    8775#
DH2       051664        8716#   9001    9015    9022    9033    9044    9054    9063    9073    9084    9094    9109    9124
                        9139
DH20      052417        2557    8779#
DH21      052452        2261    2562    8784#
DH22      052500        2337    2547    2687    2702    3089    3094    3119    3124    8788#
DH24      052532        2297    2302    2317    2322    2332    2542    3064    3069    8793#
DH25      052557        2367    2607    2612    2637    2858    2863    2999    8797#
DH26      052602        2216    2221    2241    2246    2271    2282    2382    2392    2933    2938    8801#
DH27      052632        2206    2211    2256    2266    2352    2372    2377    2417    2492    2517    2532    2537    8805#
DH28      052663        8810#   9103    9118    9133
DH29      052673        8812#   9107    9122    9137
DH3       051734        8723#   9003    9017    9024    9035    9046    9056    9065    9075    9086    9096    9111    9126
                        9141
DH30      052703        2572    2788    2798    2803    3109    3114    3149    3154    8814#
DH31      052735        8819#   9105    9120    9135
DH32      052762        2226    2231    2251    2287    2292    2397    2402    2497    8823#
DH39      053014        2823    2828    3099    3104    3139    3144    8828#
DH40      053047        8833#   9061
DH41      053103        3004    8838#
DH42      053130        2958    8842#
DH43      053201        2963    8849#
DH44      053252        2968    8856#   9080
DH47      053331        2487    8864#
DH49      053401        8871#   9029
DH51      053456        2347    2577    2642    2647    2652    2657    8879#
DH52      053511        2236    8884#
DH53      053545        2307    2312    2407    2412    8889#
DH54      053573        2422    8893#
DH56      053645        8901#   9092
DH57      053671        2502    2507    8905#
DH6       052005        8730#   9042
DH8       052040        2327    8735#
DH9       052061        2567    8738#
DI      = 040000        1200#
DISPLA    001142        1864#   3214*   3222*   7191*   7213*
DISPRE    000174        1346#   3222
DLT     = 100000        1219#   4813
DLY       033010        3472    3583    5652    6246#   6249
DMD     = 000040        1260#
DOCMD     032224        3795    3838    3853    3979    4431    4457    4495    4526    4558    4636    4696    4738    4858
                        5058    5098    5166    5181    5217    5257    5407    5422    5582    5597    5748    5906    6102#
                        6400    6925    6942    6957
DOTIM     007522        2140#   3328*   3337*   5629
DPAT1     001502        2022#   5468    5521    5534
DPAT2     001504        2023#
DRA     = 000001        1242#   3519    3620
DRDY    = 000200        1249#
DRIVS     007474        2124#   3416*   3423*   3440    3456*   3486*   3505    3607*   3674    5943    6043*
DRIVO     007476        2129#   3310    3418    3458    3569    6026
```

```
DRIV1     007500        2130#
DRIV2     007502        2131#
DRIV3     007504        2132#
DRIV4     007506        2133#
DRIV5     007510        2134#
DRIV6     007512        2135#
DRIV7     007514        2136#
DROT   =  000040        1247#
DRPAR  =  000010        1226#
DRVMSK=   000007        1206#   3479    3590
DRVPTR    001342        1962#   3310*   3679    3697*
DSC    =  040000        1253#
DSWR   =  177570        1058#   1863    3213
DTE    =  010000        1235#
DTYE   =  000040        1228#   3510    3611
DT1       053732        2161    2167    2173    2179    2190    2196    2202    2207    2212    2217    2222    2227    2237
                        2247    2252    2262    2278    2298    2303    2338    2343    2348    2353    2388    2418    2423
                        2428    2463    2468    2473    2493    2498    2513    2518    2553    2558    2563    2568    2583
                        2588    2593    2598    2608    2613    2638    2688    2703    2789    2799    2804    2809    2814
                        2824    2829    2864    2934    2939    2944    2949    2959    2964    2969    3005    8915#
DT13      054270        2257    2267    2272    2283    2288    2293    2308    2313    2318    2323    2373    2378    2383
                        2393    2398    2403    2408    2413    2643    2648    2653    2658    2909    2914    3065    3070
                        3090    3095    3100    3105    3110    3115    3120    3125    3130    3135    3140    3145    3150
                        3155    8963#
DT14      054350        2333    2358    2363    2488    2503    2508    2543    2548    2573    2578    8973#
DT15      054440        8984#
DT3       053772        2328    8922#
DT4       053776        2533    2538    2859    8923#
DT6       054044        2232    2242    8931#
DT7       054110        2834    8939#
DT8       054156        3000    8947#
DT9       054224        2368    8955#
D.ACLO=   000100        1302#
D.BRHM=   000100        1289#   3887    4081    4100    4188    4207    4254    4271    4333    4352    4386    4478    4505
                        4541    4568    4641    4659    4711    4760    4823    4949    4968    5002    5021    5063    5081
                        5222    5240    5294    5313    5365    5384    5488    5507    5542    5561    5667    5796    5841
D.CART=   000400        1291#   3887    4081    4100    4188    4207    4254    4271    4333    4352    4386    4478    4505
                        4541    4568    4641    4659    4711    4760    4823    4949    4968    5002    5021    5063    5081
                        5222    5240    5294    5313    5365    5384    5488    5507    5542    5561    5667    5796    5841
D.DDT  =  000400        1277#   3718
D.DOOR=   000200        1290#   3887    4081    4100    4188    4207    4254    4271    4333    4352    4386    4478    4505
                        4541    4568    4641    4659    4711    4760    4823    4949    4968    5002    5021    5063    5081
                        5222    5240    5294    5313    5365    5384    5488    5507    5542    5561    5667    5796    5841
D.DRA  =  000040        1274#   3885    4079    4098    4186    4205    4252    4269    4331    4350    4384    4476    4503
                        4539    4566    4639    4657    4709    4758    4821    4947    4966    5000    5019    5061    5079
                        5220    5238    5292    5311    5363    5382    5486    5505    5540    5559    5665    5794    5839
D.DRDY=   000200        1276#   3885    4079    4098    4186    4205    4252    4269    4331    4350    4384    4476    4503
                        4539    4566    4657    4709    4758    4821    4947    4966    5000    5019    5079    5238    5292
                        5311    5363    5382    5486    5505    5540    5559    5665    5794    5839
D.DROT=   020000        1309#   5784    5829
D.DSC  =  040000        1283#   4657    4758    5079    5238
D.FLT  =  000200        1303#
D.FORM=   001000        1278#   4497    4560
D.FWD  =  002000        1293#
D.HDFL=   000200        1317#
D.HDHM=   000040        1288#   6605
```

```
D.IDAE= 000040     1301#
D.ILF  = 000400    1304#
D.LIMD= 020000     1323#
D.LOAD= 010000     1295#   6622
D.MHD  = 000400    1318#
D.NMOV= 010000     1322#
D.OFF  = 002000    1279#   4639   4657   4705   4709   4752   4758   4842   5061   5079   5220   5238
D.PAR  = 001000    1305#
D.PIP  = 020000    1282#   4639   5061   5220
D.PLO  = 004000    1321#
D.REV  = 004000    1294#
D.RTZ  = 020000    1296#   3843   5171   5412   5587   6947
D.SECT= 000020     1314#
D.SKI  = 002000    1306#
D.SPIN= 010000     1281#   3885   4079   4098   4186   4205   4252   4269   4331   4350   4384   4476   4503
                   4539    4566   4639   4657   4709   4758   4821   4947   4966   5000   5019   5061   5079
                   5220    5238   5292   5311   5363   5382   5486   5505   5540   5559   5665   5794   5839
D.SPLS= 010000     1308#
D.SPOK= 001000     1292#   3887   4081   4100   4188   4207   4254   4271   4333   4352   4386   4478   4505
                   4541    4568   4641   4659   4711   4760   4823   4949   4968   5002   5021   5063   5081
                   5222    5240   5294   5313   5365   5384   5488   5507   5542   5561   5667   5796   5841
D.SSP  = 000020    1287#   3887   4081   4100   4188   4207   4254   4271   4333   4352   4386   4478   4505
                   4541    4568   4641   4659   4711   4760   4823   4949   4968   5002   5021   5063   5081
                   5222    5240   5294   5313   5365   5384   5488   5507   5542   5561   5667   5796   5841
D.SUNS= 040000     1324#
D.TIB  = 002000    1320#
D.UNLD= 040000     1297#
D.UNS  = 040000    1310#
D.VV   = 000100    1275#   3785   3798   3885   4079   4098   4186   4205   4252   4269   4331   4350   4384
                   4476    4503   4539   4566   4639   4657   4709   4758   4821   4947   4966   5000   5019
                   5061    5079   5220   5238   5292   5311   5363   5382   5486   5505   5540   5559   5665
                   5794    5839
D.WCUR= 000040     1315#
D.WGAT= 000100     1316#
D.WLE  = 004000    1307#
D.WRL  = 004000    1280#
D.XERR= 001000     1319#
ECCW   = 020000    1268#
ECH    = 000100    1229#
EMTVEC= 000030     1147#   3197*  3198*
EM1      046137    6052    8376#
EM10     046713    2194    8442#
EM100    051560    2531    8700#
EM101    051615    2571    8705#
EM11     046776    2200    8451#
EM12     047042    2205    2215   2225   2296   2551   2556   2561   2566   2581   2606   2686   2787   2822
                   2932    8457#
EM13     047100    2301    2386   2461   2611   8462#
EM14     047122    2326    8465#
EM15     047160    2341    8470#
EM16     047267    2942    2947   2957   2962   8483#
EM17     047332    2306    2371   2381   2396   2641   2907   3063   3118   3138   3148   8489#
EM18     047353    2255    2270   2286   2406   2646   3068   3088   3098   3108   3128   8492#
EM19     047374    2311    2316   2376   2391   2401   2651   2912   3123   3143   3153   8495#
EM2      046213    2159    8384#
EM20     047415    2265    2281   2291   2321   2411   2656   3093   3103   3113   3133   8498#
```

```
EM21     047436     2210   2220   2250   2260   2802   2827   2862   2937   8501#
EM22     047460     2466   8504#
EM23     047565     2471   8517#
EM24     047665     2276   8529#
EM25     047720     2351   8534#
EM26     047772     2416   8541#
EM27     050051     2426   8549#
EM3      046303     6993   8394#
EM36     050170     2536   2857   2998   8563#
EM39     050233     2356   2501   2636   2807   8569#
EM4      046356     8402#
EM40     050302     2361   2506   2812   8576#
EM41     050340     2331   8581#
EM5      046432     2165   2596   8410#
EM55     050406     2701   8588#
EM6      046453     2171   8413#
EM63     050441     2797   8593#
EM65     050462     2832   8596#
EM69     050504     2967   8599#
EM7      046474     2177   8416#
EM73     050535     2586   8604#
EM74     050556     3003   8607#
EM79     050603     2591   8611#
EM8      046563     8426#
EM80     050624     2230   8614#
EM81     050663     2235   8620#
EM82     050727     2240   8627#
EM83     051001     2245   8635#
EM84     051036     2587   2592   2597   8640#
EM85     051127     2336   2346   8650#
EM86     051172     2541   2546   2576   8656#
EM88     051230     2421   8661#
EM9      046637     218P   8434#
EM93     051271     2366   8667#
EM94     051352     2486   8676#
EM95     051406     2491   2496   8681#
EM96     051442     2511   8686#
EM97     051473     2516   8691#
ERRVEC=  000004     1140#  3211   3212*  3223*  3230*  3231*  3244*  3245*  3313*  3317*  3324*  3362*  3380*
                    3381*  7152   7153*  7155*  7158*
ESEC     001400     1981#
E.A0     007444     2101#  3885*  4079*  4098*  4186*  4205*  4252*  4269*  4331*  4350*  4384*  4476*  4503*
                    4539*  4566*  4639*  4657*  4709*  4758*  4821*  4947*  4966*  5000*  5019*  5061*  5079*
                    5220*  5238*  5292*  5311*  5363*  5382*  5486*  5505*  5540*  5559*  5665*  5794*  5839*
                    6269*  6273*  6277   6279*  6308   8963   8973   8984
E.A1     007450     2103#  3887*  4081*  4100*  4188*  4207*  4254*  4271*  4333*  4352*  4386*  4478*  4505*
                    4541*  4568*  4641*  4659*  4711*  4760*  4823*  4949*  4968*  5002*  5021*  5063*  5081*
                    5222*  5240*  5294*  5313*  5365*  5384*  5488*  5507*  5542*  5561*  5667*  5796*  5841*
                    6270*  6281   6283*  6328   8963   8973   8984
E.A2     007454     2105#  3889*  4083*  4102*  4190*  4209*  4256*  4273*  4335*  4354*  4388*  4480*  4507*
                    4543*  4570*  4661*  4762*  4825*  4951*  4970*  5004*  5023*  5083*  5242*  5296*  5315*
                    5367*  5386*  5490*  5509*  5544*  5563*  5669*  5798*  5843*  6271*  6285   6287*  8973
                    8984
E.A3     007460     2107#  6272*
E.B0     007446     2102#  3886*  4080*  4099*  4187*  4206*  4253*  4270*  4332*  4351*  4385*  4477*  4504*
                    4540*  4567*  4640*  4658*  4710*  4759*  4822*  4948*  4967*  5001*  5020*  5062*  5080*
```

```
                           5221*  5239*  5293*  5312*  5364*  5383*  5487*  5506*  5541*  5560*  5666*  5795*  5840*
                           6289   6291*  6318   8963   8973   8984
E.B1      007452           2104#  3888*  4082*  4101*  4189*  4208*  4255*  4272*  4334*  4353*  4387*  4479*  4506*
                           4542*  4569*  4642*  4660*  4712*  4761*  4824*  4950*  4969*  5003*  5022*  5064*  5082*
                           5223*  5241*  5295*  5314*  5366*  5385*  5489*  5508*  5543*  5562*  5668*  5797*  5842*
                           6293   6295*  6338   8963   8973   8984
E.B2      007456           2106#  3890*  4084*  4103*  4191*  4210*  4257*  4274*  4336*  4355*  4389*  4481*  4508*
                           4544*  4571*  4662*  4763*  4826*  4952*  4971*  5005*  5024*  5084*  5243*  5297*  5316*
                           5368*  5387*  5491*  5510*  5545*  5564*  5670*  5799*  5844*  6297   6299*  8973   8984
E.B3      007462           2108#  3891*  4085*  4104*  4192*  4211*  4258*  4275*  4337*  4356*  4390*  4482*  4509*
                           4545*  4572*  4663*  4764*  4827*  4953*  4972*  5006*  5025*  5085*  5244*  5298*  5317*
                           5369*  5388*  5492*  5511*  5546*  5565*  5671*  5800*  5845*  6301   6303*  8984
E.DDT     015444           3711*  3718*  3728#  6273
FATT1     032634           3847   5175   5416   5591   6199#  6929   6951
FATT2     032730           3983   4653   4742   4862   5075   5234   5752   5910   6226#
FHDHM     035070           6602#  6610
FHDTAB    035212           4137   4424   4594   5882   6635#
FLGTST    035472           6677   6681   6688   6692   6704#
FLOAD     035144           6619#
FMTE    = 000020           1227#
FMT1      001470           2016#  6645*  6646*  6647*  6652
FORM      031174           5773   5865   5868#  6965
FORMAT    001466           2015#  4136*  4423*  4593*  5881*  6645   6674
FRCYL     001344           1964#  5763*  8923   8947
FRDY      032320           3451   3465   3474   3576   3585   5656   6105   6116   6128#  6131   6462
FRDY1     032366           6142#  6145   6448
FSEC23    035000           5637   6578#
FTITLE    001340           1960#  5999   6001*
GBA       032126           3280   6066#
GDRVS     031766           3278   6021#
GINT      032154           3282   6079#
GNS     = ****** U         1345   8103   8104   8105   8106   8107   8109   8111   8112   8113   8114   8115   8116
                           8117
GO      = 000001           1191#
GSTAT     033664           3749   3842   3880   4037   4061   4143   4168   4233   4319   4377   4600   4704   4815
                           4929   4988   5112   5170   5274   5327   5351   5411   5475   5528   5586   5658   5732
                           5783   5828   5888   6209   6213   6233   6237   6397#  6466   6477   6489   6524   6604
                           6621   6946
GSTAT1    033720           6267   6408#
GSTAT2    034150           6413   6421   6429   6436   6444#
GTSWR   = 104406           6010   8109#
HASOF     007372           2072#  6167*  6183   8919   8928   8936   8944   8952   8960   8970   8981   8993
HBA       007362           2068#  6163*  8919   8928   8936   8944   8952   8960   8970   8981   8993
HCS1      007354           2065#  3468*  3469*  3470   3476   3579*  3580*  3581   3587   3794*  3837*  3852*  3877*
                           3881   3978*  3986   4034*  4038   4058*  4062   4140*  4144   4165*  4169   4230*  4234
                           4316*  4320   4374*  4427*  4430*  4433   4453*  4456*  4459   4494*  4522*  4525*  4528
                           4557*  4597*  4601   4635*  4695*  4737*  4745   4799*  4802   4857*  4865   4926*  4930
                           4985*  4989   5057*  5097*  5109*  5165*  5180*  5216*  5256*  5271*  5275   5348*  5352
                           5406*  5421*  5472*  5476   5525*  5529   5581*  5596*  5645*  5646*  5647   5661   5729*
                           5733   5747*  5755   5779*  5788   5824*  5833   5885*  5889   5905*  5913   6102*  6103
                           6113*  6114   6160*  6372   6376   6399*  6444*  6445*  6446   6467   6506*  6507*  6508
                           6778   6924*  6941*  6956*  8916   8925   8933   8941   8949   8957   8967   8978   8990
HCS2      007356           2066#  3515   3517   3521   3533   3616   3618   3622   3625   4322   4378   4530   4991
                           5113   5354   5531   6161*  6381   6385   8916   8925   8933   8941   8949   8957   8967
                           8978   8990
HDA       007364           2069#  6164*  6821   8919   8928   8936   8944   8952   8960   8970   8981   8993
```

```
HDB        007376          2074#
HDC        007374          2073#   6168*   6813    8919    8928    8936    8944    8952    8960    8970    8981    8993
HDS        007366          2070#   3519    3620    6165*   8916    8925    8933    8941    8949    8957    8967    8978    8990
HDTAB      001522          2033#   4022    4028    4031    4130    4415    4586    5715    5722    5725    5875    6637
HDWD       001510          2026#   8939
HD1        001464          2014#   6639*   6640*   6641*   6642*   6643*   6644*   6651
HEAD       001460          2012#   4135*   4422*   4592*   5860*   6639    6716
HEADA      001462          2013#   6568*   6569*   6570*   6571*
HER        007370          2071#   3510    3611    4065    4172    4236    4243    4462    4933    5278    5479    6166*   8916
                           8925    8933    8941    8949    8957    8967    8978    8990
HMR1       007400          2075#   6169*   8916    8925    8933    8941    8949    8957    8967    8978    8990
HMR2       007402          2076#   3478    3589    3750    3785    3798    3843    4497    4560    4705    4752    4842    5171
                           5412    5587    6146*   6151*   6170*   6415    6423    6431    6438    6490    6605    6622    6947
                           8916    8925    8933    8941    8949    8957    8967    8978    8990
HMR3       007404          2077#   5784    5829    6147*   6152*   6171*   6416    6424    6432    6439    6478    6525    8916
                           8925    8933    8941    8949    8957    8967    8978    8990
HOLD       032454          6132    6136    6160#   6409
HPAT       007410          2079#   6173*   8919    8928    8936    8944    8952    8960    8970    8981    8993
HPEND      007342          2047#   3307*   5920    5922*   6937*
HPOS       007406          2078#   6172*   8919    8928    8936    8944    8952    8960    8970    8981    8993
HT       = 000011          1050#   7287    7328
HVRC     = 000400          1231#
HWC        007360          2067#   6162*   8919    8928    8936    8944    8952    8960    8970    8981    8993
HZ         001364          1973#   6843
H.A0       007424          2090#   6308    6438*   8963    8976    8987
H.A1       007430          2092#   6328    6431*   8963    8976    8987
H.A2       007434          2094#   6423*   6535    8976    8987
H.A3       007440          2096#   6415*
H.B0       007426          2091#   6318    6439*   8963    8976    8987
H.B1       007432          2093#   6338    6432*   8963    8976    8987
H.B2       007436          2095#   6424*   6548    8976    8987
H.B3       007442          2097#   6416*   6558    6568    8987
IDAE     = 002000          1233#
IE       = 000100          1192#
ILF      = 000001          1223#
INTER      036720          6094    7033#
IOTVEC=    000020          1145#   3195*   3196*
IR       = 000100          1210#
LC         015434          3710*   3717*   3724#   3875
LCLKF      007516          2138#   3315*
LCVEC      001330          1952#   3316
LF       = 000012          1051#   7322    7328    8123    8131    8133    8137    8147    8156    8163    8171    8176    8181
                           8186    8192    8198    8204    8208    8214    8219    8226    8235    8240    8245    8251    8253
                           8257    8263    8271    8281    8286    8293    8297    8299    8303    8307    8315    8319    8327
                           8335    8341    8349    8355    8362    8367    8370    8376    8394    8402    8477    8504    8511
                           8517    8522    8534    8554
LIMERR     001514          2029#
LKS        001326          1950#   3314    6832*   6853*
MASK       015440          3713*   3720*   3726#   6496    6541
MASK1      015442          3714*   3721*   3727#   6491    6517    6526    6536    6549
MCLK     = 000400          1263#
MC1        015436          3712*   3719*   3725#   5743    5864    5899
MDS      = 001000          1213#   3515    3616    6385
MEMBAS=    172100          1956#   3233    3242
MEMERR     036656          3236    7019#
MEMPAR     013056          3230#
```

```
MEMVEC= 000114          1955#   3236*   3237*
MERD  = 001000          1264#
MESMSK= 000017          1258#
MEWD  = 002000          1265#
MIND  = 000200          1262#
MSG1    043420          6002    8123#
MSG10   044713          3415    3455    8253#
MSG11   044740          7066    8257#
MSG12   045000          5193    8263#
MSG13   045145          3338    8281#
MSG14   045243          6255    8293#
MSG15   045266          3535    3698    8297#
MSG16   045300          3747    8299#
MSG17   045324          3967    8303#
MSG2    044307          3261    8204#
MSG26   045354          4014    4041    4093    4147    4200    4240    4345    4436    4490    4553    4604    4805    4817
                        4961    5014    5306    5377    5500    5554    5736    5892    8307#
MSG27   045432          3676    8315#
MSG3    044370          3277    8214#
MSG37   045462          5145    8319#
MSG38   045540          5158    8327#
MSG39   045620          5121    8335#
MSG4    044422          3279    8219#
MSG40   045662          4013    8341#
MSG5    044467          3281    8226#
MSG6    044550          7038    8235#
MSG7    044604          3296    8240#
MSG74   045775          6916    8355#
MSG75   046043          6918    8362#
MSG76   046100          6969    8367#
MSG77   046122          6974    8370#
MSG8    044641          5034    8245#
MSG9    044700          5038    8251#
MSP   = 000100          1261#
MULT6   035656          6736    6762#
M.ALGN= 040000          1334#
M.DPID  037042          7081#
M.DP40  037100          7092#   7114
M.DP41  037134          7097    7105#
M.DP42  037144          7103    7109#
M.DP44  037176          7116    7122#
M.DP50  037210          7090    7127#
M.DRV = 000007          1328#
M.HEAD= 007000          1336#   6569
M.ID  = 000003          1333#
M.PAR = 100000          1337#   6882
M.SECT= 000760          1335#   6479    6559
M.SER = 077770          1329#
NED   = 010000          1216#   3521    3533    3622    3625    6381
NEM   = 004000          1215#
NUDRV   015146          3547    3655#   5945
NXF   = 000004          1225#
OFFERR  001456          2009#   5046*   5117*   5119    5135    5143    5152    5156
OFFSET= 000015          1183#   4635    5057    5216
OFST  = 000004          1244#
OPI   = 020000          1236#
```

```
OR     = 000200         1211#
O.ADR1   055472         9274*    9325*    9327*    9497     9498*    9536*    9551     9568     9759#
O.BACK   056216         9369     9410#
O.BD     057510         9555     9556     9704#
O.BKPT   055724         9320#    9373
O.BK1    056656         9268     9533#
O.BK2    056712         9538     9541#
O.BK3    056730         9540     9547#
O.BRK    056646         9285     9531#
O.BUF    057537         9613     9622     9627     9732#
O.B2     057012         9552     9561#
O.B3     057052         9560     9569#
O.CADV   057222         9386     9403     9426     9430     9460     9464     9569     9612#
O.CLGL   056040         9343     9345     9353#
O.CLGT=  000014         9357     9722#
O.CLS1   057446         9680     9682#
O.CR     057512         9689     9691     9692     9706#
O.CRET   056152         9364     9393#
O.CRLF   057456         9401     9458     9554     9593     9598     9689#
O.CRLS   057464         9338     9691#
O.CRS    057470         9690     9692#
O.CSR1   057506         9589*    9605     9700#
O.CSR2   057507         9590*    9606     9701#
O.CT     055474         9518*    9561*    9563*    9760#
O.C1     056624         9521#    9562
O.DCD    055770         9326     9328     9338#    9395     9446     9571
O.DCD1   055774         9339#    9387     9431
O.EFF    056300         9372     9437#
O.EFF1   056416         9449     9468#
O.ENTR   055504         9265#
O.ERR    055754         9312     9323     9335#    9358     9383     9441     9684
O.ERR1   056310         9441#    9514
O.ERR2   056130         9383#    9417     9490
O.FTYP   057364         9337     9405     9420     9462     9567     9638     9654     9659#
O.GET    057306         9291     9341     9635#    9640     9642     9644
O.GO     056462         9366     9488#
O.GO2    056540         9499#    9524
O.LG   = 000006         9295     9730#
O.LGCH   057515         9354     9710#    9722
O.LGDR   056070         9361     9363#    9375
O.LGL  = 000030         9375#
O.LGL1   056042         9354#    9359
O.LGL2   056062         9355     9360#
O.MSK    055464         9442     9443     9445     9752#
O.ODT    055500         1354     9263#    9741
O.OFST   056226         9370     9416#
O.OF1    056274         9429     9431#
O.OP1    056164         9367     9399#
O.OP2    056172         9316     9401#    9412
O.ORPC   055676         9308#    9368
O.P      057505         9283*    9488*    9513     9515*    9564*    9698#
O.PRI    055462         9275     9537     9541     9751#
O.PROC   056570         9374     9513#
O.PR1    056612         9517     9519#
O.RALL   055744         9286     9321     9327#
O.RCSR=  177560         9253#    9589     9591*    9601     9603     9605*    9635
```

```
O.RDB = 177562          9252#   9637
O.REGT  055626          9291#   9365
O.RSE1  057204          9602    9605#
O.RSP   055636          9293#   9296
O.RST   055542          9264    9273#
O.RSTT  057154          9494    9520    9598#
O.RST1  055572          9272    928C#
O.RTIT  056566          9282*   9508#
O.SCAN  056000          9302    9341#   9352
O.SP    055670          9294    9303#
O.SPC   057234          9615#   9623
O.SP1   055656          9299#   9304
O.STM = 000340          9242#   9284    9493    9519    9521
O.STRT  055530          9263    9270#
O.SVR   057064          9273    9533    9575#
O.SVTT  057122          9553    9565    9589#
O.T     057504          9495*   9522*   9534    9697#
O.TBIT  056512          9495#   9535
O.TBT = 000020          9243#   9496    9523    9559
O.TC    057450          9677    9683#
O.TCLS  057432          9308    9393    9399    9410    9676#
O.TCSR= 177564          9255#   9590    9592*   9599    9606*   9659    9665
O.TDB = 177566          9254#   9661*   9667*
O.TL    057531          9292    9295    9303    9724#   9730
O.TRTC  057546          9327    9498    9737#
O.TVEC= 000014          9241#   9266    9284*   9285*
O.TYPE  057350          9557    9628    9651#   9655    9693
O.TYP1  057430          9652    9663    9671#
O.TYP2  057410          9665#   9666    9669
O.UIN   055476          9274    9497*   9536    9761#
O.UPC   055456          9267*   9492*   9507    9531*   9548    9550*   9749#
O.URO   055440          9270    9300    9742#
O.USP   055454          9271*   9576*   9577    9748#
O.UST   055460          9265*   9496*   9506    9523*   9532*   9539    9559*   9750#
O.WDS   056306          9438    9440#
O.WDS2  056324          9445#   9467
O.WDS3  056354          9456#   9469    9475    9484
O.WDS4  056412          9456    9466#
O.WRD   056120          9363    9379#
O.WRDA  056144          9380    9388#
O.WRD1  056126          9382#   9389    9406
O.WSCH  056304          9371    9439#
O.XXX   057502          9575*   9583    9696#
O.45    055604          9281    9283#
PACK  = 000003          1178#   3794
PARAM   001336          1959#   3161*   3164*   3274
PARSRT  012546          1351    3161#
PAT   = 000020          1259#
PCA   = 004000          1266#
PCD   = 010000          1267#
PCLKF   007520          2139#   3319*   3326*   6830    6851
PCVEC   001332          1953#   3320    3327
PCYL    001352          1967#
PFSRT   015446          3730#   7068
PGE   = 002000          1214#
PIP   = 020000          1252#
```

```
PIRQ   = 177772     1057#
PIRQVE= 000240      1151#
PKRB     001324     1949#
PKS      001320     1947#   3318    3325    6835*   6855*
PKSB     001322     1948#   6834*
PPTP     007472     2123#   3265*
PRGSRT   012562     3162    3165#   6053
PR0    = 000000     1074#   3167    3357
PR1    = 000040     1075#
PR2    = 000100     1076#
PR3    = 000140     1077#
PR4    = 000200     1078#
PR5    = 000240     1079#   1946
PR6    = 000300     1080#   3330
PR7    = 000340     1081#   3231    3237    3381    7063    7064
PS     = 177776     1054#   1055
PSEC     001376     1980#
PSW    = 177776     1055#
PWRVEC= 000024      1146#   3201*   3202*   7053*   7062*   7063*
QKCYLD   034430     6504#
RCYLA    034656     6353    6548#
RCYLD    034604     6350    6535#
RDCHR  = 104410     7730    8112#
RDCYLA   034534     6523#
RDCYLD   034344     6488#
RDDATA= 000021      1185#   3877    4230
RDGATE= 100000      1270#
RDHEAD= 000025      1187#   4799
RDLIN  = 104411     6021    7804    8113#
RDOCT  = 104412     6066    6079    8114#
RDSEC    034274     6476#   6579    6583    6586    6734
RDTAB    006336     2041#   4226    4284    4292
RDY    = 000200     1193#   5654    6128    6142    6510
RECAL  = 000013     1182#   3837    5165    5406    5581    6941
RESREG= 104414      7864    7916    8116#   9210
RESVEC= 000010      1141#
RHEAD    034752     6358    6568#
RHTAB    001726     2034#   4798    4836    6741    6743    6747    6750    8955
RKASOF= 000016      1164#   3371    4632*   4677*   4721*   4773*   5051*   5213*   6167    6202    6228
RKBA   = 000004     1159#   3367    3871*   3924*   3942*   3951*   3960*   4031*   4054*   4130*   4160*   4226*   4299*
                    4312*   4371*   4415*   4448*   4518*   4586*   4920*   4980*   5105*   5267*   5343*   5468*   5521*
                    5641*   5725*   5775*   5820*   5875*   6163
RKCS1  = 000000     1157#   3364    3470*   3581*   3835*   3850*   4075*   4182*   4472*   4693*   4703*   4943*   5095*
                    5163*   5178*   5254*   5288*   5404*   5419*   5579*   5594*   5647*   5654    6103*   6114*   6128
                    6142    6160    6410*   6418*   6426*   6434*   6446*   6508*   6510    6939*   6954*
RKCS2  = 000010     1161#   3365    3449*   3463*   3467*   3574*   3578*   3836*   3851*   4162*   4311*   4449    4519*
                    4694*   4813    4921*   4981*   5096*   5106*   5164*   5179*   5255*   5268*   5344*   5405*   5420*
                    5469*   5522*   5580*   5595*   5642*   5776*   5821*   6161    6398*   6411*   6419*   6427*   6435*
                    6460*   6464*   6940*   6955*
RKDA   = 000006     1160#   3368    3873*   3926*   3944*   3953*   3962*   4053*   4159*   4225*   4301*   4314*   4373*
                    4447*   4520*   4919*   4983*   5053*   5108*   5266*   5346*   5644*   6164
RKDB   = 000024     1166#   3373    4324    4532    4808    4809    4810    4993    5116    5356    5533
RKDC   = 000020     1165#   3372    3875*   4793*   5467*   5520*   5727*   5745*   5774*   5877*   5902*   6168
RKDS   = 000012     1162#   3369    6165
RKECPS= 000030      1170#   3377    6172
RKECPT= 000032      1171#   3378    6173
```

F16 .


```
RKER   = 000014       1163#   3370    6166
RKMR1  = 000026       1167#   3374    3748*   3841*   5169*   5326*   5410*   5585*   6169    6208*   6212*   6232*   6236*
                      6412*   6420*   6428*   6465*   6476*   6488*   6505*   6523*   6603*   6620*   6945*
RKMR2  = 000034       1168#   3375    6146    6151    6170    6516
RKMR3  = 000036       1169#   3376    6147    6152    6171
RKPRI    001316       1946#   6095
RKVEC    001314       1945#   3301*   6083*   6086*   6053
RKWC   = 000002       1158#   3366    3876*   4032*   4055*   4131*   4163*   4227*   4313*   4372*   4416*   4450*   4517*
                      4587*   4922*   4982*   5107*   5269*   5345*   5470*   5523*   5643*   5726*   5777*   5822*   5876*
                      6162
RLS    = 000010       1207#
RSEC     034714       6357    6558#
RTT    = 000006       9245#
SAVREG= 104413        7849    7890    8115#   9154
SBPAR    036156       6278    6282    6286    6290    6294    6298    6302    6868#
SCLR   = 000040       1209#   3449    3463    3574    6460
SCOP1  = 104415       3460    3571    3788    4046    4152    4219    4305    4365    4440    4580    5336    5765    5813
                      8117#
SCOP1$   036230       6892#   8117
SEC      001370       1977#   6844*
SECFLG   035372       6653    6673#
SECNT    001374       1979#
SECTOR   001402       1982#   4052*   4053    4070*   4071    4158*   4159    4177*   4178    4225    4301    4314    4373
                      4446*   4447    4467*   4468    4520    4918*   4919    4938*   4939    4983    5108    5265*   5266
                      5283*   5284    5346    6478*   6479*   6480*   6481*   6482*   6483*   6558*   6559*   6560*   6561*
                      6562*   6563*   6580    6584    6587    6735*   6739    6740*   6741*   6752    6762*   6763    6764*
                      6765*
SEEK   = 000017       1184#   3978    4737    4857    5747    5905
SELDRV= 000001        1177#   3468    3579    4430    4456    4494    4525    4557    6399    6444    6506
SETINT   032206       3302    6084    6093#
SIZFLG   007524       2141#   3303*   3453    6039*   6059*
SKI    = 000002       1224#
SORT     035550       6732#
SPACE2   055414       9188    9197    9208    9220#
SRTSPL= 000011        1181#   6924
SRTTAB   002132       2035#   6744
ST     = 177776       9239#   9265    9266*   9279*   9493*   9519*   9521*   9547*
STACK  = 001100       1045#   3166    3193    3355    3406    3461    3567    3572    3670    3740    3780    3789    3829
                      4009    4047    4125    4153    4220    4306    4366    4410    4441    4581    4625    4913    5206
                      5337    5456    5627    5707    5766    5814    5941    6363    7027    7065    9216
STAPT    031634       5970    5976#
START    012556       1349    3164#
START1   013164       3255#
STKLMT= 177774        1056#
STOP     036262       5924    6907#   7571    7654
ST2      013234       3263    3274#
ST3      013300       3264    3292#
ST4      013332       3276    3297    3300#
ST5      013356       3283    3305#   3446    3509    5982    6976
SUBCLR   034214       3744    3782    3791    3831    3862    3900    3974    4017    4049    4127    4155    4222    4308
                      4368    4412    4443    4583    4629    4789    4854    4915    5048    5210    5339    5464    5634
                      5712    5770    5816    5872    6460#   6909    6921
SVAL   = 100000       1254#
SWR      001140       1863#   3191    3213*   3215    3221*   3228*   6008    6314    6324    6334    6344    6892    6991
                      6999    7004    7023    7147    7161    7163    7169    7176    7214    7221    7233    7237    7574
                      7613    7668*   9212
```

```
SWREG     000176          1347#   3221    6008    7574    7613    7636
SW0     = 000001          1109#
SW00    = 000001          1099#   1109
SW01    = 000002          1098#   1108
SW02    = 000004          1097#   1107
SW03    = 000010          1096#   1106
SW04    = 000020          1095#   1105
SW05    = 000040          1094#   1104
SW06    = 000100          1093#   1103
SW07    = 000200          1092#   1102
SW08    = 000400          1091#   1101
SW09    = 001000          1090#   1100
SW1     = 000002          1108#
SW10    = 002000          1089#
SW11    = 004000          1088#
SW12    = 010000          1087#   9212
SW13    = 020000          1086#   6991
SW14    = 040000          1085#   7004
SW15    = 100000          1084#
SW2     = 000004          1107#
SW3     = 000010          1106#
SW4     = 000020          1105#
SW5     = 000040          1104#
SW6     = 000100          1103#
SW7     = 000200          1102#
SW8     = 000400          1101#
SW9     = 001000          1100#   6314    6324    6334    6344    6892    6999    7023
TBITVE= 000014            1142#
TEMP1     007412          2082#   3450*   3464*   3471*   3473*   3478*   3479*   3480    3575*   3582*   3584*   3589*   3590*
                          3591    3982*   4652*   4741*   4861*   5074*   5233*   5649*   5751*   5909*   6104*   6115*   6130*
                          6144*   6200*   6204*   6230*   6246    6248*   6275    6277*   6279    6281*   6283    6285*   6287
                          6289*   6291    6293*   6295    6297*   6299    6301*   6303    6305*   6397    6402*   6408    6441*
                          6447*   6461*   6504    6509*   6512*   6518*   6578*   6590*   6602*   6607*   6619*   6624*   6655*
                          6658*   6659    6874*   6882*   8939
TEMP2     007414          2083#   3846*   3865*   3903*   3904    3940*   3949*   3958*   5174*   5415*   5590*   6206*   6609*
                          6656*   6657*   6659*   6660    6928*   6950*
TEMP3     007416          2084#   3866*   3906    3911    3916    3937*   3938    3947    3956
TEMP4     007420          2085#   3870*   3871    3924    3941*   3942    3950*   3951    3959*   3960
TEMP5     007422          2086#   3872*   3873    3925*   3926    3943*   3944    3952*   3953    3961*   3962    5633*   5644
                          5680    5682*   5710*   5741*   5870*   5897*
TERR1     036000          6782    6786    6793    6797    6811#   6816    6823
TIMUP     001372          1978#   6829*   6840*   6846*
TITLE     031706          3257    3295    5999#
TKVEC   = 000060          1149#   7551*   7552*
TOCYL     001346          1965#   4132*   4134    4417*   4421    4588*   4591    4792*   4836    5709*   5717    5720    5727
                          5742*   5743    5762*   5768    5774    5808    5863*   5864    5869*   5877    5879    5898*   5899
                          8923    8947    8955
TPVEC   = 000064          1150#
TRAPPC    001334          1957#   7020*   8922
TRAPVE= 000034            1148#   3199*   3200*   7064*
TRT     = 000003          9244#   9737
TRTVEC= 000014            1143#
TRUERR    035700          4067    4174    4464    4935    5280    5481    6776#
TSTATN    032602          3855    4698    5100    5183    5259    5424    5599    6181#   6959
TST1      013546          3331    3353#
TST10     016704          3930    3966    4007#
```

```
TST11     017406     4123#
TST12     021156     4408#
TST13     022400     4623#
TST14     023676     4887   4911#
TST15     025442     5153   5204#
TST16     026756     5454#
TST17     027736     5625#
TST2      013730     3382   3404#
TST20     030236     5631   5639   5681   5705#
TST3      014554     3565#
TST4      015156     3601   3668#
TST5      015446     3715   3722   3738#
TST6      015574     3743   3778#
TST7      015670     3786   3799   3827#
TYPDS  =  104405     5959   8107#
TYPE   =  104401     3261   3277   3279   3281   3296   3338   3415   3425   3455   3489   3535   3676   3698
                     3747   3967   3971   3972   4013   4014   4041   4093   4147   4200   4240   4345   4436
                     4490   4553   4604   4805   4817   4961   5014   5034   5038   5044   5121   5127   5128
                     5145   5158   5193   5306   5377   5500   5554   5736   5892   5957   5960   6002   6052
                     6255   6916   6918   6969   6974   6993   7038   7066   7216   7224   7292   7386   7512
                     7568   7580   7634   7635   7638   7649   7659   7670   7689   7737   7743   7748   7752
                     7757   7758   7760   7763   7767   7833   7835   7975   8103#  9164   9165   9169   9170
                     9179   9188   9192   9197   9199   9205   9208
TYPERR    055140     7223   9154#
TYPOC  =  104402     7637   8104#  9185
TYPON  =  104404     8106#
TYPOS  =  104403     3428   3492   3538   3702   5041   5124   6258   6996   7041   8105#
T.A2   =  000001     2112#  4088   4107   4195   4214   4261   4278   4340   4359   4485   4548   4666   4715
                     4767   4830   4956   4975   5009   5028   5088   5247   5301   5320   5372   5391   5495
                     5514   5549   5568   5803   5848   6348
T.B2   =  000002     2113#  4088   4107   4195   4214   4261   4278   4340   4359   4485   4548   4666   4715
                     4767   4830   4956   4975   5009   5028   5088   5247   5301   5320   5372   5391   5495
                     5514   5549   5568   5803   5848   6351
T.B3   =  000004     2114#  6355
T1        001404     1984#
T10       001406     1985#  3450   3464   3473   3575   3584   3846   5174   5415   5590   6104   6447   6461
                     6509   6950
T100      001414     1988#  6928
T50       001410     1986#
T500      001412     1987#
T5000     001416     1989#  4652   5074   5233   6578
T50000    001420     1990#  3471   3582   3982   4741   4861   5751   5909   6115
UFE    =  000400     1212#  3517   3618
UNLD      007336     2043#  3305*  6912   6932*
UNLOAD=   000007     1180#
UNS    =  040000     1237#
UPE    =  020000     1217#
VERIFY    014554     3442   3447   3549#
VV     =  000100     1248#
WCE    =  040000     1218#  4322   4378   4530   4991   5113   5354   5531
WDCNT     001472     2017#  8939
WD1       001452     2006#  4285*  4324*  4532*  4993*  5116*  5356*  5533*  8931
WD2       001454     2007#  4286*  4325*  4533*  4994*  5357*  5534*  8931
WLE    =  004000     1234#
WORD      001506     2025#
WRDATA=   000023     1186#  4058   4165   4453   4926   5271   5472   5645   5779   5824
```

```
WRHEAD= 000027              1188#   4034    4140    4427    4597    5729    5885
WRL   = 004000              1251#
WRTCHK= 000031              1189#   4316    4374    4522    4985    5109    5348    5525
WRTGAT= 040000              1269#
$APTHD  001000              1376    1382#
$ASTAT= ****** U            7429    7444
$ATYC   040432              7400    7402#
$ATY1   040406              7398#
$ATY3   040414              7277    7399#
$ATY4   040424              7229    7401#
$AUTOB  001134              1860#   6012*   7627    7784
$BASE   001264              1930#   3300*   3363    6070*   6072*
$BDADR  001122              1855#
$BDDAT  001126              1857#
$BELL   001200              1881#   7216    7249    7580    7777
$CDW1   001270              1932#
$CDW2   001272              1933#
$CHARC  040156              7294*   7304*   7311    7320*   7325#
$CKSWR  041400              7613#   8111
$CMTAG  001100              1843#   3188    3189    3197    3203    3204    3205
$CM3  = 000000              1873#
$CM4  = 000006              1873#   1874#   1875#   1876#   1877#   1878#   1879#
$CNTLC  042310              7568    7649    7777#
$CNTLG  042322              7634    7779#
$CNTLU  042315              7659    7752    7778#
$CPUOP  001236              1904#
$CRLF   001205              1883#   3425    3489    3971    3972    5044    5127    5128    7224    7249    7293    7328    7670
                            7757    7777    7838    9164    9169    9179    9192    9205
$DBLK   040376              7352    7386    7394#
$DB2D   042632              7890#   7952
$DB20   042512              3969    7849#
$DDW0   001274              1934#
$DDW1   001276              1935#
$DDW2   001300              1936#
$DDW3   001302              1937#
$DDW4   001304              1938#
$DDW5   001306              1939#
$DDW6   001310              1940#
$DDW7   001312              1941#
$DECVL  043012              7892    7938#
$DEVCT  001220              1895#   3248*   3311*   3680    5942*   5943
$DEVM   001266              1931#   3419
$DOAGN  031610              5953    5962    5968#
$DTBL   040366              7355    7390#
$ENDAD  031600              1362    5964#   7244
$ENDCT  031546              3203    5955#
$ENDMG  031617              5957    5972#
$ENULL  031614              5960    5971#
$ENV    001230              1900#   5976    6006    7226    7272    7407    7431
$ENVM   001231              1901#   3226    3411    7274    7279    7409
$EOP    031452              4015    4042    4094    4148    4201    4241    4346    4437    4491    4554    4605    4806    4818
                            4962    5015    5307    5378    5501    5555    5737    5893    5936#   9217
$EOPCT  031540              3203*   5952#   5956    6970*
$EOP1   031512              3386    3677    5944    5946#   6972    7015
$ERFLG  001103              1846#   6894    7136    7165    7167    7173*   7195    7211*   7249    9214
$ERMAX  001115              1852#   3206*   7167    7190*   7195
```

```
$ERROR   037500             3197     7209#
$ERRPC   001116             1853#    7218*    7219*    7220     7249     8915     8923     8931     8939     8947     8955     8963     8973
                            8984
$ERRTB   007526             2156#    9161
$ERTTL   001112             1850#    7217*    7249
$ESCAP   001176             1880#    3205*    3308*    4634*    4651*    5056*    5073*    5215*    5232*    5773*    5868*    5903*    5939*
                            6306     6310*    6320*    6330*    6340*    6360*    6364*    6971*    7019*    7022*    7189*    7240     7242
                            7249
$ETABL   001230             1899#
$ETEND   001314             1388     1942#
$FATAL   001212             1892#    7435*
$FFLG    040652             7398*    7401*    7429     7438*    7446#
$FILLC   001156             1871#    7297     7328
$FILLS   001155             1870#    7328
$GDADR   001120             1854#
$GDDAT   001124             1856#
$GET42   031570             5961#
$GTSWR   041470             7635#    8109
$HD    = 000000             1016
$HIBTS   001000             1383#
$HIOCT   042510             7826*    7837#
$ICNT    001104             1847#    7180*    7181     7183*    7194
$INTAG   001135             1861#    7632*    7651     7671     7784
$ITEMB   001114             1851#    7220*    7228     7249     9155
$LF      001206             1884#    7249     7328     7767     7777     7838
$LFLG    040651             7439*    7445#
$LPADR   001106             1848#    3207*    7002     7171*    7187*    7192     7194
$LPERR   001110             1849#    3208*    6365     6896     6899*    7028     7171     7188*    7194     7239
$MADR1   001242             1917#
$MADR2   001246             1921#
$MADR3   001252             1924#
$MADR4   001256             1927#
$MAIL    001210             1384     1388     1890#    3225     6006     7186     7226     7272
$MAMS1   001240             1911#
$MAMS2   001244             1919#
$MAMS3   001250             1922#
$MAMS4   001254             1925#
$MBADR   001002             1384#
$MFLG    040650             7399*    7405     7440*    7444#
$MNEW    042340             7638     7782#
$MSGAD   001224             1897#    7415*    7418
$MSGLG   001226             1898#    7420*
$MSGTY   001210             1891#    7413     7421*    7433     7437*
$MSWR    042327             7635     7780#
$MTYP1   001241             1912#
$MTYP2   001245             1920#
$MTYP3   001251             1923#
$MTYP4   001255             1926#
$MULT    043122             7994#
$MXCNT   037476             7184     7194#
$NULL    001154             1869#    7299     7328
$NWTST= 000001              3343#    3345     3389#    3391     3550#    3552     3659#    3661     3731#    3733     3772#    3774     3804#
                            3806     3997#    3999     4113#    4115     4400#    4402     4610#    4612     4890#    4892     5194#    5196
                            5432#    5434     5607#    5609     5686#    5688
$OCNT    041076             7484*    7513*    7526#
$OCTVL   042614             3751     7851     7876#
```

```
$OMODE  041100          7479*   7483*   7488    7491*   7502*   7528#
$OVER   037462          7148    7164    7172    7182    7191#
$PASS   001216          1894#   3225*   3742    3965    5949*   5950*   5958    5971    5978    7178    7195
$PASTM  001006          1386#
$PWRCT  037040          7059*   7060*   7070#
$PWRDN  036754          3201    7053#   7062
$PWRUP  036766          7053    7059#
$QUES   001204          1882#   7249    7328    7689    7760    7777    7835    7838
$RDCHR  041742          7702#   8112
$RDDEC= ****** U        8115
$RDLIN  042032          7725#   8113
$RDOCT  042352          7799#   8114
$RDSZ = 000022          7718#
$RESRE  043272          8060#   8116
$RTNAD  031612          5970#
$R2A  = ****** U        8117
$SAVRE  043234          8044#   8115
$SB2D   043026          7950#
$SCOPE  037220          3195    7145#
$SETUP= 000137          3158#   3194    3195    3197    3199    3201    3203    3204    3205    3207    5947    6003    7146
                        7210    7236    7244    7572    7577    7578    7608    7784
$STUP = 177777          3158#
$SUPRS  043062          3768    3970    7967#
$SVLAD  037426          7156    7185#
$SVPC = 000244          1360#   1365
$SWR  = 167400          1001#   1016    1021    1022    1023    1024    1025    1026    1027    1028    1879    1880    1881
                        3204    3205    3207    3208    3354    3405    3566    3669    3739    3779    3828    4008    4124
                        4409    4624    4912    5205    5455    5626    5706    5933    5948    5963    5969    5971    7137
                        7138    7139    7140    7141    7147    7159    7161    7162    7165    7166    7167    7174    7175
                        7176    7188    7191    7194    7201    7202    7203    7204    7205    7214    7221    7233    7237
                        7249
$SWREG  001232          1902#   3228
$SWRMK= 000000          1028    1029    7141    7142    7163
$TESTN  001214          1893#   3671*   7186*   8915    8922    8923    8931    8939    8947    8955    8963    8973    8984
$TIMES  001174          1879#   3204*   3354*   3405*   3566*   3669*   3739*   3779*   3828*   4008*   4124*   4409*   4624*
                        4912*   5205*   5455*   5626*   5706*   5940*   5948*   7174*   7181    7184*   7194
$TKB    001146          1866#   7532    7553    7564    7589    7617    7644
$TKCNT  041102          7533#   7548*   7578    7595*   7709    7711*
$TKINT  041112          3171    7548#   7569    7630
$TKQEN= 041111          7537#   7603    7714
$TKQIN  041104          7534#   7549*   7550    7601*   7602*   7603    7605*
$TKQOU  041106          7535#   7550*   7712    7713*   7714    7716*
$TKQSR  041110          7536#   7549    7605    7716
$TKS    001144          1865#   7532    7554*   7585*   7587    7593*   7615    7631*   7641    7653*   7673*
$TKSRV  041162          7551    7564#
$TMP0   001160          1873#
$TMP1   001162          1874#
$TMP2   001164          1875#
$TMP3   001166          1876#
$TMP4   001170          1877#   3309*   3469    3488    3502*   3512*   3580    3595    3598*   3613*   3657*   3696*   3708
                        5458    5646    6102    6113    6445    6507
$TMP5   001172          1878#   6306*   6360    6364
$TN   = 000021          1002#   1016    3331    3343    3354#   3382    3389    3405#   3550    3566#   3601    3659    3669#
                        3671    3672    3715    3722    3731    3739#   3743    3772    3779#   3786    3799    3804    3828#
                        3930    3966    3997    4008#   4113    4124#   4400    4409#   4610    4624#   4887    4890    4912#
                        5153    5194    5205#   5432    5455#   5607    5626#   5631    5639    5681    5686    5706#
```

```
$TNPWR  042742          7897    7898    7918#
$TPB    001152          1868#   7317*   7328
$TPFLG  001157          1872#   7266    7328
$TPS    001150          1867#   7315    7328
$TRAP   043330          3199    8080#
$TRAP2  043352          8091#   8102
$TRP  = 000016          8095#   8104#   8105#   8106#   8107#   8108#   8109    8110#   8111    8112#   8113#   8114#   8115#
                        8116#   8117#   8118#
$TRPAD  043364          8085    8102#
$TSTM   001004          1385#
$TSTNM  001102          1845#   3672*   5947*   5980*   7136    7163    7185*   7186    7191    7195    7213    7249
$TTYIN  042266          7727    7728    7740    7758    7772    7776#
$TYPBN= ****** U        8108
$TYPDS  040162          7340#   8107
$TYPE   037700          7266#   7426    8095    8103
$TYPEC  040112          7296    7303    7310    7315#   7316    7675
$TYPEX  040160          7321    7323    7326#
$TYPOC  040700          7482#   8104
$TYPON  040714          7481    7484#   8106
$TYPOS  040654          7477#   8105
$UNIT   001222          1896#   3249*   3312*   3682*   3689    3699    3836    3851    4694    5096    5164    5179    5255
                        5405    5420    5580    5595    6182    6201    6227    6269    6270    6271    6272    6398    6411
                        6419    6427    6435    6464    6940    6955
$UNITM  001010          1387#
$USWR   001234          1903#
$VECT1  001260          1928#
$VECT2  001262          1929#
$XTSTR  037232          7150#
$$GET4= 000000          5963#
$OFILL  041077          7478*   7482*   7492    7527#
$4OCAT= ****** U        7147    7223
.     = 055500          1341#   1345#   1350#   1353#   1360    1361#   1363#   1365#   1366#   1372    1373#   1375#   1377#
                        1842#   1885    2033#   2034#   2035#   2037#   2038#   2039#   2040#   2041#   3192    3207    3208
                        5971    5975    7055    7111    7194    7195    7249    7328    7394#   7447#   7532    7536#   7537
                        7538#   7776#   7777    7784#   7838    7876#   7938#   8914#   9229#   9230#   9375    9600    9604
                        9636    9660    9722    9730    9733#   9741#
.$ASTA= ****** U        7399    7402
.$X   = 001000          1372#   1377
```

```
CALIB    1479#  3834   5161   5402   5577   6938
CHECK    1433#  3892   4086   4105   4193   4212   4259   4276   4338   4357   4391   4483   4510   4546   4573
         4643   4664   4713   4765   4828   4954   4973   5007   5026   5065   5086   5224   5245   5299   5318
         5370   5389   5493   5512   5547   5566   5672   5801   5846
COMMEN   1152#
CWD2     1448#  5325
DRCLR    1461#  3849   4692   5094   5177   5253   5418   5593   6953
ENDCOM   1152#
EOPGM    1800#  5937
ERROR    1046#  3385   3444   3452   3466   3475   3498   3507   3526   3530   3541   3545   3577   3586   3604
         3631   3635   3639   3643   3745   3783   3792   3796   3800   3832   3839   3845   3848   3854   3857
         3863   3879   3883   3895   3896   3897   3898   3901   3908   3913   3918   3921   3933   3975   3980
         3984   3988   4018   4036   4040   4050   4060   4073   4077   4089   4090   4091   4092   4095   4108
         4109   4110   4111   4128   4142   4146   4156   4167   4180   4184   4196   4197   4198   4199   4202
         4215   4216   4217   4218   4223   4232   4238   4245   4248   4262   4263   4264   4265   4279   4280
         4281   4282   4290   4309   4318   4326   4329   4341   4342   4343   4344   4360   4361   4362   4363
         4369   4376   4380   4394   4395   4396   4397   4413   4429   4432   4435   4444   4455   4458   4470
         4474   4486   4487   4488   4489   4492   4496   4499   4513   4514   4515   4516   4524   4527   4534
         4537   4549   4550   4551   4552   4559   4562   4576   4577   4578   4579   4584   4599   4603   4630
         4637   4646   4647   4648   4649   4654   4667   4668   4669   4670   4674   4684   4689   4697   4700
         4707   4716   4717   4718   4719   4728   4733   4739   4743   4747   4754   4768   4769   4770   4771
         4780   4785   4790   4801   4804   4816   4831   4832   4833   4834   4838   4844   4848   4852   4855
         4859   4863   4867   4916   4928   4941   4945   4957   4958   4959   4960   4963   4976   4977   4978
         4979   4987   4995   4998   5010   5011   5012   5013   5029   5030   5031   5032   5049   5059   5068
         5069   5070   5071   5076   5089   5090   5091   5092   5099   5102   5111   5167   5173   5176   5182
         5185   5211   5218   5227   5228   5229   5230   5235   5248   5249   5250   5251   5258   5261   5273
         5286   5290   5302   5303   5304   5305   5308   5321   5322   5323   5324   5330   5333   5340   5350
         5358   5361   5373   5374   5375   5376   5392   5393   5394   5395   5408   5414   5417   5423   5426
         5465   5474   5484   5496   5497   5498   5499   5502   5515   5516   5517   5518   5527   5535   5538
         5550   5551   5552   5553   5569   5570   5571   5572   5583   5589   5592   5598   5601   5635   5638
         5657   5659   5663   5675   5676   5677   5678   5713   5731   5735   5749   5753   5757   5771   5781
         5786   5790   5804   5805   5806   5807   5810   5817   5826   5831   5835   5849   5850   5851   5852
         5855   5873   5887   5891   5907   5911   5915   6378   6383   6387   6401   6414   6422   6430   6437
         6463   6514   6910   6922   6926   6930   6943   6949   6952   6958   6961   7021
ESCAPE   1152#
F.EAB    1413#  3884   4078   4097   4185   4204   4250   4268   4330   4349   4382   4475   4501   4538   4564
         4656   4756   4820   4946   4965   4999   5018   5078   5237   5291   5310   5362   5381   5485   5504
         5539   5558   5664   5792   5837
GETPRI   1152#
GETSWR   1152#  6003
HDCHK3   1590#  4796
HDTBL    1603#  4133   4420   4590   5878
LOOP     1397#  3459   3570   3788   4046   4152   4219   4304   4365   4440   4580   5336   5765   5812
MSG      3343#  3345   3389#  3391   3550#  3552   3659#  3661   3731#  3733   3772#  3774   3804#  3806   3997#
         3999   4113#  4115   4400#  4402   4609#  4612   4890#  4892   5194#  5196   5432#  5434   5607#  5609
         5686#  5688
MULT     1152#
NEWTST   1152#  3343   3389   3550   3659   3731   3772   3804   3997   4113   4400   4610   4890   5194   5432
         5607   5686
OFFDIR   1816#  4677   4721   4773
OFSET    1776#  4633   5055   5214
OWNTAG   1390#  1944
POP      1152#  7381   7441   7442   7827   8023   8065
PUSH     1152#  7340   7402   7404   7425   7801   7994   8045
QKSEEK   1615#  3977   4736   4856   5746   5904
QKSRT    1507#  6919
```

```
RDATA    1694#   4229
RDHDR    1558#   4797
REPORT   1152#
SCOPE    1047#   3353    3404    3565    3668    3738    3778    3827    4007    4123    4408    4623    4911    5204    5454
         5625    5705    5938    5946
SETPRI   1152#   7705
SETTRA   8095#   8104    8105    8106    8107    8109    8111    8112    8113    8114    8115    8116    8117
SETUP    1152#   3187
SKIP     1152#   3331    3382    3601    3715    3722    3743    3786    3799    3930    3966    4887    5153    5631    5639
         5681
SLASH    1152#
SPACE    1152#
STARS    1152#   1358    1369    1371    1378    1838    1885    1888    3343    3352    3389    3403    3550    3564    3659
         3667    3731    3737    3772    3777    3804    3826    3997    4006    4113    4122    4400    4407    4610    4622
         4890    4910    5194    5203    5432    5453    5607    5624    5686    5704    5930    7133    7197    7251    7330
         7397    7454    7531    7608    7623    7694    7718    7787    7840    7879    7941    7959    7982    8029    8074
         9144    9153
SWRSU    1152#   3209#
TRMTRP   8095#
TYPBIN   1152#
TYPDEC   1152#   5958
TYPNAM   1152#
TYPNUM   1152#
TYPOCS   1152#   3426    3490    3536    3700    5039    5122    6256    6994    7039
TYPOCT   1152#   7636
TYPTXT   1152#
WDATA    1642#   4057    4164    4452    4925    5270    5471
WRCHK    1740#   4315    4521    4984    5347    5524
WRHDR    1528#   4033    4139    4426    4596    5728    5884
$SCMRE   1836#
$SCMTM   1836#   1873    1874    1875    1876    1877    1878
$SESCA   1152#
$SNEWT   1152#   3343    3389    3550    3659    3731    3772    3804    3997    4113    4400    4610    4890    5194    5432
         5607    5686
$SSET    8095#   8104    8105    8106    8107    8109    8111    8112    8113    8114    8115    8116    8117
$SSETM   3225#
$SSKIP   1152#   3331    3382    3601    3715    3722    3743    3786    3799    3930    3966    4887    5153    5631    5639
         5681
.EQUAT   1001#   1042
.HEADE   1001#   1006
.SETUP   1001#   3158
.SWRHI   1001#   1017
.SWRLO   1001#   1029#
.$ACT1   1001#   1356
.$APTB   1886#
.$APTH   1001#   1367
.$APTY   1001#   7395
.$CATC   1001#   1339
.$CMTA   1001#   1836
.$DB2D   1001#   7877
.$DB2O   1001#   7838
.$EOP    1001#   5928
.$ERRO   1001#   7195
.$MULT   1001#   7980
.$RDOC   1001#   7785
.$READ   1001#   7529
```

```
.$SAVE     1001#    8027
.$SB2D     1001#    7939
.$SCOP     1001#    7131
.$SUPR     1001#    7957
.$TRAP     1001#    8072
.$TYPD     1001#    7328
.$TYPE     1001#    7249
.$TYPO     1001#    7452

. ABS.  057550      000

 ERRORS DETECTED:  0

 DSKZ:CZR6IE,DSKZ:CZR6IE.SEQ/SOL/CRF/NL:TOC/DOC=DSKM:CZR6IE.P11
 RUN-TIME: 33 25 3 SECONDS
 RUN-TIME RATIO: 273/62=4.3
 CORE USED:  30K  (59 PAGES)

 DOCUMENT PAGES:  209
```