RK06

DISK DRIVE DIAG. PART 3

MD-11-DZR6J-C

EP-DZR6J-C-DL-A

FICHE 1 OF 1

DEC 1976

digital

MADE IN USA

```
.REM %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6J-C-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 3
DATE:	DECEMBER 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

TABLE OF CONTENTS

1.0 ABSTRACT

2.0 REQUIREMENTS

2.1 HARDWARE
2.2 PRELIMINARY TESTING AND PROGRAMS

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY
3.2 XXDP
3.3 ACT/APT
3.3.1 APT ETABLE DEFINITIONS
3.4 DUAL ACCESS
3.5 MEMORY MANAGEMENT
3.6 PARITY CHECK ENABLED
3.7 BAD SECTORS
3.8 EXECUTION TIME
3.9 FAULT ISOLATION
3.10 ERROR CORRECTION & FAILURE RATE ANALYSIS
3.11 DEFAULT UNIBUS ADDRESSES & VECTORS

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING
4.2 STARTING LOCATIONS
4.3 CONSOLE SWITCH REGISTERS
4.4 SOFTWARE SWITCH REGISTER
4.5 INPUT DIALOGUE
4.6 PROGRAM EXAMPLE
4.7 HALTING THE PROGRAM

5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL
5.2 TEST DESCRIPTIONS

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION
6.2 ERROR PRINTOUT EXAMPLE

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 3 OF THE DRIVE DIAGNOSTICS TO
INSURE THAT THE DISK IS CAPABLE OF PROPERLY PERFORMING
ALL OPERATOR INTERVENTION FUNCTIONS.
ERROR DETECTION LOGIC IS CHECKED BY MANUAL & SOFTWARE ERROR FORCING.

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS PART,
PRECEEDED BY THE SUCCESSFUL RUN OF PARTS 1 & 2,
IT CAN BE ASSERTED THAT THE RK06 DRIVE WILL WORK SUCCESSFULLY
IN THE STAND-ALONE MODE. SYSTEMS INTERACTION, & ERROR RATE
ANALYSIS ARE LEFT TO OTHER PROGRAMS.

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

*****CAUTION*****

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

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

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

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

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

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFU
FOLLOWED BY THE RK06 DRIVE DIAGNOSTICS - PARTS 1 & 2.

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 SHOULD NOT BE CHAINED BY XXDP.

CHAIN MODE OPERATION (MONITOR)

BY DEFINITION, ANY PROGRAM THAT REQUIRES
OPERATOR INTERVENTION SHOULD NOT BE CHANIED.

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.
HOWEVER, IT SHOULD NOT BE RUN IN THE AUTO MODE.

AUTOMATIC MODE (MONITOR)

BY DEFINITION, ANY PROGRAM THAT REQUIRES
OPERATOR INTERVENTION SHOULD NOT BE
RUN IN THE AUTO MODE.

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

3.3.1 APT ETABLE DEFINITIONS

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

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

2. ENVIRONMENT MODE:
   BIT 7 = 1 ETABLE DOES SIZING
         = 0 PROGRAM DOES SIZING
   BIT 6 = 1 SPOOL 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

TOTAL TIME: APPROX 5 MINUTES TO DO ALL THE TESTS
(BASED ON THE PDP 11/50)

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 BUS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1334	210
CONTROLLER PRIORITY	1336	240
P-CLOCK STATUS REG	1340	172540
P-CLOCK SET BUFFER	1342	172542
P-CLOCK READ BUFFER	1344	172544
L-CLOCK STATUS REG	1346	177546
L-CLOCK INTERRUPT VECTOR	1350	100
P-CLOCK INTERRUPT VECTOR	1352	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564

TTY PRINTER BUFFER 1152 177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

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

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

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

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

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

4.1.4 PRESS 'START'

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

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

4.2 STARTING LOCATIONS

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

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

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

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH	FUNCTION
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUT
12	BYPASS DRIVE AFTER 20 ERRORS
11	INHIBIT ITERATION
10	BELL ON ERROR
9	LOOP ON ERROR

4.3.1 SW<15>

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

4.3.2 SW<14>

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

4.3.3 SW<13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).

4.3.4 SW<12>

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

4.3.5 SW<11>

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

4.3.6 SW<10>

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

4.3.7 SW<09>

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

4.4 '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 RK0[illegible] 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 3
MAINDEC-11-DZR6J-C-PB

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

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

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

                                WILL TEST DRIVES:
                                1
                                3

                                DRIVE 1

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


4.6     PROGRAM EXAMPLE

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

                UNIBUS RK06 DRIVE DIAGNOSTIC
                    PART 3
                MAINDEC-11-DZR6J-C-PB

                WILL TEST DRIVES:
                0
                1

                DRIVE 0

                DRIVE SERIAL NO. AAA
                CARTRIDGE SERIAL NO. BBB

                DRIVE 1

                DRIVE SERIAL NO. CCC
                CARTRIDGE SERIAL NO. DDD

                END PASS #1


                WILL TEST DRIVES:
                0
                1

                DRIVE 0

                DRIVE SERIAL NO. AAA
                CARTRIDGE SERIAL NO. BBB

                DRIVE 1

                DRIVE SERIAL NO. CCC
                CARTRIDGE SERIAL NO. DDD

                END PASS # 2

                   (ETC)
```

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

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

                4.7     HALTING THE PROGRAM

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

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

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

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

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

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

                OPERATOR INTERVENTION TESTS

                THESE TESTS CHECK OUT ALL THE DRIVE INTERLOCKS, FRONT PANEL
                SWITCHES AND LIGHTS.
```

THE OPERATOR IS INSTRUCTED TO PERFORM A TEST
AND TYPE A SPACE WHEN FINISHED.

OPERATOR INTERVENTION TESTS CAN BE INDIVIDUALLY
BYPASSED BY TYPING A CONTROL-E <↑E>. ONLY AT THE
BEGINNING OF EACH TEST, AS INSTRUCTED BY THE TYPEOUT.

IF THE PROGRAM DETERMINES IT WAS LOADED UNDER
ACT, APT, OPERATOR INTERVENTION TESTS WILL BE
BYPASSED UNLESS THE 'LOAD & DUMP MODE' IS BEING
USED.

THEY WILL BE BYPASSED IN 'MONITOR MODE' AS OPERATOR
INTERVENTION MAY NOT BE FEASIBLE IN OVER-NITE
TESTING.

5.2 TEST DESCRIPTIONS

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

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

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

TEST 2 SIZE THE BUSS

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

MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
DICATING THE OTHER PORT IS ACCESSED.

TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
DEFAULTED. EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
PROGRAM WILL ASSUME THE DRIVE IS PRESENT. IT WILL THEN CHECK
TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING. IF NOT, IT WILL
BE AN ERROR. IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED

```
                        ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                        NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                        NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
                        VERIFY IT WAS NOT SPECIFIED.


TEST 4  FIND NEXT DRIVE TO BE TESTED

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


TEST 5  PRINT DRIVE SERIAL NUMBER

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


TEST 6  SET VV WITH PACK COMMAND

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

TEST 7  UNLOAD DRIVE TO BE TESTED

        THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
        WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.


;*****************************************************************
OPERATOR INTERVENTION TESTS
;*****************************************************************


TEST 10 INTERLOCKS TESTS

        THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
        ARE OPERATING PROPERLY IN MESSAGE AD & THAT THE REMOVAL
        OF THE CARTRIDGE CLEARS VOLUME VALID.
        IT FURTHUR VERIFIES ALL REMAINING STATUS BITS OF
        MESSAGE A & B, WORDS 0 & 1.
        THIS TEST ALSO CHECKS THAT THE SPINDLE CANNOT BE STARTED
        WHEN THE DOOR IS OPEN OR THE CARTRIDGE IS REMOVED.
        IT ALSO VERIFIES THAT LOSS OF VOLUME VALID RESETS SACK WHICH
        ASSERTS NON EXISTENT DRIVE IN RKCS2


TEST 11 UNIT SELECT PLUG TEST

        THIS TEST VERIFIES THAT WHEN THE UNIT SELECT PLUG IS PULLED
        OUT, THE QUAL LOGIC RESETS ATTN & VOLUME VALID, THAT
        THE DRIVE DE-SELECTS & NON EXISTENT DRIVE ASSERTS.
        FURTHUR, THE OPERATOR IS ASKED TO INSERT ANY NUMBER OF
        UNIT SELECT PLUGS. THE PROGRAM WILL RESPOND BY TYPING
```

```
                        THE PLUG CODE NUMBER AS SOON AS IT IS INSERTED.
                        THIS PORTION OF THE TEST IS TERMINATED AT ANY TIME BY A CONTROL-C


                TEST 12 PORT SELECTION TESTS

                        THE OPERATOR IS ASKED TO SWITCH TO THE WRONG PORT
                        & THEN DESELECT BOTH PORTS.
                        IN BOTH CASES, NON EXISTENT DRIVE SHOULD ASSERT IN RKCS2



                TEST 13 FRONT PANEL RUN/STOP SWITCH TEST

                        THIS TEST ALLOWS THE HEADS TO LOAD. THE OPERATOR IS
                        ASKED TO VISUALLY VERIFY THE SEQUENCE OF HEADS LOADING &
                        UNLOADING BOTH MECHANICALLY & BY THE LIGHTS ON THE FRONT PANEL



                TEST 14 AC LOW DETECTION  PART 1

                        A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.

                        BATTERY RETRACT WILL BE TESTED LATER.
                        THE PROGRAM WAITS FOR AC LOW TO ASSERT IN RKMR3.
                        FROM THIS POINT, THERE IS APPROX 4 MS AVAILABLE BEFORE DC LOW ASSERTS
                        & THE INTERFACE SHUTS DOWN.
                        THE INDICATION OF DC LOW WILL BE NON-EXISTENT DRIVE ASSERTING IN RKCS2.
                        AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.


                TEST 15 CHECK NXF LOGIC

                        THIS TEST VERIFIES NON-EXECUTABLE FUNCTION (NXF) IS DETECTED
                        AS A RESULT OF DOING A SEEK WITH VOLUME VALID RESET.


                TEST 16 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

                        THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
                        THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
                        FACTORY DURING MANF.  ALL BAD SECTOR INFOR (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 INFOR OR AN ALIGNMENT PACK BY WRIT

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


                TEST 17 WRITE LOCK TEST

                        THIS TEST VERIFIES THAT DATA WRITTEN ON A SECTOR CANNOT BE
                        ALTERED ONCE THE WRITE LOCK SWITCH HAS BEEN ENABLED.
```

```
                        IT ALSO CHECKS THAT WRITE PROTECT TAKES EFFECT ONLY AT
                        SECTOR BOUNDRIES WHEN DOING CONTINUOUS WRITING ON CYL 0, HEAD 0


                TEST 20 AC LOW DETECTION PART 2

                        THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
                        AT SECTOR BOUNDRIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
                        THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
                        WHEN THE INTERFACE SHUTS DOWN.


                TEST 21 END OF PROGRAM

                        THIS IS NOT A TEST BUT A LINKAGE TO PERFORM ALL THE
                        ABOVE TESTS FOR THE NEXT DRIVE PRESENT.
                        THE NEXT TEST IS ENTERED ONLY AFTER ALL DRIVES PRESENT
                        HAVE BEEN TESTED.
                        DO NOT LOOP ON THIS 'TEST'.


                TEST 22 MULTIPLE DRIVE DETECTION TEST

                        THIS TEST IS PERFORMED ONLY ONCE AT THE END OF PASS 1
                        AND IS BYPASSED IF ONLY ONE DRIVE IS PRESENT.

                        THE MULTIPLE DRIVE DETECTION LOGIC IS TESTED BY THE FOLLOWING METHOD:

                                A. HEADS MUST BE LOADED (SECTOR PULSES REQ'D)
                                B. THE OPERATOR IS INSTRUCTED TO INSERT THE SAME UNIT SELECT
                                   PLUG NUMBER (1 PAIR AT A TIME) ON ANY 2 DRIVES
                                   TO BE TESTED
                                C. THE OPERATOR THEN DEPRESSES THE SPACE BAR TO CONTINUE THE TES
                                   OR A CONTROL-C TO EXIT THE TEST

                        THE PROGRAM VERIFIES THAT MULTIPLE DRIVE SELECT & DRIVE UNSAFE
                        BOTH SET & THAT THE DRIVE UNLOADS

                        THE OPERATOR IS ASKED TO VERIFY THAT HEADS UNLOAD FROM BOTH DRIVES


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

                        ANY TEST THAT MODIFIES STANDARD FORMATTING IS FOLLOWED BY A
                        'CLEAN UP' TEST TO PUT THOSE 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 EXAMPLE:

```
DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE DOOR OPEN
VERIFY SPINDLE DOES NOT START & HEADS DO NOT LOAD

DEPRESS SPACE BAR WHEN FINISHED

SPINDLE ON SET IN RKMR2
AFTER MANUALLY LOADING HEADS WITH DOOR OPEN

  TEST NO.   PC
  000010   015700
RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2   RKASOF
050343  100000  000000  140301  040200  000103  004000

MESSAGE A0 ERROR
AFTER MANUALLY LOADING HEADS WITH DOOR OPEN

  TEST NO.   PC
  000010   015736
EXPECT  EXPECT  EXPECT  EXPECT  EXPECT  EXPECT  EXPECT
A0      B0      A1      B1      A2      B2      B3
100143  100000  000543  000001
ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL
A0      B0      A1      B1      A2      B2      B3
050343  100000  001723  000001
RKCS1   RKCS2   RKASOF  RKER    RKDS    RKDC
040200  000103  004000  000000  140301  000000

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR
```

MESSAGE REGISTERS AD, BD, 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.

[END OF DOCUMENT]

%

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

                                ;DEFINE SYSMAC MACROS

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


                                .TITLE  UNIBUS RK6 DRIVE DIAGNOSTIC  PART 3
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY GARY PAPAZIAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        12            ABORT DRIVE AFTER 20 ERRORS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>


                                .SBTTL  SUMMARY OF STARTING LOCATIONS
                                ;*
                                ;*
                                ;*              200     DEFAULT PARAMETERS
                                ;*              220     INPUT PARAMETERS
                                ;*              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)
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
```

```
1150            000100          IR=     BIT6                    ;INPUT READY
1151            000200          OR=     BIT7                    ;OUTPUT READY
1152            000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1153            001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1154            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1155            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1156            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1157            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1158            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1159            100000          DLT=    BIT15                   ;DATA LATE ERROR
1160
1161                            .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1162
1163            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1164            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
1165            000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
1166            000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
1167            000020          FMTE=   BIT4                    ;FORMAT ERROR
1168            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1169            000100          ECH=    BIT6                    ;ECC HARD
1170            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
1171            000400          HVRC=   BIT8                    ;HEADER VRC ERROR
1172            001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1173            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1174            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1175            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
1176            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1177            040000          UNS=    BIT14                   ;DRIVE UNSAFE
1178            100000          DCK=    BIT15                   ;DATA CHECK
1179
1180                            .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1181
1182            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1183                                                            ;    THIS BIT IS RESET)
1184            000004          OFST=   BIT2                    ;DRIVE OFFSET
1185            000010          ACLO=   BIT3                    ;AC LOW
1186            000020          DCLO=   BIT4                    ;DC LOW
1187            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
1188            000100          VV=     BIT6                    ;VOLUME VALID
1189            000200          DRDY=   BIT7                    ;DRIVE READY
1190            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
1191            004000          WRL=    BIT11                   ;WRITE LOCK
1192            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
1193            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
1194            100000          SVAL=   BIT15                   ;STATUS VALID
1195
1196                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1197
1198            000017          MESMSK= 17                      ;MESSAGE MASK
1199            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
1200            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
1201            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1202            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
1203            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
1204            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1205            002000          MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
```

```
1206            004000          PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
1207            010000          PCD=    BIT12                   ;PRECOMPENSATION DELAY
1208            020000          ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
1209            040000          WRTGAT= BIT14                   ;WRITE GATE
1210            100000          RDGATE= BIT15                   ;READ GATE
1211
1212                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1213
1214            000040          D.DRA=  BIT5                    ;DRIVE AVAILABLE
1215            000100          D.VV=   BIT6                    ;VOLUME VALID
1216            000200          D.DRDY= BIT7                    ;DRIVE READY
1217            000400          D.DDT=  BIT8                    ;DRIVE TYPE (0=RK06)
1218            001000          D.FORM= BIT9                    ;DRIVE FORMAT
1219            002000          D.OFF=  BIT10                   ;OFFSET ON
1220            004000          D.WRL=  BIT11                   ;WRITE LOCK
1221            010000          D.SPIN= BIT12                   ;SPINDLE ON
1222            020000          D.PIP=  BIT13                   ;POSITIONING IN PROGRESS
1223            040000          D.DSC=  BIT14                   ;DRIVE STATUS CHANGE
1224
1225                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1226
1227            000020          D.SSP=  BIT4                    ;SERVO SIG PRESENT
1228            000040          D.HDHM= BIT5                    ;HEADS HOME
1229            000100          D.BRHM= BIT6                    ;BRUSHES HOME
1230            000200          D.DOOR= BIT7                    ;DOOR INTERLOCKED
1231            000400          D.CART= BIT8                    ;CARTRAGE INTERLOCK
1232            001000          D.SPOK= BIT9                    ;SPEED OK
1233            002000          D.FWD=  BIT10                   ;FORWARD
1234            004000          D.REV=  BIT11                   ;REVERSE
1235            010000          D.LOAD= BIT12                   ;HEADS LOADING
1236            020000          D.RTZ=  BIT13                   ;RETURN TO ZERO
1237            040000          D.UNLD= BIT14                   ;HEADS UNLOADING
1238
1239                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1240
1241            000040          D.IDAE= BIT5                    ;INVALID DISK ADDRESS ERROR:HEAD/CYL
1242            000100          D.ACLO= BIT6                    ;AC LOW
1243            000200          D.FLT=  BIT7                    ;DRIVE FAULT
1244            000400          D.NXF=  BIT8                    ;NON-EXECUTABLE FUNCTION CODE
1245            001000          D.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
1246            002000          D.SKI=  BIT10                   ;SEEK INCOMPLETE
1247            004000          D.WLE=  BIT11                   ;WRITE LOCK ERROR
1248            010000          D.SPLS= BIT12                   ;SPEED LOSS
1249            020000          D.DROT= BIT13                   ;DRIVE OFF TRACK
1250            040000          D.UNS=  BIT14                   ;R/W UNSAFE
1251
1252                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1253
1254            000020          D.SECT= BIT4                    ;SECTOR ERROR
1255            000040          D.WCUR= BIT5                    ;WRITE CURRENT AND NO WRITE GATE
1256            000100          D.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
1257            000200          D.HDFL= BIT7                    ;HEAD FAULT
1258            000400          D.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
1259            001000          D.XERR= BIT9                    ;INDEX ERROR
1260            002000          D.TIB=  BIT10                   ;TRIBIT ERROR
1261            004000          D.PLO=  BIT11                   ;PLO ERROR
```

```
1262            010000          D.NMOV= BIT12                   ;SEEK AND NO MOTION
1263            020000          D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1264            040000          D.SUNS= BIT14                   ;SERVO UNSAFE
1265
1266                            .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)
1267
1268            000007          M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1269            017760          M.CDIF= 17760                   ;CYLINDER DIFF, BYTE 10
1270            017760          M.OFST= 17760                   ;OFFSET VALUE, BYTE 10
1271            077770          M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1272
1273                            .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)
1274
1275            000003          M.ID=   3                       ;BYTE ID, ALL BYTES
1276            017760          M.CADD= 17760                   ;CYLINDER ADDRESS, BYTE 10
1277            040000          M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1278            000760          M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1279            007000          M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1280            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  010060          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                        000220          .=220
000220  000137  010050          JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING

                        000240          .=240
000240  000137  062002          JMP     0.ODT           ;ENTER ODT11

                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000244          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  024470                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
000052  120000                  .WORD   120000          ;;2)SET LOC.52 TO 120000
                        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  000454          $TSTM:  .WORD   300.    ;;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  000052                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)


                                .LIST   MD

                                ;
```

```
                        ;USE LOOP  X TO OMIT SUBCLR
                        ;
                        ;
                        ;THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD
                        ;BITS SEE A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
                        ;NOTE: A CAN BE ANY BIT COMBINATION DESIRED.
                        ;
                        ;
                        ;THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
                        ;USE     A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
                        ;USE     G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
                        ;        H=T.B2  TO READ MSG B2 & PUT INFO INTO 'CYLADD'
                        ;        I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
                        ;
                        ;USE     F=<ERROR DESCRIPTION>
                        ;
                        ;
                        ;A=CYL DIFF/OFFSET ERROR #
                        ;B=CYL ADDR ERROR #
                        ;C=<ERROR DESCRIPTION>
                        ;
                        ;
                        ;
                        ; A=WRHEAD/<CFMT!WRHEAD>
                        ; USE WRHDR      <A>,X   TO OMIT CHECKING A0, B0, A1, B1
                        ;
                        ;
                        ; A=RDHEAD/<CFMT!RDHEAD>
                        ;USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
                        ;
                        ;
                        ;
                        ;
                        ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
                        ;
                        ;
                        ;QUICK SEEK.     ENTER WITH CYL# IN RKDC
                        ;
                        ;
                        ;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
```

```
                        ;
                        ;
                        ;
                        ;A=RDDATA/<CFMT!RDDATA>
                        ;USE RDATA      <A>,X TO OMIT CKWD12
                        ;
                        ;
                        ;
                        ;A=WRTCHK/<CFMT!WRTCHK>
                        ;C=16 FOR STD ERROR MSG
                        ;C=134/135/136/137 FOR ERROR MSG USED IN 'SBOUND' ROUTINE
                        ;USE WRCHK      <A>,C,X TO OMIT CKWD12
                        ;
                        ;
                        ;MACRO TO TEST THAT WRITE CHECK OCCURRED AT SECTOR BOUNDRY
                        ;A&B=134,135 FOR WRITE PROTECT SW TEST
                        ;A&B=136,137 FOR AC LOW TEST PART 2
                        ;C=JUMP ADDR TO REPEAT TEST
                        ;
                        ;
                        ;QUICK START SPINDLE
                        ;
                        ;
                        ;ROUTINE TO ISSUE PACK COMMAND
                        ;
                        ;QUICK UNLOAD
                        ;


                                .NLIST  MD
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  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                  001210  $MAIL:                  ;;APT MAILBOX
001210  000000                  $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
```

```
1492  001212  000000          $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
1493  001214  000000          $TESTN: .WORD   ATESTN  ;;TEST NUMBER
1494  001216  000000          $PASS:  .WORD   APASS   ;;PASS COUNT
1495  001220  000000          $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
1496  001222  000000          $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
1497  001224  000000          $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
1498  001226  000000          $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1499  001230                  $ETABLE:                ;;APT ENVIRONMENT TABLE
1500  001230      000         $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1501  001231      000         $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1502  001232  000000          $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1503  001234  000000          $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1504  001236  000000          $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1505                          :*                      BITS 15-11=CPU TYPE
1506                          :*                           11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1507                          :*                           11/70=06,PDQ=07,Q=10
1508                          :*                      BIT 10=REAL TIME CLOCK
1509                          :*                      BIT  9=FLOATING POINT PROCESSOR
1510                          :*                      BIT  8=MEMORY MANAGEMENT
1511  001240      000         $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1512  001241      000         $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1513                          :*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1514                          :*                           900 NSEC CORE=001
1515                          :*                           300 NSEC BIPOLAR=002
1516                          :*                           500 NSEC MOS=003
1517  001242  000000          $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1518                          :*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1519  001244      000         $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1520  001245      000         $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1521  001246  000000          $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1522  001250      000         $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1523  001251      000         $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1524  001252  000000          $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1525  001254      000         $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1526  001255      000         $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1527  001256  000000          $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1528  001260  000000          $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1529  001262  000000          $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1530  001264  177440          $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1531  001266  000000          $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1532  001270  000000          $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1533  001272  000000          $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1534  001274  000000          $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
1535  001276  000000          $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
1536  001300  000000          $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
1537  001302  000000          $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1538  001304  000000          $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1539  001306  000000          $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1540  001310  000000          $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1541  001312  000000          $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1542  001314  000000          $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
1543  001316  000000          $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
1544  001320  000000          $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
1545  001322  000000          $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
1546  001324  000000          $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
1547  001326  000000          $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
```

```
001330  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
001332  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15

001334                          $SETEND:

        177440                  ABASE=  177440          ;DEFAULT BUSS ADDRESS
001334  000210                  RKVEC:  210             ;DEFAULT CONTROLLER INTERRUPT VECTOR
001336  000240                  RKPRI:  PR5             ;PRIORITY
001340  172540                  PKS:    172540          ;P-CLOCK STATUS REG
001342  172542                  PKSB:   172542          ;P-CLOCK SET BUFFER
001344  172544                  PKRB:   172544          ;P-CLOCK READ BUFFER
001346  177546                  LKS:    177546          ;L-CLOCK STATUS REG.

001350  000100                  LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
001352  000104                  PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.

        000114                  MEMVEC= 114             ;MEMORY PARITY VECTOR
        172100                  MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDP

001354  000000                  TRAPPC: 0               ;PC FOR MEMORY PARITY ERROR TRAP

001356  000000                  PARAM:  0               ;1 FOR 220 START, NO DEFAULT
001360  000000                  FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE

001362  000000                  DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
                                                        ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.

        000040                  SPBAR=  40              ;SPACE BAR
001364  000000                  FRCYL:  0               ;FROM CYLINDER
001366  000000                  TOCYL:  0               ;TO CYLINDER
001370  000000                  CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
001372  000000                  PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
001374  000000                  CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
001376  000000                  CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
001400  000000                  CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
001402  000000                  CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE

001404  000074                  HZ:     60.             ;60 FOR 60 CPS
                                                        ;50 FOR 50 CPS
001406  000000                  COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
                                                        ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
001410  000000                  SEC:    0               ;SECOND COUNTER
001412  000000                  TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
001414  000000                  SECNT:  0               ;SECTOR COUNT
001416  000000                  PSEC:   0               ;PREVIOUS SECTOR
001420  000000                  ESEC:   0               ;EXPECTED SECTOR
001422  000000                  SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3

001424  000001                  T1:     1               ;TIMEOUT CONSTANTS
001426  000012                  T10:    10.
001430  000062                  T50:    50.
001432  000764                  T500:   500.
001434  000144                  T100:   100.
001436  011610                  T5000:  5000.
001440  141520                  T50000: 50000.
```

```
1604
1605  001442  000077          CYL:    63.             ;CYLINDER NUMBERS USED IN
1606  001444  000177                  127.            ;CURRENT CROSSOVER TEST
1607  001446  000277                  191.
1608  001450  000377                  255.
1609  001452  000477                  319.
1610  001454  000577                  383.
1611
1612  001456  000000          TIM1:   0               ;USED IN TIMING TESTS
1613  001460  000000          TIM2:   0
1614  001462  000000          TIM3:   0
1615  001464  000000          TIM4:   0
1616
1617  001466  000000          LPCNT:  0               ;LOOP CTR USED IN CALCLK
1618  001470  000000          LPTIM:  0               ;LOOP TIME IN USEC
1619
1620  001472  000000          SUM:    0               ;LO ORDER FOR TIMING TESTS
1621  001474  000000                  0               ;HI ORDER FOR TIMING TESTS
1622  001476  000000          SUM1:   0               ;LO ORDER FOR TIMING TESTS
1623  001500  000000                  0               ;HI ORDER FOR TIMING TESTS
1624
1625  001502  000000          WD1:    0               ;ACTUAL HEADER/DATA WORD
1626  001504  000000          WD2:    0               ;EXPECTED DATA WORD
1627
1628  001506  000000          OFFERR: 0               ;SET WHEN WRITE CHECK ERROR ON OFFSET
1629
1630
1631  001510  000000          HEAD:   0               ;HEAD NUMBER
1632  001512  000000          HEADA:  0               ;HEAD # FROM H.B3, RT. JUSTIFIED
1633  001514  000000          HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
1634  001516  000000          FORMAT: 0               ;FORMAT TYPE
1635  001520  000000          FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
1636  001522  000000          WDCNT:  0               ;WORD COUNT
1637
1638  001524  000000          DATA0:  0               ;ALL 0'S
1639  001526  052525          DATA01: 52525           ;0101 PATT
1640  001530  177777          DATA1:  177777          ;ALL 1'S
1641  001532  133467          DPAT1:  133467
1642  001534  070627          DPAT2:  70627
1643
1644  001536  000000          WORD:   0               ;HEADER/DATA WORD
1645  001540  000000          HDWD:   0               ;HEADER WORD FROM RKDB
1646
1647  001542  000000          BSERR:  0               ;CANNOT READ BSE INFO WHEN SET
1648  001544  000000          LIMERR: 0               ;LIMIT DETECT ERROR FLAG
1649  001546  000000          MDSERR: 0               ;MULT DRIVE SEL ERROR FLAG
1650  001550  000000          BYPCERR:0               ;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
1651  001552  000000          CHKFLG: 0               ;WORDS TO BE TESTED
1652
1653  001554  000102          HDTAB:  .BLKW 66.       ;CALCULATED HEADER WORD TABLE
1654  001760  000102          RHTAB:  .BLKW 66.       ;FILLED AFTER READ HEADER CMD
1655  002164  000102          SRTTAB: .BLKW 66.       ;ABOVE RHTAB SORTED STARTING FORM
1656                                                  ;SECTOR 0 BY SORT ROUTINE
1657  002370  000400          BSE22H: .BLKW 256.      ;22 SECTOR HARDWARE BSE INFO.
1658  003370  000400          BSE22S: .BLKW 256.      ;22 SECTOR SOFTWARE BSE INFO.
1659  004370  000400          RDTAB:  .BLKW 256.      ;FILLED AFTER READ DATA CMD
```

```
1660
1661  005370  000000                UNLD:   0                       ;SET TO 0 IF HEADS ARE LOADED
1662                                                                ;SET TO 1 IF HEADS UNLOADED
1663  005372  000000                BADHDR: 0                       ;SET TO 0 IF FORMATTING OK
1664                                                                ;SET TO 1 IF FORMATTING ALTERED
1665  005374  000000                HPEND:  0                       ;SET TO 0 IF HALT NOT PENDING
1666                                                                ;SET TO 1 IF HALT PENDING
1667
1668                                                                ;THE ABOVE 3 FLAGS ARE USED
1669                                                                ;BY 'STOP' ROUTINE TO BRING
1670                                                                ;THE CPU TO A VALID HALT.
1671
1672
1673  005376    001     002     004 ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
1674  005401    010     020     040
1675  005404    100     200
1676                                .EVEN
1677                              ;
1678                              ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
1679                              ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
1680                              ;
1681
1682  005406  000000              HCS1:   0                       ;HOLD RKCS1
1683  005410  000000              HCS2:   0                       ;HOLD RKCS2
1684  005412  000000              HWC:    0                       ;HOLD RKWC
1685  005414  000000              HBA:    0                       ;ETC.
1686  005416  000000              HDA:    0
1687  005420  000000              HDS:    0
1688  005422  000000              HER:    0
1689  005424  000000              HASOF:  0
1690  005426  000000              HDC:    0
1691  005430  000000              HDB:    0
1692  005432  000000              HMR1:   0
1693  005434  000000              HMR2:   0
1694  005436  000000              HMR3:   0
1695  005440  000000              HPOS:   0
1696  005442  000000              HPAT:   0
1697
1698  005444  000000              TEMP1:  0                       ;TEMPORARY STORAGE.
1699  005446  000000              TEMP2:  0
1700  005450  000000              TEMP3:  0
1701  005452  000000              TEMP4:  0
1702  005454  000000              TEMP5:  0
1703                              ;
1704                              ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
1705                              ;
1706  005456  000000              H.A0:   0
1707  005460  000000              H.B0:   0
1708  005462  000000              H.A1:   0
1709  005464  000000              H.B1:   0
1710  005466  000000              H.A2:   0
1711  005470  000000              H.B2:   0
1712  005472  000000              H.A3:   0
1713  005474  000000              H.B3:   0
1714                              ;
1715                              ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
```

```
                                ;
005476  000000                  E.A0:   0
005500  000000                  E.B0:   0
005502  000000                  E.A1:   0
005504  000000                  E.B1:   0
005506  000000                  E.A2:   0
005510  000000                  E.B2:   0
005512  000000                  E.A3:   0
005514  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.
                                ;
005516  000000                  DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
005520  000000                  DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
005522  000000                  ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
005524  000000                  PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
005526  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.
005530  000000                  DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
005532  000000                  DRIV1:  0                       ;FOR DRIVE 1
005534  000000                  DRIV2:  0                       ;FOR DRIVE 2
005536  000000                  DRIV3:  0                       ;FOR DRIVE 3
005540  000000                  DRIV4:  0                       ;FOR DRIVE 4
005542  000000                  DRIV5:  0                       ;FOR DRIVE 5
005544  000000                  DRIV6:  0                       ;FOR DRIVE 6
005546  000000                  DRIV7:  0                       ;FOR DRIVE 7
                                ;
005550  000000                  LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
005552  000000                  PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
005554  000000                  DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
005556  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


005560                          $ERRTB:

                                ;ERROR 1
005560  045754                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
005562  054012                          DH1
005564  060506                          DT1
005566  061102                          DF1

                                ;ERROR 2
005570  046173                          EM5                     ;DETECTED MDS
005572  054012                          DH1
005574  060506                          DT1
005576  061102                          DF1

                                ;ERROR 3
005600  046214                          EM6                     ;DETECTED UFE
005602  054012                          DH1
005604  060506                          DT1
005606  061102                          DF1

                                ;ERROR 4
005610  046235                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
005612  054012                          DH1
005614  060506                          DT1
005616  061102                          DF1
                                ;ERROR 5
005620  046324                          EM8                     ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
005622  054012                          DH1
005624  060506                          DT1
005626  061102                          DF1

                                ;ERROR 6
005630  046400                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
005632  054012                          DH1
005634  060506                          DT1
005636  061102                          DF1

                                ;ERROR 7
005640  046454                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
005642  054012                          DH1
005644  060506                          DT1
005646  061102                          DF1
```

```
1814
1815
1816  005650  046537          ;ERROR 10
1817  005652  054012                  EM11                    ;DRA & NED BOTH SET
1818  005654  060506                  DH1
1819  005656  061102                  DT1
1820                                   DF1
1821  005660  046603          ;ERR 11
1822  005662  055407                  EM12                    ;NO RDY
1823  005664  060506                  DH27                    ;AFTER WRITE DATA CMD
1824  005666  061172                  DT1
1825                                   DF10
1826  005670  047073          ;ERR 12
1827  005672  055407                  EM21                    ;CERR SET
1828  005674  060506                  DH27
1829  005676  061172                  DT1
1830                                   DF10
1831  005700  046603          ;ERR 13
1832  005702  055357                  EM12                    ;NO RDY
1833  005704  060506                  DH26                    ;AFTER READ DATA CMD
1834  005706  061172                  DT1
1835                                   DF10
1836  005710  047073          ;ERR 14
1837  005712  055357                  EM21                    ;CERR SET
1838  005714  060506                  DH26
1839  005716  061172                  DT1
1840                                   DF10
1841  005720  046603          ;ERR 15
1842  005722  055616                  EM12                    ;NO RDY
1843  005724  060506                  DH32                    ;AFTER WRITE CHECK CMD
1844  005726  061172                  DT1
1845                                   DF10
1846  005730  052565          ;ERR 16
1847  005732  055616                  EM80                    ;WRITE CHECK ERROR SET
1848  005734  060506                  DH32                    ;AFTER WRITE CHECK CMD
1849  005736  061172                  DT1
1850                                   DF10
1851  005740  046603          ;ERR 17
1852  005742  055072                  EM12                    ;CONTR NOT RDY
1853  005744  060506                  DH18                    ;AFTER UNLD CMD
1854  005746  061172                  DT1
1855                                   DF10
1856  005750  046641          ;ERR 20
1857  005752  055072                  EM13                    ;NO ATTN
1858  005754  060506                  DH18
1859  005756  061172                  DT1
1860                                   DF10
1861  005760  052742          ;ERR 21
1862  005762  055357                  EM83                    ;DATA CHECK ERROR
1863  005764  060506                  DH26                    ;AFTER READ DATA CMD
1864  005766  061172                  DT1
1865                                   DF10
1866  005770  047073          ;ERR 22
1867  005772  055616                  EM21                    ;CERR SET
1868  005774  060506                  DH32                    ;AFTER WRITE CHECK CMD
1869  005776  061172                  DT1
                                       DF10
```

```
1870
1871  006000  047010                ;ERR 23
1872  006002  055407                        EM18            ;MSG B0 ERROR
1873  006004  060716                        DH27            ;AFTER WRITE DATA CMD
1874  006006  061306                        DT13
1875                                  DF21
1876  006010  047073                ;ERROR 24
1877  006012  055175                        EM21            ;CERR SET
1878  006014  060506                        DH21            ;AFTER SCLR
1879  006016  061172                        DT1
1880                                  DF10
1881  006020  047052                ;ERR 25
1882                                  EM20            ;MSG B1 ERROR
1883
1884
1885
```

```
1886
1887  006022  055407                    DH27
1888  006024  060716                    DT13
1889  006026  061306                    DF21
1890                          ;ERR 26
1891  006030  047010                    EM18
1892  006032  055357                    DH26          ;AFTER READ DATA CMD
1893  006034  060716                    DT13
1894  006036  061306                    DF21
1895
1896                          ;ERROR 27
1897  006040  047206                    EM24          ;VOL VALID NOT SET
1898  006042  055117                    DH19          ;AFTER PACK CMD
1899  006044  060506                    DT1
1900  006046  061172                    DF10
1901                          ;ERR 30
1902  006050  047052                    EM20          ;MSG B1 ERROR
1903  006052  055357                    DH26          ;AFTER READ DATA CMD.
1904  006054  060716                    DT13
1905  006056  061306                    DF21
1906                          ;ERR 31
1907  006060  047010                    EM18          ;MSG B0 ERROR
1908  006062  055616                    DH32          ;AFTER WRITE CHECK CMD
1909  006064  060716                    DT13
1910  006066  061306                    DF21
1911                          ;ERR 32
1912  006070  047052                    EM20          ;MSG B1 ERROR
1913  006072  055616                    DH32
1914  006074  060716                    DT13
1915  006076  061306                    DF21
1916                          ;ERR 33
1917  006100  050316                    EM44          ;VV NOT CLEARED
1918  006102  054126                    DH4           ;AFTER PACK RE-INSERTED
1919  006104  060506                    DT1
1920  006106  061172                    DF10
1921                          ;ERR 34
1922  006110  052462                    EM76          ;NO DRIVES FOUND IN DEVICE MAP
1923  006112  054012                    DH1
1924  006114  060506                    DT1
1925  006116  061102                    DF1
1926                          ;ERR 35
1927  006120  046767                    EM17          ;MSG A0 ERROR
1928  006122  054100                    DH3           ;AFTER AC SW OFF
1929  006124  060716                    DT13
1930  006126  061306                    DF21
1931                          ;ERR 36
1932  006130  047031                    EM19          ;MSG A1 ERROR
1933  006132  054100                    DH3
1934  006134  060716                    DT13
1935  006136  061306                    DF21
1936                          ;ERR 37
1937  006140  047052                    EM20          ;MSG A1 ERROR
1938  006142  054100                    DH3           ;AFTER OFFSET CMD
1939  006144  060716                    DT13
1940  006146  061306                    DF21
1941                          ;ERR 40
```

```
1942  006150  052113                EM71            ;DETECTED 10 BAD SECTORS
1943  006152  055407                DH27            ;AFTER WRITE DATA CMD
1944  006154  060506                DT1
1945  006156  061172                DF10
1946                          ;ERR 41
1947  006160  046663                EM14            ;WRONG ATTN
1948  006162  055072                DH18            ;AFTER UNLOAD CMD
1949
1950  006164  060506                DT1
1951  006166  061172                DF10
1952                          ;ERR 42
1953  006170  046710                EM15            ;DRDY NOT CLEARED
1954  006172  055072                DH18
1955
1956  006174  060506                DT1
1957  006176  061172                DF10
1958                          ;ERR 43
1959  006200  046742                EM16            ;DSC NOT SET
1960  006202  055072                DH18
1961  006204  060506                DT1
1962  006206  061172                DF10
1963                          ;ERR 44
1964  006210  047115                EM22            ;DOOR NOT CLEARED
1965  006212  054374                DH8             ;AFTER DRIVE UNLOADED & DOOR OPENED
1966  006214  060506                DT1
1967  006216  061172                DF10
1968                          ;ERR 45
1969  006220  046767                EM17            ;MSG AD ERROR
1970  006222  054374                DH8
1971  006224  060716                DT13
1972  006226  061306                DF21
1973                          ;ERR 46
1974  006230  047010                EM18            ;MSG BD ERROR
1975  006232  054374                DH8
1976  006234  060716                DT13
1977  006236  061306                DF21
1978                          ;ERR 47
1979  006240  052163                EM72            ;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
1980  006242  055407                DH27            ;AFTER WRITE DATA CMD
1981  006244  060506                DT1
1982  006246  061172                DF10
1983
1984                          ;ERR 50
1985  006250  052242                EM73            ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1986  006252  054012                DH1
1987  006254  060506                DT1
1988  006256  061102                DF1
1989                          ;ERR 51
1990  006260  053405                EM93            ;WRONG CYL# IN HEADER WORD
1991  006262  055334                DH25            ;AFTER SEEK CMD
1992  006264  060660                DT9
1993  006266  061272                DF20
1994                          ;ERR 52
1995  006270  046767                EM17            ;MSG AD ERROR
1996  006272  055407                DH27            ;AFTER WRITE DATA CMD
1997  006274  060716                DT13
```

```
1998   006276  061306                  DF21
1999
2000   006300  047031          ;ERR 53
2001   006302  055407                  EM19                    ;MSG A1 ERROR
2002   006304  060716                  DH27
2003   006306  061306                  DT13
2004                                   DF21
2005   006310  046767          ;ERR 54
2006   006312  055357                  EM17                    ;MSG A0 ERROR
2007   006314  060716                  DH26                    ;AFTER READ DATA CMD
2008   006316  061306                  DT13
2009                                   DF21
2010   006320  046641          ;ERROR 55
2011   006322  055046                  EM13                    ;NO ATTN
2012   006324  060506                  DH17                    ;AFTER RECAL CMD
2013   006326  061172                  DT1
2014                                   DF10
2015   006330  047031          ;ERR 56
2016   006332  055357                  EM19                    ;MSG A1 ERROR
2017   006334  060716                  DH26
2018   006336  061306                  DT13
2019                                   DF21
2020   006340  046767          ;ERR 57
2021   006342  055616                  EM17                    ;MSG A0 ERROR
2022   006344  060716                  DH32                    ;AFTER WRITE CHECK CMD
2023   006346  061306                  DT13
2024                                   DF21
2025   006350  047031          ;ERR 60
2026   006352  055616                  EM19                    ;MSG A1 ERROR
2027   006354  060716                  DH32
2028   006356  061306                  DT13
2029                                   DF21
2030   006360  051724          ;ERR 61
2031   006362  054012                  EM69                    ;NO DRIVES PRESENT
2032   006364  060506                  DH1
2033   006366  061102                  DT1
2034                                   DF1
2035   006370  052406          ;ERR 62
2036   006372  055407                  EM75                    ;FOUND 10 BAD CYL
2037   006374  060506                  DH27                    ;AFTER WRITE DATA CMD
2038   006376  061172                  DT1
2039                                   DF10
2040   006400  047031          ;ERR 63
2041   006402  054374                  EM19                    ;MSG A1 ERROR
2042   006404  060716                  DH8                     ;AFTER DRIVE UNLOADED & DOOR OPENED
2043   006406  061306                  DT13
2044                                   DF21
2045   006410  047052          ;ERR 64
2046   006412  054374                  EM20                    ;MSG B1 ERROR
2047   006414  060716                  DH8
2048   006416  061306                  DT13
2049                                   DF21
2050   006420  047156          ;ERR 65
2051   006422  054541                  EM23                    ;SPIN SET
2052   006424  060506                  DH11                    ;AFTER LOADING HEADS WITH DOOR OPEN
2053   006426  061172                  DT1
                                       DF10
```

```
2054
2055  006430  046767          ;ERR 66
2056  006432  054541                EM17                    ;MSG A0 ERROR
2057  006434  060716                DH11
2058  006436  061306                DT13
2059                                 DF21
2060  006440  047010          ;ERR 67
2061  006442  054541                EM18                    ;MSG B0 ERROR
2062  006444  060716                DH11
2063  006446  061306                DT13
2064                                 DF21
2065  006450  047031          ;ERR 70
2066  006452  054541                EM19                    ;MSG A1 ERROR
2067  006454  060716                DH11
2068  006456  061306                DT13
2069                                 DF21
2070  006460  047052          ;ERR 71
2071  006462  054541                EM20                    ;MSG B1 ERROR
2072  006464  060716                DH11
2073  006466  061306                DT13
2074                                 DF21
2075  006470  047244          ;ERR 72
2076  006472  054615                EM25                    ;CARTRIDGE NOT CLEARED
2077  006474  060506                DH12                    ;AFTER DISK PACK REMOVED
2078  006476  061172                DT1
2079                                 DF10
2080  006500  000000          ;ERR 73
2081  006502  000000                0
2082  006504  000000                0
2083  006506  000000                0
2084                                 0
2085  006510  046767          ;ERR 74
2086  006512  054126                EM17                    ;MSG A0 ERROR
2087  006514  060716                DH4                     ;AFTER PACK RE-INSERTED & HDS LOADED
2088  006516  061306                DT13
2089                                 DF21
2090  006520  047010          ;ERR 75
2091  006522  054126                EM18                    ;MSG B0 ERROR
2092  006524  060716                DH4
2093  006526  061306                DT13
2094                                 DF21
2095  006530  047031          ;ERR 76
2096  006532  054126                EM19                    ;MSG A1 ERROR
2097  006534  060716                DH4
2098  006536  061306                DT13
2099                                 DF21
2100  006540  047052          ;ERR 77
2101  006542  054126                EM20                    ;MSG B1 ERROR
2102  006544  060716                DH4
2103  006546  061306                DT13
2104                                 DF21
2105  006550  047156          ;ERR 100
2106  006552  054645                EM23                    ;SPIN SET
2107  006554  060506                DH13                    ;AFTER LOADING HEADS WITH CART. OUT
2108  006556  061172                DT1
2109                                 DF10
                               ;ERR 101
```

```
2110  006560  000000                    0
2111  006562  000000                    0
2112  006564  000000                    0
2113  006566  000000                    0
2114                          ;ERR 102
2115  006570  000000                    0
2116  006572  000000                    0
2117  006574  000000                    0
2118  006576  000000                    0
2119                          ;ERR 103
2120  006600  000000                    0
2121  006602  000000                    0
2122  006604  000000                    0
2123  006606  000000                    0
2124                          ;ERR 104
2125  006610  000000                    0
2126  006612  000000                    0
2127  006614  000000                    0
2128  006616  000000                    0
2129                          ;ERR 105
2130  006620  050765                    EM52          ;UNS NOT SET
2131  006622  057670                    DH64          ;AFTER MDS FOUND
2132  006624  060506                    DT1
2133  006626  061172                    DF10
2134                          ;ERR 106
2135  006630  047401                    EM28          ;VV SET
2136  006632  054762                    DH15          ;WITHOUT PACK CMD
2137  006634  060506                    DT1
2138  006636  061172                    DF10
2139                          ;ERR 107
2140  006640  047433                    EM29          ;DSC NOT SET
2141  006642  057250                    DH59          ;AFTER EVEN PARITY ISSUED
2142  006644  060506                    DT1
2143  006646  061172                    DF10
2144                          ;ERR 110
2145  006650  050554                    EM48          ;WRL NOT CLEARED
2146  006652  056555                    DH48          ;AFTER WRITE LOCK SWITCH DISABLED
2147  006654  060506                    DT1
2148  006656  061172                    DF10
2149                          ;ERR 111
2150  006660  047460                    EM30          ;ATTN NOT CLEARED
2151  006662  055007                    DH16          ;AFTER UNIT SELECT PLUG REMOVED
2152  006664  060506                    DT1
2153  006666  061172                    DF10
2154                          ;ERR 112
2155  006670  047354                    EM27          ;NED NOT SET
2156  006672  055007                    DH16
2157  006674  060506                    DT1
2158  006676  061172                    DF10
2159                          ;ERR 113
2160  006700  046641                    EM13          ;ATTN NOT SET
2161  006702  057250                    DH59          ;AFTER EVEN PARITY ISSUED
2162  006704  060506                    DT1
2163  006706  061172                    DF10
2164                          ;ERROR 114
2165  006710  051237                    EM58          ;PARITY NOT SET
```

```
2166  006712  057250                DH59
2167  006714  060506                DT1
2168  006716  061172                DF10
2169                        ;ERR 115
2170  006720  050614                EM49          ;WRL NOT SET
2171  006722  056616                DH49          ;AFTER WRITE LOCK SW ENABLED
2172  006724  060506                DT1
2173  006726  061172                DF10
2174                        ;ERROR 116
2175  006730  046603                EM12          ;CONT NOT RDY
2176  006732  055117                DH19          ;AFTER PACK CMD
2177  006734  060506                DT1
2178  006736  061172                DF10
2179                        ;ERROR 117
2180  006740  046603                EM12          ;CONT NOT RDY
2181  006742  055142                DH20          ;AFTER SEL DR CMD
2182  006744  060506                DT1
2183  006746  061172                DF10
2184                        ;ERROR 120
2185  006750  046603                EM12
2186  006752  055175                DH21          ;AFTER SUBSYS CLEAR
2187  006754  060506                DT1
2188  006756  061172                DF10
2189                        ;ERR 121
2190  006760  050650                EM50          ;WLE NOT SET
2191  006762  056656                DH50          ;AFTER WRITING WITH WRITE LOCK SET
2192  006764  060506                DT1
2193  006766  061172                DF10
2194                        ;ERR 122
2195  006770  047354                EM27          ;NED NOT SET
2196  006772  055255                DH23          ;AFTER WRONG PORT SELECTED
2197  006774  060506                DT1
2198  006776  061172                DF10
2199                        ;ERR 123
2200  007000  046767                EM17          ;MSG A0 ERROR
2201  007002  056656                DH50          ;AFTER WRITING WITH WRITE LOCK ENABLED
2202  007004  060716                DT13
2203  007006  061306                DF21
2204                        ;ERROR 124
2205  007010  046603                EM12
2206  007012  055046                DH17          ;AFTER RECAL CMD
2207  007014  060506                DT1
2208  007016  061172                DF10
2209                        ;ERR 125
2210  007020  047010                EM18          ;MSG B0 ERROR
2211  007022  056656                DH50          ;AFTER WITING WITH WRL ENABLED
2212  007024  060716                DT13
2213  007026  061306                DF21
2214                        ;ERR 126
2215  007030  047031                EM19          ;MSG A1 ERROR
2216  007032  056656                DH50
2217  007034  060716                DT13
2218  007036  061306                DF21
2219                        ;ERR 127
2220  007040  047052                EM20          ;MSG B1 ERROR
2221  007042  056656                DH50
```

```
2222  007044  060716                    DT13
2223  007046  061306                    DF21
2224                              ;ERR 130
2225  007050  047511                    EM31          ;NED NOT CLEARED
2226  007052  055474                    DH29          ;AFTER CORRECT PORT SELECTED
2227  007054  060506                    DT1
2228  007056  061172                    DF10
2229                              ;ERROR 131
2230  007060  046603                    EM12          ;NO RDY
2231  007062  055334                    DH25          ;AFTER SEEK CMD
2232  007064  060506                    DT1
2233  007066  061172                    DF10
2234                              ;ERROR 132
2235  007070  046641                    EM13          ;NO ATTN
2236  007072  055334                    DH25
2237  007074  060506                    DT1
2238  007076  061172                    DF10
2239                              ;ERR 133
2240  007100  047354                    EM27          ;NED NOT SET
2241  007102  055440                    DH28          ;AFTER BOTH PORTS DESELECTED
2242  007104  060506                    DT1
2243  007106  061172                    DF10
2244                              ;ERR 134
2245  007110  050712                    EM51          ;WRITE LOCK NOT SET SECTOR BOUNDRY
2246  007112  056656                    DH50          ;AFTER WRITING WITH WRL ENABLED
2247  007114  060530                    DT3
2248  007116  061112                    DF3
2249                              ;ERR 135
2250  007120  050712                    EM51
2251  007122  057324                    DH60          ;AFTER WRITE LOCK ENABLED WHILE WRITING
2252  007124  060530                    DT3
2253  007126  061126                    DF4
2254                              ;ERR 136
2255  007130  050712                    EM51
2256  007132  057617                    DH63          ;AFTER WRITE LOCK ENABLED FROM AC OFF
2257  007134  060530                    DT3
2258  007136  061112                    DF3
2259                              ;ERR 137
2260  007140  050712                    EM51
2261  007142  057617                    DH63
2262  007144  060530                    DT3
2263  007146  061126                    DF4
2264                              ;ERR 140
2265  007150  047542                    EM32          ;SPINDLE ON NOT SET
2266  007152  055562                    DH31          ;AFTER DRIVE MANUALLY LOADED
2267  007154  060506                    DT1
2268  007156  061172                    DF10
2269                              ;ERR 141
2270  007160  047576                    EM33          ;DRIVE NOT READY
2271  007162  056077                    DH38          ;AFTER AC POWERED UP
2272  007164  060506                    DT1
2273  007166  061172                    DF10
2274                              ;ERR 142
2275  007170  047627                    EM34
2276  007172  055562                    DH31
2277  007174  060506                    DT1
```

```
2278  007176  061172                    DF10
2279                          ;ERR 143
2280  007200  046603                    EM12          ;CONI NOT READY
2281  007202  055674                    DH34          ;AFTER ST SPIN. CMD
2282  007204  060506                    DT1
2283  007206  061172                    DF10
2284                          ;ERR 144
2285  007210  046641                    EM13          ;NO ATTN
2286  007212  055674                    DH34
2287  007214  060506                    DT1
2288  007216  061172                    DF10
2289                          ;ERR 145
2290  007220  047670                    EM35          ;HEADS NOT HOME
2291  007222  055730                    DH35          ;AFTER MANUAL UNLOAD
2292  007224  060506                    DT1
2293  007226  061172                    DF10
2294                          ;ERR 146
2295  007230  051211                    EM57          ;CERR NOT SET
2296  007232  056043                    DH37          ;AFTER TIMEOUT TO POWER DOWN
2297  007234  060506                    DT1
2298  007236  061172                    DF10
2299                          ;ERR 147
2300  007240  047767                    EM37          ;AC LOW NOT SET
2301  007242  056043                    DH37
2302  007244  060506                    DT1
2303  007246  061172                    DF10
2304                          ;ERR 150
2305  007250  050237                    EM42          ;NED NOT SET
2306  007252  056043                    DH37
2307  007254  060506                    DT1
2308  007256  061172                    DF10
2309                          ;ERROR 151
2310  007260  046603                    EM12          ;NO RDY
2311  007262  055223                    DH22          ;AFTER CLEAR CMD
2312  007264  060506                    DT1
2313  007266  061172                    DF10
2314                          ;ERR 152
2315  007270  050264                    EM43          ;AC LO NOT CLEARED
2316  007272  056077                    DH38          ;AFTER AC POWERED UP
2317  007274  060506                    DT1
2318  007276  061172                    DF10
2319                          ;ERR 153
2320  007300  050316                    EM44          ;VV NOT CLEARED
2321  007302  056077                    DH38
2322  007304  060506                    DT1
2323  007306  061172                    DF10
2324                          ;ERROR 154
2325  007310  051112                    EM55          ;ATTN NOT CLEARED
2326  007312  055223                    DH22
2327  007314  060506                    DT1
2328  007316  061172                    DF10
2329                          ;ERR 155
2330  007320  050360                    EM45          ;VV SET AFTER HDS LOADED
2331  007322  054762                    DH15          ;WITHOUT 'PACK' CMD
2332  007324  060506                    DT1
2333  007326  061172                    DF10
```

```
 2334
 2335  007330  050435          ;ERR 156
 2336  007332  056410                EM46                    ;NXF=0
 2337  007334  060506                DH45                    ;AFTER SEEK WITH VV=0
 2338  007336  061172                DT1
 2339                            DF10
 2340  007340  050514          ;ERR 157
 2341  007342  056410                EM47                    ;CYL ADDR CHANGED FROM 0
 2342  007344  060756                DH45
 2343  007346  061342                DT14
 2344                            DF22
 2345  007350  046767          ;ERR 160
 2346  007352  056410                EM17                    ;MSG A0 ERROR
 2347  007354  060716                DH45
 2348  007356  061306                DT13
 2349                            DF21
 2350  007360  047010          ;ERR 161
 2351  007362  056410                EM18                    ;MSG B0 ERROR
 2352  007364  060716                DH45
 2353  007366  061306                DT13
 2354                            DF21
 2355  007370  047031          ;ERR 162
 2356  007372  056410                EM19                    ;MSG A1 ERROR
 2357  007374  060716                DH45
 2358  007376  061306                DT13
 2359                            DF21
 2360  007400  047052          ;ERR 163
 2361  007402  056410                EM20                    ;MSG B1 ERROR
 2362  007404  060716                DH45
 2363  007406  061306                DT13
 2364                            DF21
 2365  007410  050435          ;ERR 164
 2366  007412  056447                EM46                    ;NXF NOT SET
 2367  007414  060506                DH46                    ;AFTER WRITE DATA WITH VV=0
 2368  007416  061172                DT1
 2369                            DF10
 2370  007420  051661          ;ERR 165
 2371  007422  056260                EM68                    ;CANNOT READ BSE INFO
 2372  007424  060506                DH42                    ;ON SECTORS 0, 2, 4, 6, 8
 2373  007426  061252                DT1
 2374                            DF17
 2375  007430  000000          ;ERR 166
 2376  007432  000000                0
 2377  007434  000000                0
 2378  007436  000000                0
 2379                            0
 2380  007440  051661          ;ERR 167
 2381  007442  060422                EM68
 2382  007444  060506                DH74                    ;ON SEC 10, 12....20
 2383  007446  061252                DT1
 2384                            DF17
 2385  007450  000000          ;ERR 170
 2386  007452  000000                0
 2387  007454  000000                0
 2388  007456  000000                0
 2389                            0
                                 ;ERROR 171
```

ERROR POINTER TABLE

```
2390  007460  046603                EM12                    ;NO RDY
2391  007462  055530                DH30                    ;AFTER READ HEADER CMD
2392  007464  060506                DT1
2393  007466  061172                DF10
2394                          ;ERROR 172
2395  007470  051346                EM61                    ;NXF DID NOT SET FAULT
2396  007472  056410                DH45                    ;AFTER SEEK WITH VV=0
2397  007474  060506                DT1
2398  007476  061172                DF10
2399                          ;ERROR 173
2400  007500  051414                EM63                    ;DLT SET
2401  007502  055530                DH30
2402  007504  060572                DT5
2403  007506  061236                DF15
2404                          ;ERROR 174
2405  007510  047073                EM21                    ;CERR SET
2406  007512  055530                DH30
2407  007514  060572                DT5
2408  007516  061236                DF15
2409                          ;ERR 175
2410  007520  051012                EM53                    ;UNLD NOT SET
2411  007522  057670                DH64                    ;AFTER MDS FOUND
2412  007524  060506                DT1
2413  007526  061172                DF10
2414                          ;ERR 176
2415  007530  051042                EM54                    ;CANNOT FIND MDS
2416  007532  057724                DH65                    ;AFTER SEARCHING ALL DRIVES
2417  007534  060506                DT1
2418  007536  061172                DF10
2419                          ;ERROR 177
2420  007540  047312                EM26                    ;VV NOT CLEARED
2421  007542  055007                DH16                    ;AFTER UNIT SEL PLUG REMOVED
2422  007544  060506                DT1
2423  007546  061172                DF10
2424                          ;ERROR 200
2425  007550  046603                EM12                    ;NO RDY
2426  007552  056123                DH39                    ;AFTER WRITE HEADER CMD
2427  007554  060572                DT5
2428  007556  061236                DF15
2429                          ;ERROR 201
2430  007560  047073                EM21                    ;CERR SET
2431  007562  056123                DH39
2432  007564  060572                DT5
2433  007566  061236                DF15
2434                          ;ERROR 202
2435  007570  051145                EM56                    ;UNEXP MEMORY PARITY ERROR
2436  007572  057767                DH66                    ;TEST #,PRAP PC
2437  007574  060614                DT6
2438  007576  061142                DF5
2439                          ;ERROR 203
2440  007600  047010                EM18                    ;MSG BD ERROR
2441  007602  054100                DH3                     ;AFTER AC SWITCHED OFF
2442  007604  060716                DT13
2443  007606  061306                DF21
2444                          ;ERR 204
2445  007610  051211                EM57                    ;CERR NOT SET
```

```
2446  007612  060010                DH67                    ;AFTER TIMEOUT TO ENABLE WRL
2447  007614  060506                DT1
2448  007616  061172                DF10
2449                          ;ERR 205
2450  007620  050650                EM50                    ;WRL NOT SET
2451  007622  060010                DH67
2452  007624  060506                DT1
2453  007626  061172                DF10
2454                          ;ERROR 206
2455  007630  051435                EM64                    ;WCE AT CYL 411,TRK 2, SEC 21
2456  007632  054012                DH1
2457  007634  060676                DT10
2458  007636  061162                DF7
2459                          ;ERROR 207
2460  007640  051211                EM57                    ;CERR NOT SET
2461  007642  056656                DH50                    ;AFTER WRITING WITH WRL ENABLED
2462  007644  060506                DT1
2463  007646  061172                DF10
2464                          ;ERROR 210
2465  007650  047073                EM21                    ;CERR SET
2466  007652  055334                DH25
2467  007654  060506                DT1
2468  007656  061172                DF10
2469                          ;ERR 211
2470  007660  051267                EM59                    ;CTO SET
2471  007662  052022                EM70                    ;WHILE WAITING FOR OR REC'D CONTR RDY MSG A & B BAD
2472  007664  060506                DT1
2473  007666  061206                DF12
2474                          ;ERR 212
2475  007670  051640                EM67                    ;NED SET
2476  007672  052022                EM70
2477  007674  060506                DT1
2478  007676  061206                DF12
2479                          ;ERR 213
2480  007700  046173                EM5                     ;MDS SET
2481  007702  052022                EM70
2482  007704  060506                DT1
2483  007706  061206                DF12
2484                          ;ERR 214
2485  007710  052361                EM74                    ;RTZ NOT SET
2486  007712  056233                DH41                    ;DURING RECD CMD
2487  007714  060506                DT1
2488  007716  061172                DF10
2489                          ;ERR 215
2490  007720  046767                EM17                    ;MSG A0 ERROR
2491  007722  056123                DH39                    ;AFTER WRITE HEADER CMD.
2492  007724  060716                DT13
2493  007726  061306                DF21
2494                          ;ERR 216
2495  007730  047010                EM18                    ;B0 ERROR
2496  007732  056123                DH39
2497  007734  060716                DT13
2498  007736  061306                DF21
2499                          ;ERR 217
2500  007740  047031                EM19                    ;A1 ERROR
2501  007742  056123                DH39
```

```
2502  007744  060716                    DT13
2503  007746  061306                    DF21
2504                              ;ERR 220
2505  007750  047052                    EM20                    ;B1 ERROR
2506  007752  056123                    DH39
2507  007754  060716                    DT13
2508  007756  061306                    DF21
2509                              ;ERROR 221
2510  007760  000000                    0
2511  007762  000000                    0
2512  007764  000000                    0
2513  007766  000000                    0
2514                              ;ERROR 222
2515  007770  000000                    0
2516  007772  000000                    0
2517  007774  000000                    0
2518  007776  000000                    0
2519                              ;ERROR 223
2520  010000  000000                    0
2521  010002  000000                    0
2522  010004  000000                    0
2523  010006  000000                    0
2524                              ;ERROR 224
2525  010010  000000                    0
2526  010012  000000                    0
2527  010014  000000                    0
2528  010016  000000                    0
2529                              ;ERROR 225
2530  010020  000000                    0
2531  010022  000000                    0
2532  010024  000000                    0
2533  010026  000000                    0
2534                              ;ERROR 226
2535  010030  046603                    EM12                    ;NO RDY
2536  010032  055357                    DH26                    ;AFTER READ DATA CMD
2537  010034  060506                    DTI
2538  010036  061172                    DF10
2539                              ;ERROR 227
2540  010040  047073                    EM21                    ;CERR SET
2541  010042  055357                    DH26
2542  010044  060572                    DT5
2543  010046  061236                    DF15
```

```
2544
2545                                    .SBTTL  PROGRAM SETUP
2546
2547  010050  012737  000001  001356  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
2548  010056  000402                          BR      PRGSRT          ;START PROGRAM
2549
2550  010060  105037  001356          START:  CLRB    PARAM           ;CLEAR FOR 200 START
2551  010064  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
2552  010066  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
2553  010072  012746  000000                  MOV     #PRO,-(SP)      ;PSW LOADED TO BE
2554  010076  012746  010104                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
2555  010102  000002                          RTI                     ;ENABLE ALL INTERRUPTS
2556
2557  010104  004737  033712          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
2558                                                                   ;& TURN ON KB INTERRUPT
2559
2560
2561                                    ;*** CPU PRIORITY LEVEL NOW AT 0 ***
2562                                    ;*** ANY DEVICE WHICH SETS ITS   ***
2563                                    ;*** INTERRUPT ENABLE BIT WILL   ***
2564                                    ;*** SERVICED.                   ***
2565
2566                                    ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
2567                                    ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
2568                                    ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
2569
2570                                    ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
2571
2572                                                                    ;SYSMAC 'SETUP'
2573                                    .SBTTL  INITIALIZE THE COMMON TAGS
2574                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2575  010110  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2576  010114  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2577  010116  022706  001140                  CMP     #SWR,R6 ;;DONE?
2578  010122  001374                          BNE     .-6             ;;LOOP BACK IF NO
2579  010124  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2580                                    ;;INITIALIZE A FEW VECTORS
2581  010130  012737  032020  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2582  010136  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2583  010144  012737  032300  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2584  010152  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2585  010160  012737  036130  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2586  010166  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
2587  010174  012737  031554  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2588  010202  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2589  010210  013737  024436  024430          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2590  010216  005037  001174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2591  010222  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2592  010226  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2593  010234  012737  010234  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2594  010242  012737  010242  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2595                                    ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2596                                    ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2597  010250  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2598  010254  012737  010310  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2599  010262  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
```

```
2600  010270  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2601  010276  022777  177777  170634          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2602  010304  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2603                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
2604  010306  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
2605  010310  012716  010316          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2606  010314  000002                          RTI
2607  010316  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2608  010324  012737  000174  001142          MOV     #DISPREG,DISPLAY
2609  010332  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2610
2611  010336  005037  001216                  CLR     $PASS           ;;CLEAR PASS COUNT
2612  010342  132737  000200  001231          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2613  010350  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
2614  010352  012737  001232  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2615  010360                          67$:
2616
2617  010360  012737  010424  000004  MEMPAR: MOV     #1$,ERRVEC      ;TIMEOUT VECTOR
2618  010366  012737  000340  000006          MOV     #PR7,ERRVEC+2
2619
2620  010374  012701  172100                  MOV     #MEMBAS,R1      ;ADDR OF MEM CSR
2621  010400  005011                  3$:     CLR     (R1)            ;SEE IF CAN REFERENCE
2622  010402  012711  000001                  MOV     #1,(R1)         ;SET ENABLE BIT IF YES
2623  010406  012737  031456  000114          MOV     #MEMERR,MEMVEC  ;LD MEMORY CHK VECTOR IF DONT TIMEOUT
2624  010414  012737  000340  000116          MOV     #PR7,MEMVEC+2
2625  010422  000401                          BR      2$
2626
2627  010424  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
2628  010426  062701  000002          2$:     ADD     #2,R1           ;TRY NEXT CSR
2629  010432  020127  172140                  CMP     R1,#MEMBAS+40   ;ALL TRIED?
2630  010436  001360                          BNE     3$              ;BR IN NO
2631  010440  012737  000006  000004          MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
2632  010446  005037  000006                  CLR     ERRVEC+2
2633
2634  010452  004737  024524                  JSR     PC,CLRFLG       ;CLEAR DDUMP THRU SIZFLG
2635  010456  005037  001220                  CLR     $DEVCT
2636  010462  005037  001222                  CLR     $UNIT
2637
2638                                  ;
2639                                  ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
2640                                  ;
2641
2642  010466  005737  000042          START1: TST     42
2643  010472  001014                          BNE     1$              ;BR IF AUTO
2644  010474  004737  024544                  JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
2645  010500  123727  000041  000013          CMPB    41,#13          ;13=LOADED BY XXDP
2646  010506  001010                          BNE     2$
2647  010510  005237  005516                  INC     DDUMP           ;SET RK06 DUMP MODE FLAG
2648  010514  104401  037123                  TYPE    ,MSG2           ;REPLACE DR0 PACK W/SCRATCH & DO<CR>
2649  010520  000137  010534                  JMP     ST2
2650  010524  000137  010600          1$:     JMP     ST3
2651  010530  005237  005524          2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG
2652
2653                                  ;
2654                                  ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
2655                                  ;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>
                                ;
010534  005737  001356          ST2:    TST     PARAM
010540  001002                          BNE     1$              ;BR IF 220 START
010542  000137  010632                  JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
010546  104401  037174          1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
010552  004737  024624                  JSR     PC,GDRVS        ;GET DR NOS.
010556  104401  037226                  TYPE    ,MSG4           ;BUSS ADDR
010562  004737  024764                  JSR     PC,GBA          ;GET BA
010566  104401  037273                  TYPE    ,MSG5           ;CONT INT VECTOR
010572  004737  025012                  JSR     PC,GINT         ;GET INT VECTOR
010576  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
                                ;
010600  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
010606  001007                          BNE     1$
010610  005237  005520                  INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
010614  004737  024544                  JSR     PC,TITLE
010620  104401  037410                  TYPE    ,MSG7           ;DRO NOT TSTD
010624  000402                          BR      ST4
010626  005237  005522          1$:     INC     ACT11           ;SET ACT AUTO FLAG.

010632  012737  177440  001264  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
010640  012737  000210  001334          MOV     #210,RKVEC      ;DEFAULT VALUE
010646  004737  025044                  JSR     PC,SETINT
010652  005237  005556                  INC     SIZFLG          ;DO "SIZE THE BUSS" TEST

010656  005037  005370  ST5:    CLR     UNLD            ;INITIALIZE FLAGS
010662  005037  005372          CLR     BADHDR                  ;USED IN 'STOP ROUTINE
010666  005037  005374          CLR     HPEND                   ;FOR VALID PROGRAM HALTS
010672  012737  005530  001362  MOV     #DRIVO,DRVPTR   ;SETUP
010700  005037  001220          CLR     $DEVCT          ;NO. OF DRVS DONE
010704  005037  001222          CLR     $UNIT           ;CURRENT DRV UNDER TEST
010710  012737  010756  000004  MOV     #1$,ERRVEC              ;SETUP TIMEOUT ERROR VECTOR
010716  005777  170424          TST     @LKS            ;SEE IF L-CLOCK THERE
010722  005237  005550          INC     LCLKF           ;PRESENT, SET FLAG.
010726  013700  001350          MOV     LCVEC,R0        ;VECTOR ADDR
010732  012737  011020  000004  MOV     #2$,ERRVEC
010740  005777  170374          TST     @PKS            ;SEE IF P-CLOCK THERE
010744  005237  005552          INC     PCLKF           ;PRESENT, SET FLAG
010750  013700  001352          MOV     PCVEC,R0        ;VECTOR ADDR
010754  000412                  BR      3$

010756  022626          1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
010760  012737  011024  000004  MOV     #4$,ERRVEC
010766  005777  170346          TST     @PKS            ;SEE IF P-CLOCK THERE
010772  005237  005552          INC     PCLKF           ;PRESENT, SET FLAG
010776  013700  001352          MOV     PCVEC,R0        ;VECTOR ADDR
```

```
2712  011002  005237  005554  3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
2713  011006  012720  030564          MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
2714  011012  012710  000300          MOV     #PR6,(R0)
2715  011016  000407                  BR      TST1            ;;GO TO NEXT TEST
2716
2717  011020  022626          2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
2718  011022  000767                  BR      3$
2719
2720  011024  022626          4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
2721  011026  005037  005554          CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
2722  011032  104401  037617          TYPE    ,MSG13          ;ALL TIMING TESTS BYPASSED
2723
2724
```

```
                                .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'
                                ;*
                                ;;**********************************************************************
011036  000004                  TST1:   SCOPE
011040  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
011046  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

011052  012746  000000                  MOV     #PRO,-(SP)      ;RESET PSW TO PRIORITY 0
011056  012746  011064                  MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
011062  000002                          RTI
011064                          5$:

011064  012737  011202  000004          MOV     #1$,ERRVEC              ;SETUP TIMOUT ERROR VECTOR
011072  013705  001264                  MOV     $BASE,R5        ;SETUP INDEX REG.
011076  005765  000000                  TST     RKCS1(R5)       ;REFERENCE ALL THE
011102  005765  000010                  TST     RKCS2(R5)       ;CONTROLLER REGISTERS
011106  005765  000002                  TST     RKWC(R5)
011112  005765  000004                  TST     RKBA(R5)
011116  005765  000006                  TST     RKDA(R5)
011122  005765  000012                  TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
011126  005765  000014                  TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
011132  005765  000016                  TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
011136  005765  000020                  TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
011142  005765  000024                  TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
011146  005765  000026                  TST     RKMR1(R5)
011152  005765  000034                  TST     RKMR2(R5)
011156  005765  000036                  TST     RKMR3(R5)
011162  005765  000030                  TST     RKECPS(R5)
011166  005765  000032                  TST     RKECPT(R5)

011172  012737  031370  000004          MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
011200  000404                          BR      TST2            ;;GO TO NEXT TEST

011202  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
011204  104007                          ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
011206  000137  024402                  JMP     $EOP


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

```
                                ;*      TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                ;*      MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                ;*      DICATING THE OTHER PORT IS ACCESSED.
                                ;*
                                ;;*****************************************************************
011212  000004                  TST2:   SCOPE
011214  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
011222  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

011226  005237  001550                  INC     BYPCERR         ;DO NOT TEST CERR IN FRDY

011232  132737  000200  001231          BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
011240  001002                          BNE     14$             ;BR IF YES
011242  000137  011356                  JMP     12$             ;ELSE DO NORM SIZING OR VERIFY

011246  104401  037523          14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
011252  005037  005526                  CLR     DRIVS           ;# OF DRIVES PRESENT
011256  005000                          CLR     R0              ;DRV ADDR
011260  012701  005530                  MOV     #DRIVD,R1       ;DRV FLAG
011264  013702  001266                  MOV     $DEVM,R2        ;APT DEVICE MAP

011270  032702  000001          15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
011274  001410                          BEQ     16$             ;BR IF NO
011276  005237  005526                  INC     DRIVS           ;ELSE INCR DRIVE COUNT
011302  005211                          INC     (R1)            ;& SET DRIVE PRESENT FLAG
011304  104401  001205                  TYPE    ,$CRLF
011310  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DRIVE #
011312  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
011314     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
011315     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

011316  005721          16$:    TST     (R1)+           ;ADV POINTER TO NEXT FLAG
011320  005200                  INC     R0              ;INC DRIVE #
011322  022700  000010          CMP     #8.,R0          ;ALL 8 TESTED?
011326  001402                  BEQ     17$             ;BR IF YES

011330  006002                  ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
011332  000756                  BR      15$             ;& TRY AGAIN

011334  005737  005526  17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
011340  001402                  BEQ     18$             ;BR IF NO
011342  000137  012276          JMP     NUDRV           ;ELSE EXIT TEST

011346  104034          18$:    ERROR   34              ;NO DRIVES FOUND IN $DEVM
011350  000000                  HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
011352  000137  010656          JMP     ST5             ;TO TRY AGAIN

011356  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR

011364  013737  001426  005444          MOV     T10,TEMP1       ;SET TIMEOUT
011372  004737  025062                  JSR     PC,FRDY         ;FIND RDY
```

```
011376  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR


011400  005737  005556                  TST     SIZFLG
011404  001562                          BEQ     TST3            ;;DO NOT SIZE, GOTO NEXT TEST
011406  104401  037523                  TYPE    ,MSG10          ;WILL TEST DRIVES
011412  005037  005526                  CLR     DRIVS           ;# OF DRIVES PRESENT
011416  005000                          CLR     R0              ;DRV ADDR
011420  012701  005530                  MOV     #DRIV0,R1       ;DRV FLAG
011424                          1$:
011424  104415                          SCOP1
011426  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR


011432  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
011440  013737  001426  005444          MOV     T10,TEMP1       ;SET TIMEOUT
011446  004737  025062                  JSR     PC,FRDY         ;FIND RDY
011452  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
011454  010065  000010                  MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
011460  012765  000001  000000          MOV     #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
011466  013737  001426  005444          MOV     T10,TEMP1
011474  004737  025062                  JSR     PC,FRDY         ;FIND RDY
011500  104117                          ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
011502  032737  100000  005406          BIT     #CERR,HCS1
011510  001046                          BNE     2$
011512  013737  005434  005444          MOV     HMR2,TEMP1
011520  042737  177770  005444          BIC     #^C<DRVMSK>,TEMP1
011526  020037  005444                  CMP     R0,TEMP1        ;S/B SAME
011532  001016                          BNE     3$
011534  005700                          TST     R0
011536  001003                          BNE     4$
011540  005737  005520                  TST     DDPCH           ;SEE IF XXDP CHAIN MODE
011544  001014                          BNE     5$
011546  005237  005526          4$:     INC     DRIVS           ;INC DRIVE COUNT.
011552  005211                          INC     (R1)            ;SET DRIVE PRESENT FLAG
011554  104401  001205                  TYPE    ,$CRLF
011560  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;TYPE DR #
011562  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
011564     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
011565     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
011566  000403                          BR      5$


011570  004737  025570          3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
011574  104001                          ERROR   1               ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #


011576  005721                  5$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR. FLAG
011600  005200                          INC     R0              ;INC DR #
011602  022700  000010                  CMP     #8.,R0
011606  001306                          BNE     1$              ;MORE LEFT.
011610  005737  005526                  TST     DRIVS
011614  001054                          BNE     10$
011616  104061                          ERROR   61              ;NO DRIVES SEEN
011620  000000                          HALT                    ;SETUP & PRESS 'CONTINUE'
011622  000137  010656                  JMP     ST5


011626  032737  001000  005410  2$:     BIT     #MDS,HCS2
```

```
2893  011634  001015                    BNE     6$
2894  011636  032737  000400  005410    BIT     #UFE,HCS2
2895  011644  001015                    BNE     7$
2896  011646  032737  000001  005420    BIT     #DRA,HDS
2897  011654  001015                    BNE     8$
2898  011656  032737  010000  005410    BIT     #NED,HCS2
2899  011664  001424                    BEQ     9$
2900  011666  000743                    BR      5$
2901
2902  011670  004737  025570    6$:     JSR     PC,BYP          ;TYPE BYP DR #
2903  011674  104002                    ERROR   2               ;MDS DETECTED
2904  011676  000737                    BR      5$
2905
2906  011700  004737  025570    7$:     JSR     PC,BYP
2907  011704  104003                    ERROR   3               ;UFE DETECTED
2908  011706  000733                    BR      5$
2909
2910  011710  032737  010000  005410  8$:     BIT     #NED,HCS2
2911  011716  001713                    BEQ     4$
2912  011720  104401  037730            TYPE    ,MSG15          ;DRV#
2913  011724  010046                    MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2914                                                            ;;TYPE DR#
2915  011726  104403                    TYPOS                   ;;GO TYPE--OCTAL ASCII
2916  011730     001                    .BYTE   1               ;;TYPE 1 DIGIT(S)
2917  011731     000                    .BYTE   0               ;;SUPPRESS LEADING ZEROS
2918  011732  104010                    ERROR   10              ;DRA & NED BOTH SET
2919  011734  000720                    BR      5$
2920
2921  011736  004737  025570    9$:     JSR     PC,BYP
2922  011742  104004                    ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
2923  011744  000714                    BR      5$
2924  011746  000137  012276    10$:    JMP     NUDRV
2925
2926                                    ;;*****************************************************************
2927                                    ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
2928                                    ;*
2929                                    ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
2930                                    ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
2931                                    ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
2932                                    ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
2933                                    ;*      TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
2934                                    ;*      BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
2935                                    ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
2936                                    ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
2937                                    ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
2938                                    ;*      VERIFY IT WAS NOT SPECIFIED.
2939                                    ;*
2940                                    ;;*****************************************************************
2941  011752  000004            TST3:   SCOPE
2942  011754  012737  000001  001174    MOV     #1,$TIMES       ;;DO 1 ITERATION
2943  011762  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
2944  011766  005000                    CLR     R0              ;DRIVE ADDR
2945  011770  012701  005530            MOV     #DRIV0,R1       ;DRIVE FLAG
2946  011774                    1$:
2947  011774  104415                    SCOP1
2948  011776  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
```

```
 2949
 2950  012002  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
```

```
 2951  012010  013737  001426  005444          MOV     T10,TEMP1       ;SET TIME OUT
 2952  012016  004737  025062                  JSR     PC,FRDY         ;FIND RDY
 2953  012022  104120                          ERROR   120             ;NO RDY AFTER SCLR
 2954  012024  010065  000010                  MOV     R0,RKCS2(R5)    ;DRV ADDR
 2955  012030  012765  000001  000000          MOV     #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
 2956  012036  013737  001426  005444          MOV     T10,TEMP1
 2957  012044  004737  025062                  JSR     PC,FRDY         ;FIND RDY
 2958  012050  104117                          ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
 2959  012052  032737  100000  005406          BIT     #CERR,HCS1
 2960  012060  001036                          BNE     2$
 2961  012062  013737  005434  005444          MOV     HMR2,TEMP1
 2962  012070  042737  177770  005444          BIC     #^C<DRVMSK>,TEMP1
 2963  012076  020037  005444                  CMP     R0,TEMP1        ;S/B SAME
 2964  012102  001010                          BNE     3$
 2965  012104  005711                  11$:    TST     (R1)
 2966  012106  001417                          BEQ     5$
 2967  012110  005721                  4$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR FLAG
 2968  012112  005200                          INC     R0              ;INC DR#
 2969  012114  022700  000010                  CMP     #8.,R0
 2970  012120  001325                          BNE     1$              ;MORE LEFT
 2971  012122  000467                          BR      TST4            ;;GO TO NEXT TEST
 2972
 2973  012124  004737  025570          3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
 2974  012130  104001                          ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
 2975  012132  005711                          TST     (R1)
 2976  012134  001765                          BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
 2977  012136  005337  005526          12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
 2978  012142  005011                          CLR     (R1)            ;CLEAR DRIVE FLAG
 2979  012144  000761                          BR      4$
 2980
 2981  012146  004737  025570          5$:     JSR     PC,BYP
 2982  012152  104005                          ERROR   5               ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
 2983  012154  000755                          BR      4$
 2984
 2985  012156  032737  001000  005410  2$:     BIT     #MDS,HCS2
 2986  012164  001027                          BNE     6$
 2987  012166  032737  000400  005410          BIT     #UFE,HCS2
 2988  012174  001027                          BNE     7$
 2989  012176  032737  000001  005420          BIT     #DRA,HDS
 2990  012204  001005                          BNE     8$
 2991  012206  032737  010000  005410          BIT     #NED,HCS2
 2992  012214  001423                          BEQ     9$
 2993  012216  000404                          BR      10$
 2994  012220  032737  010000  005410  8$:     BIT     #NED,HCS2
 2995  012226  001726                          BEQ     11$
 2996  012230  005711                  10$:    TST     (R1)
 2997  012232  001726                          BEQ     4$
 2998
 2999  012234  004737  025570                  JSR     PC,BYP          ;TYPE BYPASS DRIVE#
 3000  012240  104006                          ERROR   6
 3001  012242  000735                          BR      12$
 3002
 3003  012244  004737  025570          6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
 3004  012250  104002                          ERROR   2               ;MDS DETECTED
 3005  012252  000762                          BR      8$
 3006
```

```
3007  012254  004737  025570          7$:     JSR     PC,BYP
3008  012260  104003                          ERROR   3               ;UFE DETECTED
3009  012262  000756                          BR      8$
3010
3011  012264  004737  025570          9$:     JSR     PC,BYP
3012  012270  104004                          ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
3013  012272  000752                          BR      8$
3014
3015  012274  001237                          BNE     1$              ;BRANCH IF MORE LEFT.
3016
3017
3018                                          ;
3019                                          ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
3020                                          ;DRIVE PRESENT
3021                                          ;
3022                                          ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
3023                                          ;UNDER TEST
3024
3025  012276  005037  001550          NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
3026                                                                  ;TEST CERR IN 'FRDY'
3027
3028                                  ;;**************************************************************
3029                                  ;*TEST 4        FIND NEXT DRIVE TO BE TESTED
3030                                  ;*
3031                                  ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
3032                                  ;*      ADDRESS IN '$UNIT'.
3033                                  ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
3034                                  ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
3035                                  ;*
3036                                  ;;**************************************************************
3037  012302  000004                  TST4:   SCOPE
3038  012304  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3039  012312  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3040  012316  012737  000004  001214          MOV     #$TN-1,$TESTN
3041  012324  012737  000004  001102          MOV     #$TN-1,$TSTNM
3042
3043  012332  005737  005526                  TST     DRIVS           ;ANY DRIVES PRESENT?
3044  012336  001004                          BNE     4$              ;YES BRANCH
3045  012340  104401  040432                  TYPE    ,MSG27          ;ALL DRIVES TESTED
3046  012344  000137  024402                  JMP     $EOP            ;NO, GO TO END
3047
3048  012350  013701  001362          4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
3049  012354  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
3050  012360  001402                          BEQ     2$              ;YES, BRANCH
3051  012362  005237  001222          1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
3052  012366  005721                  2$:     TST     (R1)+           ;IS DRIVE PRESENT?
3053  012370  001774                          BEQ     1$              ;NO,FIND NEXT DRIVE PRESENT
3054  012372  005737  005520                  TST     DDPCH           ;DDP CHAIN MODE?
3055  012376  001403                          BEQ     3$              ;NO, BRANCH
3056  012400  005737  001222                  TST     $UNIT           ;YES, IS IT DRIVE 0?
3057  012404  001766                          BEQ     1$              ;IF YES, DON'T TEST DR 0
3058  012406  010137  001362          3$:     MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT DR. FLAG
3059  012412  104401  037730                  TYPE    ,MSG15          ;"DRIVE"
3060  012416  013700  001222                  MOV     $UNIT,R0
3061  012422  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3062                                                                  ;;DRIVE #
```

```
3063  012424  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
3064  012426  001                           .BYTE   1               ;;TYPE 1 DIGIT(S)
3065  012427  000                           .BYTE   0               ;;SUPPRESS LEADING ZEROS
3066
3067  012430  104401  001205                TYPE    ,$CRLF
3068
3069  012434                        PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
3070                                ;;**************************************************************
3071                                ;*TEST 5         PRINT DRIVE SERIAL NUMBER
3072                                ;*
3073                                ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
3074                                ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
3075                                ;*
3076                                ;;**************************************************************
3077  012434  000004                TST5:   SCOPE
3078  012436  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
3079  012444  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
3080
3081  012450  005737  001216                TST     $PASS
3082  012454  001046                        BNE     TST6            ;;GO TO NEXT IF NOT FIRST PASS
3083  012456  004737  026772                JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
3084  012462  104024                        ERROR   24              ;CERR AFTER SCLR
3085
3086  012464  104401  037742                TYPE    ,MSG16          ;DRIVE SERIAL NO.
3087  012470  012765  000003  000026        MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
3088  012476  004737  026420                JSR     PC,GSTAT        ;GET STATUS
3089  012502  013701  005434                MOV     HMR2,R1         ;GET SERIAL #
3090  012506  012704  035414                MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
3091  012512  010446                        MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
3092  012514  012703  000003                MOV     #3,R3           ;SETUP CHAR COUNT
3093  012520  006101                        ROL     R1              ;INITIALIZE BIT POSITIONS
3094  012522  006101                        ROL     R1
3095  012524  006101                1$:     ROL     R1              ;GET NEXT 4 BITS
3096  012526  006101                        ROL     R1
3097  012530  006101                        ROL     R1
3098  012532  006101                        ROL     R1
3099  012534  010100                        MOV     R1,R0           ;GET WORKING COPY
3100  012536  042700  177760                BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
3101  012542  052700  000060                BIS     #60,R0          ;CONVERT TO ASCII DIGIT
3102  012546  110024                        MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
3103  012550  005303                        DEC     R3
3104  012552  001364                        BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
3105  012554  105014                        CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR
3106
3107  012556  004737  035662                JSR     PC,$SUPRS       ;TYPE
3108  012562  104401  001205                TYPE    ,$CRLF
3109  012566  104401  001205                TYPE    ,$CRLF
3110
3111                                ;;**************************************************************
3112                                ;*TEST 6         SET VV WITH PACK COMMAND
3113                                ;*
3114                                ;*      IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
3115                                ;*
3116                                ;;**************************************************************
3117  012572  000004                TST6:   SCOPE
3118  012574  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
3119  012602  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
3120
3121  012606  004737  026772                JSR     PC,SUBCLR
3122  012612  104024                        ERROR   24              ;CERR AFTER SCLR
3123
3124  012614  032737  000100  005434        BIT     #D.VV,HMR2
3125  012622  001024                        BNE     TST7            ;;GO TO NEXT TEST IF VV SET
3126
3127  012624  104415                        SCOP1
3128  012626  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
3129
3130  012632  004737  026772                JSR     PC,SUBCLR
3131  012636  104024                        ERROR   24              ;CERR AFTER SCLR
3132
3133  012640  012765  000003  000000        MOV     #PACK,RKCS1(R5) ;CMD TO SET VV
3134  012646  012737  000010  005444        MOV     #10,TEMP1
3135  012654  004737  025062                JSR     PC,FRDY         ;FIND RDY
3136  012660  104116                        ERROR   116             ;RDY NOT SET AFTER PACK CMD
3137
3138  012662  032737  000100  005434        BIT     #D.VV,HMR2
3139  012670  001001                        BNE     TST7            ;;GO TO NEXT TEST IF VV NOW SET
3140  012672  104027                        ERROR   27              ;PACK DID NOT SET VV
3141
3142                                ;;*****************************************************************
3143                                ;*TEST 7        UNLOAD DRIVE TO BE TESTED
3144                                ;*
3145                                ;*      THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
3146                                ;*      WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.
3147                                ;*
3148                                ;;*****************************************************************
3149  012674  000004                TST7:   SCOPE
3150  012676  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
3151  012704  012706  001100                MOV     #STACK,SP
3152
3153  012710  005237  005370                INC     UNLD            ;USED TO CHECK FOR VALID HALT
3154
3155  012714  004737  026772                JSR     PC,SUBCLR
3156  012720  104024                        ERROR   24              ;CERR AFTER SCLR
3157
3158  012722  012765  000007  000000        MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
3159  012730  013737  001426  005444        MOV     T10,TEMP1
3160  012736  004737  025062                JSR     PC,FRDY         ;FIND CONTR RDY
3161  012742  104017                        ERROR   17              ;NO RDY AFTER UNLD CMD
3162  012744  004737  025344                JSR     PC,TSTATN
3163  012750  104020                        ERROR   20              ;NO ATTN AFTER UNLOAD CMD
3164
3165
3166                                .SBTTL OPERATOR INTERVENTION TESTS
3167
3168                                ;;*****************************************************************
3169                                ;*TEST 10       INTERLOCKS TESTS
3170                                ;*
3171                                ;*      THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
3172                                ;*      ARE OPERATING PROPERLY IN MESSAGE A0 & THAT THE REMOVAL
3173                                ;*      OF THE CARTRIDGE CLEARS VOLUME VALID.
3174                                ;*      IT FURTHUR VERIFIES ALL REMAINING STATUS BITS OF
```

```
                                        ;*      MESSAGE A & B, WORDS 0 & 1.
                                        ;*      THIS TEST ALSO CHECKS THAT THE SPINDLE CANNOT BE STARTED
                                        ;*      WHEN THE DOOR IS OPEN OR THE CARTRIDGE IS REMOVED.
                                        ;*      IT ALSO VERIFIES THAT LOSS OF VOLUME VALID RESETS SACK WHICH
                                        ;*      ASSERTS NON EXISTENT DRIVE IN RKCS2
                                        ;*
                                        ;;*****************************************************************
012752  000004                        TST10:  SCOPE
012754  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
012762  012706  001100                        MOV     #STACK,SP       ;RESET STACK PTR
                                        
012766  004737  026772                        JSR     PC,SUBCLR
012772  104024                                ERROR   24              ;CERR AFTER SCLR
                                        
012774  005237  001550                        INC     BYPCERR         ;DONT DO CKCERR ROUTINE
                                        
013000  104401  037445                        TYPE    ,MSG8           ;INTERLOCKS TEST
013004  104401  043041                        TYPE    ,MSG52          ;CONT-E TO EXIT OR SPACE TO CONTINUE
013010  004737  030746                        JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
013014  000137  013630                        JMP     B$              ;RET HERE FOR CONT-E
                                        
013020  104401  042336                        TYPE    ,MSG47          ;VERIFY DOOR CANNOT BE OPENED
013024  104401  041265                        TYPE    ,MSG37          ;DEPRESS SPACE WHEN DONE
013030  004737  031006                        JSR     PC,GETSP        ;GET SPACE
                                        
013034  104401  044637                        TYPE    ,MSG65          ;DEPRESS RUN/STOP SW TO 'STOP'
013040  104401  040570                        TYPE    ,MSG30          ;OPEN DOOR & LEAVE IT OPEN
013044  104401  041265                        TYPE    ,MSG37          ;DEPRESS SPACE BAR  WHEN DONE
013050  004737  031006                        JSR     PC,GETSP        ;INPUT  SPACE
013054  012765  000001  000026                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
013062  004737  026420                        JSR     PC,GSTAT
013066  032737  000200  005434                BIT     #D.DOOR,HMR2
013074  001401                                BEQ     1$
013076  104044                                ERROR   44              ;DOOR STATUS BIT NOT CLEARED
                                        
013100  012737  000140  005476  1$:           MOV     #<D.DRA!D.VV>,E.A0  ;EXPECTED A0
013106  005037  005500                        CLR     E.B0
013112  012737  000540  005502                MOV     #<D.HDHM!D.BRHM!D.CART>,E.A1
013120  012737  000001  005504                MOV     #1,E.B1
                                        
013126  004737  025604                        JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
013132  000000                                .WORD   0!0!0           ;& MSGS SPECIFIED HERE
013134  104045                                ERROR   45              ;MSG A0 ERROR AFTER DRIVE UNLOADED & DOOR OPENED
013136  104046                                ERROR   46              ;MSH B0 ERROR
013140  104063                                ERROR   63              ;MSG A1 ERROR
013142  104064                                ERROR   64              ;MSG B1 ERROR
                                        
013144  004737  026772                        JSR     PC,SUBCLR
013150  104024                                ERROR   24              ;CERR AFTER SCLR
                                        
013152  104401  040720                        TYPE    ,MSG33          ;PRESS 'RUN-STOP' TO 'RUN'
013156  104401  041005                        TYPE    ,MSG34          ;VERIFY DOES NOT START
013162  104401  041265                        TYPE    ,MSG37
013166  004737  031006                        JSR     PC,GETSP        ;GET SPACE FROM TTY
                                        
013172  004737  026420                        JSR     PC,GSTAT
```

```
3231  013176  032737  010000  005434          BIT     #D.SPIN,HMR2
3232  013204  001401                          BEQ     2$
3233  013206  104065                          ERROR   65                  ;SPIN SET IN MSGA0
3234
3235  013210  012737  000140  005476  2$:     MOV     #<D.DRA!D.VV>,E.A0  ;EXPECTED MSG A0
3236  013216  005037  005500                  CLR     E.B0
3237  013222  012737  000540  005502          MOV     #<D.HDHM!D.BRHM!D.CART>,E.A1
3238  013230  012737  000001  005504          MOV     #1,E.B1
3239
3240  013236  004737  025604                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
3241  013242  000000                          .WORD   0!0!0               ;& MSGS SPECIFIED HERE
3242  013244  104066                          ERROR   66                  ;MSG A0 ERROR AFTER ATTEMPT TO START SPIN & DOOR OPEN
3243  013246  104067                          ERROR   67                  ;MSH B0 ERROR
3244  013250  104070                          ERROR   70                  ;MSG A1 ERROR
3245  013252  104071                          ERROR   71                  ;MSG B1 ERROR
3246
3247  013254  004737  026772                  JSR     PC,SUBCLR
3248  013260  104024                          ERROR   24                  ;CERR AFTER SCLR
3249
3250  013262  104401  044637                  TYPE    ,MSG65              ;DEPRESS 'RUN-STOP' SW TO STOP
3251  013266  104401  040660                  TYPE    ,MSG32              ;REMOVE PACK & CLOSE DOOR
3252  013272  104401  041265                  TYPE    ,MSG37
3253  013276  004737  031006                  JSR     PC,GETSP            ;GET SPACE FROM TTY
3254
3255  013302  012765  000001  000026          MOV     #1,RKMR1(R5)        ;SELECT WORD 1
3256  013310  004737  026420                  JSR     PC,GSTAT
3257  013314  032737  000400  005434          BIT     #D.CART,HMR2
3258  013322  001401                          BEQ     3$
3259  013324  104072                          ERROR   72                  ;CARTRIDGE STATUS BIT NOT RESET
3260
3261
3262  013326  004737  026772          3$:     JSR     PC,SUBCLR
3263  013332  104024                          ERROR   24                  ;CERR AFTER SCLR
3264
3265  013334  104401  041073                  TYPE    ,MSG35              ;PRESS 'RUN-STOP' TO 'RUN'
3266  013340  104401  041005                  TYPE    ,MSG34              ;VERIFY DOES NOT START
3267  013344  104401  041265                  TYPE    ,MSG37
3268  013350  004737  031006                  JSR     PC,GETSP            ;GET SPACE FROM TTY
3269
3270  013354  004737  026420                  JSR     PC,GSTAT
3271  013360  032737  010000  005434          BIT     #D.SPIN,HMR2
3272  013366  001401                          BEQ     5$
3273  013370  104100                          ERROR   100                 ;SPIN SET IN MSG A0
3274
3275  013372  104401  044637          5$:     TYPE    ,MSG65              ;PRESS 'RUN-STOP' TO 'STOP'
3276  013376  104401  041170                  TYPE    ,MSG36              ;INSERT CARTRIDGE & CLOSE DOOR
3277  013402  104401  045222                  TYPE    ,MSG70              ;VERIFY HEADS LOAD
3278  013406  104401  045145                  TYPE    ,MSG69              ;DEPRESS SPACE WHEN READY GOES ON
3279  013412  004737  031006                  JSR     PC,GETSP            ;GET SPACE FROM TTY
3280  013416  005037  005370                  CLR     UNLD                ;FOR VALID HALT
3281
3282  013422  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3283  013430  005065  000026                  CLR     RKMR1(R5)           ;SELECT WORD 0
3284  013434  004737  026420                  JSR     PC,GSTAT
3285  013440  032737  000100  005434          BIT     #D.VV,HMR2
3286  013446  001401                          BEQ     7$
```

```
3287  013450  104033                          ERROR   33              ;VV NOT CLEARED AFTER PACK RE-INSERTED
3288
3289  013452  012737  050240  005476  7$:     MOV     #<D.DSC!D.SPIN!D.DRDY!D.DRA>,E.A0       ;EXPECTED A0
3290  013460  005037  005500                  CLR     E.B0
3291  013464  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.SSP!D.BRHM!D.DOOR>,E.A1
3292  013472  012737  000001  005504          MOV     #1,E.B1
3293
3294  013500  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
3295  013504  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
3296  013506  104074                          ERROR   74              ;MSG A0 ERROR AFTER PACK REINSERTED & HEADS LOADED
3297  013510  104075                          ERROR   75              ;MSH B0 ERROR
3298  013512  104076                          ERROR   76              ;MSG A1 ERROR
3299  013514  104077                          ERROR   77              ;MSG B1 ERROR
3300
3301  013516  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3302  013524  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE #
3303  013532  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
3304  013540  013737  001426  005444          MOV     T10,TEMP1
3305  013546  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
3306  013552  104116                          ERROR   116             ;CONTR NOT RDY
3307
3308  013554  032737  000100  005434          BIT     #D.VV,HMR2
3309  013562  001001                          BNE     64$
3310  013564  104027                          ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
3311  013566                          64$:
3312  013566  005237  005370                  INC     UNLD            ;FOR VALID HALT
3313
3314  013572  004737  026772                  JSR     PC,SUBCLR
3315  013576  104024                          ERROR   24              ;CERR AFTER SCLR
3316
3317  013600  012765  000007  000000          MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
3318  013606  013737  001426  005444          MOV     T10,TEMP1
3319  013614  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
3320  013620  104017                          ERROR   17              ;NO RDY AFTER UNLD CMD
3321  013622  004737  025344                  JSR     PC,TSTATN
3322  013626  104020                          ERROR   20              ;NO ATTN AFTER UNLOAD CMD
3323  013630                          8$:
3324
3325                                  ;;*****************************************************************
3326                                  ;*TEST 11       UNIT SELECT PLUG TEST
3327                                  ;*
3328                                  ;*      THIS TEST VERIFIES THAT WHEN THE UNIT SELECT PLUG IS PULLED
3329                                  ;*      OUT, THE QUAL LOGIC RESETS ATTN & VOLUME VALID, THAT
3330                                  ;*      THE DRIVE DE-SELECTS & NON EXISTENT DRIVE ASSERTS.
3331                                  ;*      FURTHUR, THE OPERATOR IS ASKED TO INSERT ANY NUMBER OF
3332                                  ;*      UNIT SELECT PLUGS. THE PROGRAM WILL RESPOND BY TYPING
3333                                  ;*      THE PLUG CODE NUMBER AS SOON AS IT IS INSERTED.
3334                                  ;*      THIS PORTION OF THE TEST IS TERMINATED AT ANY TIME BY A CONT-E
3335                                  ;*
3336                                  ;;*****************************************************************
3337  013630  000004                  TST11:  SCOPE
3338  013632  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3339  013640  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR
3340
3341  013644  004737  026772                  JSR     PC,SUBCLR
3342  013650  104024                          ERROR   24              ;CERR AFTER SCLR
```

```
3343
3344  013652  104401  040016                  TYPE    ,MSG18          ;UNIT SELECT PLUG TEST
3345  013656  104401  043041                  TYPE    ,MSG52          ;CONT-E TO EXIT OR SPACE TO CONTINUE
3346  013662  004737  030746                  JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
3347  013666  000137  014500                  JMP     12$             ;RETURN HERE FOR CONT-E
3348
3349  013672  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY TO GET DSC & ATTN
3350  013700  004737  026420                  JSR     PC,GSTAT
3351  013704  032737  001000  005436          BIT     #D.PAR,HMR3     ;SEE IF PARITY SET
3352  013712  001001                          BNE     2$              ;BR IF YES
3353  013714  104114                          ERROR   114             ;PARITY BIT NOT SET AFTER SEL DRIVE CMD
3354                                                                  ;WITH EVEN PARITY ISSUED.
3355  013716  032737  040000  005434  2$:     BIT     #D.DSC,HMR2     ;SEE IF DSC SET
3356  013724  001001                          BNE     1$
3357  013726  104107                          ERROR   107             ;DSC NOT SET AFTER SEL DRV WITH EVEN PARITY
3358
3359  013730  012765  100000  000000  1$:     MOV     #CCLR,RKCS1(R5)
3360  013736  004737  026420                  JSR     PC,GSTAT
3361  013742  004737  025344                  JSR     PC,TSTATN
3362  013746  104113                          ERROR   113             ;NO ATTN AFTER SEL DRV WITH EVEN PARITY
3363
3364  013750  104401  037471                  TYPE    ,MSG9           ;REMOVE UNIT SELECT PLUG
3365  013754  104401  041265                  TYPE    ,MSG37          ;DEPRESS SPACE WHEN DONE
3366  013760  004737  031006                  JSR     PC,GETSP        ;GET SPACE
3367
3368  013764  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3369  013772  004737  026420                  JSR     PC,GSTAT
3370  013776  004737  025344                  JSR     PC,TSTATN       ;RETURN HERE FOR SPACE
3371  014002  000401                          BR      3$              ;NO ATTN
3372  014004  104111                          ERROR   111             ;REMOVING UNIT SEL PLUG, DID NOT
3373                                                                  ;DISABLE ATTN.
3374  014006  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
3375  014014  004737  026420                  JSR     PC,GSTAT
3376  014020  032765  010000  000010          BIT     #NED,RKCS2(R5)
3377  014026  001001                          BNE     4$
3378  014030  104112                          ERROR   112             ;REMOVING UNIT SEL PLUG DID NOT
3379                                                                  ;ASSERT NON-EXISTENT DRIVE
3380
3381  014032  104401  045030          4$:     TYPE    ,MSG67          ;DESELECT ALL OTHER PORTS
3382  014036  104401  041265                  TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
3383  014042  004737  031006                  JSR     PC,GETSP        ;GET SPACE
3384
3385  014046  104401  041331                  TYPE    ,MSG38          ;INSERT UNIT SELECT PLUGS
3386  014052  104401  041467                  TYPE    ,MSG39          ;DEPRESS CONTROL-E WHEN FINISHED
3387  014056  104401  043322                  TYPE    ,MSG55          ;EXIT WITH CORRECT UNIT SELECT PLUG #
3388  014062  013746  001222                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
3389                                                                  ;;TYPE CORRECT#
3390  014066  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3391  014070     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3392  014071     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3393  014072  104401  001205                  TYPE    ,$CRLF
3394
3395  014076  105037  033710                  CLRB    $TKQSRT         ;CLEAR PREVIOUS INFO
3396  014102  005037  001160                  CLR     $TMPO
3397
3398  014106  113737  033710  001160  5$:     MOVB    $TKQSRT,$TMPO   ;GET CHAR IF THERE
```

```
3399  014114  042737  177600  001160          BIC     #^C<177>,$TMPO  ;GET RID OF JUNK, IF ANY
3400  014122  005737  001160                  TST     $TMPO
3401  014126  001422                          BEQ     6$              ;BR IF NOTHING TYPED YET
3402  014130  023727  001160  000005          CMP     $TMPO,#5        ;SEE IF CONT-E
3403  014136  001476                          BEQ     9$              ;BR IF YES
3404  014140  023727  001160  000003          CMP     $TMPO,#3        ;SEE IF CONT-C
3405  014146  001004                          BNE     13$             ;BR IF NO
3406  014150  004737  033712                  JSR     PC,$TKINT       ;ENABLE KB INT
3407  014154  000137  031036                  JMP     STOP1           ;ELSE DO VALID HALT
3408  014160  104401  040652          13$:    TYPE    ,MSG31          ;?
3409  014164  105037  033710                  CLRB    $TKQSRT
3410  014170  004737  033712                  JSR     PC,$TKINT
3411
3412  014174  005000                  6$:     CLR     R0
3413  014176  012765  100000  000000  7$:     MOV     #CCLR,RKCS1(R5)
3414  014204  010065  000010                  MOV     R0,RKCS2(R5)    ;DRIVE NO.
3415  014210  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;SELECT DRIVE CMD
3416  014216  013737  001426  005444          MOV     T10,TEMP1
3417  014224  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
3418  014230  104117                          ERROR   117             ;CONTR RDY NOT SET
3419
3420  014232  032765  010000  000010          BIT     #NED,RKCS2(R5)  ;SEE IF UNIT SELECT PLUG INSERTED
3421  014240  001405                          BEQ     8$              ;& IF MATCH DRIVE NO IN CS2. BR IF YES
3422  014242  005200                          INC     R0
3423  014244  020027  000010                  CMP     R0,#8.          ;ALL 8 DRIVE NOS TRIED?
3424  014250  001352                          BNE     7$              ;BR IF NO
3425  014252  000715                          BR      5$              ;ELSE RETURN
3426
3427  014254  032737  000100  005434  8$:     BIT     #D.VV,HMR2
3428  014262  001311                          BNE     5$              ;TYPE SAME UNIT SELECT PLUG RDY ONCE
3429  014264  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3430  014272  010065  000010                  MOV     R0,RKCS2(R5)    ;DRIVE ADDR
3431  014276  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
3432  014304  013737  001426  005444          MOV     T10,TEMP1
3433  014312  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
3434  014316  104116                          ERROR   116             ;CONTR NOT RDY
3435  014320  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3436                                                                  ;;TYPE UNIT SEL PLUG NO.
3437  014322  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3438  014324     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3439  014325     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3440  014326  104401  001205                  TYPE    ,$CRLF
3441  014332  000665                          BR      5$
3442
3443  014334  004737  033712          9$:     JSR     PC,$TKINT       ;ENABLE KB INT
3444  014340  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3445  014346  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
3446  014354  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
3447  014362  013737  001426  005444          MOV     T10,TEMP1
3448  014370  004737  025062                  JSR     PC,FRDY
3449  014374  104117                          ERROR   117             ;CONT NOT RDY AFTER SEL DRV CMD
3450
3451  014376  032765  010000  000010          BIT     #NED,RKCS2(R5)  ;SEE IF CORRECT PLUG FOR EXIT
3452  014404  001411                          BEQ     10$             ;BR IF YES
3453  014406  104401  043322                  TYPE    ,MSG55          ;EXIT WITH CORRECT UNIT SELECT PLUG NO.
3454  014412  013746  001222                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
```

```
                                                        ;;TYPE CORRECT UNIT SEL PLUG #
014416  104403                          TYPOS           ;;GO TYPE--OCTAL ASCII
014420      001                         .BYTE   1       ;;TYPE 1 DIGIT(S)
014421      000                         .BYTE   0       ;;SUPPRESS LEADING ZEROS
014422  104401  041467                  TYPE    ,MSG39  ;DEPRESS CONT-E WHEN DONE
014426  000627                          BR      5$

014430  004737  026772        10$:    JSR     PC,SUBCLR
014434  104024                          ERROR   24      ;CERR AFTER SCLR

014436  004737  026420                  JSR     PC,GSTAT
014442  032737  000100  005434          BIT     #D.VV,HMR2
014450  001005                          BNE     11$
014452  104401  041532                  TYPE    ,MSG40  ;VV NOT SET, INSERT UNIT SELECT PLUG
014456  104401  041467                  TYPE    ,MSG39  ;DEPRESS CONT-E TO EXIT TEST
014462  000611                          BR      5$

014464  104401  045250        11$:    TYPE    ,MSG71  ;SELECT CORRECT PORT ON OTHER DRIVES
014470  104401  041265                  TYPE    ,MSG37  ;TYPE SPACE WHEN DONE
014474  004737  031006                  JSR     PC,GETSP        ;GET SPACE
014500                          12$:


;;**********************************************************************
;*TEST 12       PORT SELECTION TESTS
;*
;*      THE OPERATOR IS ASKED TO SWITCH TO THE WRONG PORT
;*      & THEN DESELECT BOTH PORTS.
;*      IN BOTH CASES, NON EXISTENT DRIVE SHOULD ASSERT IN RKCS2
;*
;;**********************************************************************
014500  000004          TST12:  SCOPE
014502  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
014510  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR

014514  004737  026772                  JSR     PC,SUBCLR
014520  104024                          ERROR   24      ;CERR AFTER SCLR

014522  104401  040050                  TYPE    ,MSG19  ;PORT SELECTION TESTS
014526  104401  043041                  TYPE    ,MSG52  ;DEPRESS CONT-E TO EXIT OR SPACE BAR TO CONTINUE
014532  004737  030746                  JSR     PC,CCSP ;INPUT CONT-E OR SPACE
014536  000137  014716                  JMP     4$      ;RETURN HERE FOR CONT-E

014542  104401  041644                  TYPE    ,MSG41  ;SWITCH TO OTHER PORT
014546  104401  041265                  TYPE    ,MSG37  ;DEPRESS SPACE WHEN DONE
014552  004737  031006                  JSR     PC,GETSP        ;GET SPACE

014556  004737  026420                  JSR     PC,GSTAT
014562  032765  010000  000010          BIT     #NED,RKCS2(R5)
014570  001001                          BNE     1$
014572  104122                          ERROR   122     ;NED NOT SET AFTER WRONG PORT SELECTED

014574  104401  041724          1$:     TYPE    ,MSG42  ;SELECT CORRECT PORT
014600  104401  041265                  TYPE    ,MSG37  ;DEPRESS SPACE BAR WHEN DONE
014604  004737  031006                  JSR     PC,GETSP        ;GET SPACE
```

```
                                                
014610  004737  026772                JSR     PC,SUBCLR
014614  104024                          ERROR   24              ;CERR AFTER SCLR
                                                
014616  032765  010000  000010        BIT     #NED,RKCS2(R5)
014624  001401                          BEQ     2$
014626  104130                          ERROR   130             ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
                                                
014630  104401  042002        2$:     TYPE    ,MSG43          ;DESELECT BOTH PORTS
014634  104401  041265                TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
014640  004737  031006                JSR     PC,GETSP
014644  004737  026420                JSR     PC,GSTAT
014650  032765  010000  000010        BIT     #NED,RKCS2(R5)
014656  001001                          BNE     3$
014660  104133                          ERROR   133             ;NED NOT SET AFTER BOTH PORTS DESELECTED
                                                
014662  104401  042032        3$:     TYPE    ,MSG44          ;SELECT CORRECT PORT
014666  104401  041265                TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
014672  004737  031006                JSR     PC,GETSP
                                                
014676  004737  026772                JSR     PC,SUBCLR
014702  104024                          ERROR   24              ;CERR AFTER SCLR
                                                
014704  032765  010000  000010        BIT     #NED,RKCS2(R5)
014712  001421                          BEQ     TST13           ;;GOTO NEXT TEST
014714  104130                          ERROR   130             ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
                                                
014716  012765  100000  000000  4$:     MOV     #CCLR,RKCS1(R5)
014724  004737  026420                JSR     PC,GSTAT
014730  032765  010000  000010        BIT     #NED,RKCS2(R5)
014736  001407                          BEQ     TST13           ;;GOTO NEXT TEST
014740  104401  042062                TYPE    ,MSG45          ;CORRECT PORT NOT SELECTED, TRY AGAIN
014744  104401  041265                TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
014750  004737  031006                JSR     PC,GETSP
014754  000760                          BR      4$
                                                
                                        ;;**************************************************************
                                        ;*TEST 13       FRONT PANEL RUN/STOP SWITCH TEST
                                        ;*
                                        ;*      THIS TEST ALLOWS THE HEADS TO LOAD. THE OPERATOR IS
                                        ;*      ASKED TO VISUALLY VERIFY THE SEQUENCE OF HEADS LOADING &
                                        ;*      UNLOADING BOTH MECHANICALLY & BY THE LIGHTS ON THE FRONT PANEL
                                        ;*
                                        ;;**************************************************************
014756  000004                TST13:  SCOPE
014760  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
014766  012706  001100                MOV     #STACK,SP       ;RESET STACK POINTER.
                                                
014772  004737  026772                JSR     PC,SUBCLR
014776  104024                          ERROR   24              ;CERR AFTER SCLR
                                                
015000  104401  040101                TYPE    ,MSG20          ;RUN/STOP SWITCH TEST
015004  104401  043041                TYPE    ,MSG52          ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
015010  004737  030746                JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
015014  000137  015262                JMP     6$              ;RETURN HERE IF CONT-E
```

```
3567
3568  015020  104401  042133          TYPE    ,MSG46          ;DO MANUAL DRIVE LOAD
3569  015024  104401  045145          TYPE    ,MSG69          ;DEPRESS SPACE WHEN DRIVE 'READY' ON
3570  015030  004737  031006          JSR     PC,GETSP        ;GET SPACE
3571
3572  015034  004737  026420          JSR     PC,GSTAT
3573  015040  032737  010000  005434  BIT     #D.SPIN,HMR2
3574  015046  001001                  BNE     1$
3575  015050  104140                  ERROR   140             ;SPINDLE NOT ON AFTER DRIVE MANUALLY LOADED
3576
3577  015052  032737  000200  005434  1$:     BIT     #D.DRDY,HMR2    ;SEE IF DRIVE READY (LOADED)
3578  015060  001001                  BNE     3$              ;BR IF YES
3579  015062  104141                  ERROR   141             ;DRIVE NOT READY AFTER MANUAL LOADING
3580
3581  015064  005037  005370          3$:     CLR     UNLD            ;FOR VALID HALT
3582  015070  104401  042336          TYPE    ,MSG47          ;ATTEMPT TO OPEN DOOR
3583  015074  104401  041265          TYPE    ,MSG37          ;DEPRESS SPACE WHEN FINISHED
3584  015100  004737  031006          JSR     PC,GETSP        ;GET SPACE
3585
3586  015104  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
3587  015112  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3588  015120  004737  026420          JSR     PC,GSTAT
3589  015124  032737  000200  005434  BIT     #D.DOOR,HMR2
3590  015132  001001                  BNE     4$
3591  015134  104142                  ERROR   142             ;DOOR STATUS BIT NOT SET
3592
3593  015136  104401  042416          4$:     TYPE    ,MSG48          ;UNLOAD HEADS MANUALLY
3594  015142  104401  041265          TYPE    ,MSG37          ;SPACE BAR WHEN DONE
3595  015146  004737  031006          JSR     PC,GETSP        ;GET SPACE
3596
3597  015152  005237  005370          INC     UNLD            ;FOR VALID HALT
3598  015156  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
3599  015164  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3600  015172  004737  026420          JSR     PC,GSTAT
3601  015176  032737  000040  005434  BIT     #D.HDHM,HMR2    ;CHECK HEAD HOME
3602  015204  001001                  BNE     5$
3603  015206  104145                  ERROR   145             ;HEADS HOME NOT SET AFTER MANUAL UNLD
3604
3605  015210  104401  042133          5$:     TYPE    ,MSG46          ;DO MANUAL LOAD
3606  015214  104401  045145          TYPE    ,MSG69          ;PRESS SPACE AFTER READY ON
3607  015220  004737  031006          JSR     PC,GETSP
3608
3609  015224  004737  026772          JSR     PC,SUBCLR
3610  015230  104024                  ERROR   24              ;CERR AFTER SCLR
3611
3612  015232  012765  000007  000000  MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
3613  015240  013737  001426  005444  MOV     T10,TEMP1
3614  015246  004737  025062          JSR     PC,FRDY         ;FIND CONTR RDY
3615  015252  104017                  ERROR   17              ;NO RDY AFTER UNLD CMD
3616  015254  004737  025344          JSR     PC,TSTATN
3617  015260  104020                  ERROR   20              ;NO ATTN AFTER UNLOAD CMD
3618  015262                  6$:
3619
3620                          ;;*****************************************************************
3621                          ;*TEST 14       AC LOW DETECTION  PART 1
3622                          ;*
```

```
                                        ;*      A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.
                                        ;*      BATTERY RETRACT WILL BE TESTED LATER.
                                        ;*      THE PROGRAM WAITS FOR AC LOW TO ASSERT IN RKMR3.
                                        ;*      FROM THIS POINT, THERE IS APPROX 4 MS AVAILABLE BEFORE DC LOW ASSERTS
                                        ;*      & THE INTERFACE SHUTS DOWN.
                                        ;*      THE INDICATION OF DC LOW WILL BE NON-EXISTENT DRIVE ASSERTING IN RKCS2.
                                        ;*      AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.
                                        ;*
                                        ;**********************************************************************
015262  000004                  TST14:  SCOPE
015264  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
015272  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

015276  004737  026772                  JSR     PC,SUBCLR
015302  104024                          ERROR   24              ;CERR AFTER SCLR

015304  104401  040132                  TYPE    ,MSG21          ;AC LOW TEST
015310  104401  043041                  TYPE    ,MSG52          ;CONT-E TO BYPASS TEST
                                                                ;OR SPACE TO CONTINUE
015314  004737  030746                  JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
015320  000137  015666                  JMP     10$             ;RETURN HERE IF CONT-E
015324  104401  042613                  TYPE    ,MSG49          ;TURN OFF AC(RET HERE IF SPACE)

015330  005037  005446                  CLR     TEMP2
015334  012737  177777  005444  12$:    MOV     #-1,TEMP1       ;SETUP TIMEOUT
015342  004737  026420          1$:     JSR     PC,GSTAT
015346  032737  100000  005406          BIT     #CERR,HCS1
015354  001012                          BNE     9$
015356  005337  005444                  DEC     TEMP1
015362  001367                          BNE     1$
015364  005237  005446                  INC     TEMP2
015370  023727  005446  000002          CMP     TEMP2,#2        ;SEE IF GONE THRU 2 TIMES
015376  001356                          BNE     12$             ;BR IF NO
015400  104146                          ERROR   146             ;CERR NOT DETECTED BEFORE TIMEOUT

015402  000137  015466          9$:     JMP     3$              ;***THIS IS A TEMP JUMP***
                                        ;ACLO REQUIRES SECTOR PULSE AND SINCE HEADS ARE UNLOADED
                                        ;ACLO WILL NOT ASSERT & CERR WILL BE DUE TO NED
                                        ;THE PROBLEM IS BEING LOOKED INTO
                                        ;THIS TEST WILL EITHER BE ELIMINATED OR AC LO WILL BE MODIFIED TO ASSERT

015406  032737  000100  005436          BIT     #D.ACLO,HMR3
015414  001001                          BNE     2$
015416  104147                          ERROR   147             ;AC LOW NOT SET

015420  012737  040100  005476  2$:     MOV     #<D.DSC!D.VV>,E.A0  ;EXPECTED MSG A0
015426  012737  000300  005500          MOV     #<D.ACLO!D.FLT>,E.B0
015434  012737  000740  005476          MOV     #<D.CART!D.DOOR!D.BRHM!D.HDHM>,E.A0
015442  012737  000001  005504          MOV     #1,E.B1

015450  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
015454  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
015456  104035                          ERROR   35              ;MSG A0 ERROR AFTER AC SWITCH OFF FROM HEADS HOME
015460  104203                          ERROR   203             ;MSH B0 ERROR
015462  104036                          ERROR   36              ;MSG A1 ERROR
```

```
3679  015464  104037                          ERROR   37              ;MSG B1 ERROR
3680
3681  015466  013737  001440  005444  3$:     MOV     T50000,TEMP1
3682  015474  012765  100000  000000  4$:     MOV     #CCLR,RKCS1(R5)
3683  015502  004737  026420                  JSR     PC,GSTAT
3684  015506  032765  010000  000010          BIT     #NED,RKCS2(R5)
3685  015514  001004                          BNE     5$
3686  015516  005337  005444                  DEC     TEMP1
3687  015522  001364                          BNE     4$
3688  015524  104150                          ERROR   150             ;NED NOT SET BEFORE TIMEOUT
3689
3690  015526  104401  042666          5$:     TYPE    ,MSG50          ;SWITCH AC BACK ON
3691  015532  104401  045222                  TYPE    ,MSG70          ;VERIFY HEADS LOAD
3692  015536  104401  045145                  TYPE    ,MSG69          ;PRESS SPACE AFTER READY ON
3693  015542  004737  031006                  JSR     PC,GETSP        ;GET SPACE
3694
3695  015546  004737  026772                  JSR     PC,SUBCLR
3696  015552  104024                          ERROR   24              ;CERR AFTER SCLR
3697
3698  015554  032737  000200  005434          BIT     #D.DRDY,HMR2    ;SEE IF LOADED
3699  015562  001001                          BNE     6$              ;BR IN YES
3700  015564  104141                          ERROR   141             ;DRV NOT RDY AFTER AC UP
3701
3702  015566  005037  005370          6$:     CLR     UNLD            ;FOR VALID HALT
3703  015572  032737  000100  005436          BIT     #D.ACLO,HMR3
3704  015600  001401                          BEQ     7$
3705  015602  104152                          ERROR   152             ;ACLO NOT RESET AFTER POWER UP
3706
3707  015604  032737  000100  005434  7$:     BIT     #D.VV,HMR2
3708  015612  001401                          BEQ     8$
3709  015614  104153                          ERROR   153             ;VV NOT RESET AFTER POWER UP
3710
3711  015616                          8$:
3712  015616  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3713  015624  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE #
3714  015632  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
3715  015640  013737  001426  005444          MOV     T10,TEMP1
3716  015646  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
3717  015652  104116                          ERROR   116             ;CONTR NOT RDY
3718
3719  015654  032737  000100  005434          BIT     #D.VV,HMR2
3720  015662  001001                          BNE     64$
3721  015664  104027                          ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
3722  015666                          64$:
3723  015666                          10$:
3724
3725                                  ;;*****************************************************************
3726                                  ;*TEST 15       CHECK NXF LOGIC
3727                                  ;*
3728                                  ;*      THIS TEST VERIFIES NON-EXECUTABLE FUNCTION (NXF) IS DETECTED
3729                                  ;*      AS A RESULT OF DOING A SEEK WITH VOLUME VALID RESET.
3730                                  ;*
3731                                  ;;*****************************************************************
3732  015666  000004                  TST15:  SCOPE
3733  015670  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3734  015676  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR
```

```
3735
3736  015702  004737  026772              JSR     PC,SUBCLR
3737  015706  104024                      ERROR   24                  ;CERR AFTER SCLR
3738
3739  015710  032737  010000  005434      BIT     #D.SPIN,HMR2        ;SEE IF SPINDLE ALREADY ON
3740  015716  001021                      BNE     64$                 ;BR IF YES
3741  015720  104401  040524              TYPE    ,MSG29              ;PLEASE WAIT, HEADS BEING LOADED
3742
3743  015724  012765  000011  000000      MOV     #SRTSPL,RKCS1(R5)   ;START SPINDLE CMD
3744  015732  013737  001426  005444      MOV     T10,TEMP1
3745  015740  004737  025062              JSR     PC,FRDY             ;FIND CONTR RDY
3746  015744  104143                      ERROR   143                 ;CONTR RDY NOT SET AFTER CMD
3747
3748  015746  013737  001434  005446      MOV     T100,TEMP2
3749  015754  004737  025376              JSR     PC,FATT1            ;FIND ATTN
3750  015760  104144                      ERROR   144                 ;NO ATTN AFTER CMD
3751  015762                      64$:
3752  015762  005037  005370              CLR     UNLD                ;FOR VALID HALT
3753
3754  015766  004737  026772              JSR     PC,SUBCLR
3755  015772  104024                      ERROR   24                  ;CERR AFTER SCLR
3756
3757  015774  104401  040173              TYPE    ,MSG22              ;NXF TEST
3758  016000  104401  043041              TYPE    ,MSG52              ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
3759  016004  004737  030746              JSR     PC,CCSP             ;INPUT CONT-E OR SPACE
3760  016010  000137  016332              JMP     7$                  ;RETURN HERE FOR CONT-E
3761
3762  016014  104401  042720              TYPE    ,MSG51              ;REMOVE UNIT SELECT PLUG TO CLEAR VV
3763  016020  104401  041265              TYPE    ,MSG37              ;PRESS SPACE WHEN DONE
3764  016024  004737  031006              JSR     PC,GETSP            ;GET SPACE
3765
3766  016030  004737  026420              JSR     PC,GSTAT
3767  016034  032737  000100  005434      BIT     #D.VV,HMR2
3768  016042  001403                      BEQ     1$
3769  016044  104177                      ERROR   177                 ;VV NOT CLEARED AFTER UNIT SEL PLUG
3770  016046  000137  016332              JMP     7$                  ;EXIT TEST
3771
3772  016052  032737  000100  005434  1$: BIT     #D.VV,HMR2
3773  016060  001406                      BEQ     2$
3774  016062  104155                      ERROR   155                 ;VV SET AFTER HEADS LOADED
3775  016064  000137  016332              JMP     7$                  ;EXIT TEST
3776
3777  016070  004737  026772              JSR     PC,SUBCLR
3778  016074  104024                      ERROR   24                  ;CERR AFTER SCLR
3779
3780  016076  012765  000001  000020  2$: MOV     #1,RKDC(R5)
3781  016104  012765  000017  000000      MOV     #SEEK,RKCS1(R5)     ;SEEK CMD
3782  016112  013737  001426  005444      MOV     T10,TEMP1
3783  016120  004737  025062              JSR     PC,FRDY             ;FIND RDY
3784  016124  104131                      ERROR   131                 ;NO RDY AFTER SEEK CMD
3785
3786  016126  013737  001440  005444      MOV     T50000,TEMP1
3787  016134  004737  025472              JSR     PC,FATT2            ;FIND ATTN
3788  016140  104132                      ERROR   132                 ;NO ATTN AFTER SEEK CMD
3789
3790  016142  032737  000400  005436      BIT     #D.NXF,HMR3
```

```
3791  016150  001003                          BNE     3$
3792  016152  104156                          ERROR   156             ;NXF NOT SET AFTER SEEK WITH VV=0
3793  016154  000137  016332                  JMP     7$              ;EXIT TEST
3794
3795  016160  032737  000200  005436  3$:     BIT     #D.FLT,HMR3
3796  016166  001003                          BNE     4$
3797  016170  104172                          ERROR   172             ;NXF DID NOT SET FAULT
3798  016172  000137  016332                  JMP     7$              ;EXIT TEST
3799
3800  016176  012737  050240  005476  4$:     MOV     #<D.DSC!D.SPIN!D.DRDY!D.DRA>,E.A0  ;EXPECTED A0
3801  016204  012737  000600  005500          MOV     #<D.NXF!D.FLT>,E.B0
3802  016212  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
3803  016220  012737  000001  005504          MOV     #1,E.B1
3804
3805  016226  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
3806  016232  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
3807  016234  104160                          ERROR   160             ;MSG A0 ERROR AFTER SEEK WITH VV=0
3808  016236  104161                          ERROR   161             ;MSH B0 ERROR
3809  016240  104162                          ERROR   162             ;MSG A1 ERROR
3810  016242  104163                          ERROR   163             ;MSG B1 ERROR
3811  016244  005737  001400                  TST     CYLADD
3812  016250  001401                          BEQ     5$
3813  016252  104157                          ERROR   157             ;HEADS MOVED WITH SEEK & DXF
3814
3815  016254  004737  026772          5$:     JSR     PC,SUBCLR
3816  016260  104024                          ERROR   24              ;CERR AFTER SCLR
3817
3818  016262  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3819  016270  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE #
3820  016276  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
3821  016304  013737  001426  005444          MOV     T10,TEMP1
3822  016312  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
3823  016316  104116                          ERROR   116             ;CONTR NOT RDY
3824
3825  016320  032737  000100  005434          BIT     #D.VV,HMR2
3826  016326  001001                          BNE     65$
3827  016330  104027                          ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
3828  016332                          65$:
3829  016332                          7$:
3830
3831                                  ;;*****************************************************************
3832                                  ;*TEST 16       READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
3833                                  ;*
3834                                  ;*      THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
3835                                  ;*      THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
3836                                  ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFOR (BSE) WILL BE STORED
3837                                  ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
3838                                  ;*
3839                                  ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
3840                                  ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
3841                                  ;*
3842                                  ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
3843                                  ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
3844                                  ;*      A MESSAGE WILL BE TYPED INDICATING THAT ALL
3845                                  ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
3846                                  ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFOR OR AN ALIGNMENT PACK BY WRITING
```

```
3847                                    ;*
3848                                    ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
3849                                    ;*
3850                                    ;;*****************************************************************
3851  016332  000004                    TST16:  SCOPE
3852  016334  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
3853  016342  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
3854
3855  016346  004737  026772                    JSR     PC,SUBCLR
3856  016352  104024                            ERROR   24              ;CERR AFTER SCLR
3857  016354  005037  005446                    CLR     TEMP2           ;SECTOR CTR
3858  016360  005037  005450                    CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
3859                                                                    ;1=22 SECTOR SOFTWARE DETECTED TABLE
3860                                                                    ;2=DONE
3861  016364  012737  002370  005452            MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE ADDR.
3862  016372  013765  005452  000004            MOV     TEMP4,RKBA(R5)
3863  016400  012737  001000  005454            MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
3864  016406  013765  005454  000006            MOV     TEMP5,RKDA(R5)
3865
3866  016414  012765  000632  000020  1$:       MOV     #410.,RKDC(R5)  ;CYL 410
3867  016422  012765  177400  000002            MOV     #-256.,RKWC(R5) ;LOAD WORD CT
3868  016430  012765  000021  000000            MOV     #RDDATA,RKCS1(R5);READ DATA COMMAND
3869  016436  013737  001440  005444            MOV     T50000,TEMP1    ;SETUP TIMEOUT
3870  016444  004737  025062                    JSR     PC,FRDY         ;FIND RDY
3871  016450  104226                            ERROR   226             ;NO RDY AFTER READ DATA CMD
3872  016452  004737  026420                    JSR     PC,GSTAT        ;GET FRESH DATA
3873  016456  032737  100000  005406            BIT     #CERR,HCS1
3874  016464  001470                            BEQ     8$
3875  016466  104227                            ERROR   227             ;CERR AFTER READ DATA CMD
3876
3877  016470  012737  010340  005476            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
3878  016476  005037  005500                    CLR     E.B0            ;EXPECTED MSG B0
3879  016502  012737  001720  005502            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
3880  016510  012737  000001  005504            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
3881  016516  005037  005506                    CLR     E.A2            ;EXPECTED MSG A2
3882  016522  012737  000002  005510            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
3883  016530  012737  000003  005514            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
3884
3885  016536  004737  025604                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
3886  016542  000000                            .WORD   0!0!0           ;& MSGS SPECIFIED HERE
3887  016544  104054                            ERROR   54              ;MSG A0 ERROR AFTER READ DATA CMD
3888  016546  104026                            ERROR   26              ;MSH B0 ERROR
3889  016550  104056                            ERROR   56              ;MSG A1 ERROR
3890  016552  104030                            ERROR   30              ;MSG B1 ERROR
3891  016554  004737  026772                    JSR     PC,SUBCLR
3892  016560  104024                            ERROR   24              ;CERR AFTER SUBCLR
3893  016562  005237  005446                    INC     TEMP2
3894  016566  023727  005446  000005            CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
3895  016574  001007                            BNE     5$
3896  016576  005737  005450                    TST     TEMP3
3897  016602  001002                            BNE     2$
3898  016604  104165                            ERROR   165             ;CANT READ SECTORS 0,2,4,6,8
3899  016606  000414                            BR      3$
3900  016610  104167                    2$:     ERROR   167             ;CANT READ SECTORS 10,12,14,16,18,20
3901  016612  000412                            BR      3$
3902
```

```
3903  016614  013765  005452  000004  5$:     MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
3904  016622  062737  000002  005454          ADD     #2,TEMP5        ;SETUP TO READ 2 SECTORS FROM LAST
3905  016630  013765  005454  000006          MOV     TEMP5,RKDA(R5)
3906  016636  000666                          BR      1$
3907
3908  016640  005237  001542          3$:     INC     BSERR           ;SET BSE FLAG
3909  016644  000504                          BR      TST17           ;;GO TO NEXT TEST
3910  016646  005737  002376          8$:     TST     BSE22H+6        ;TEST CARTRIDGE TYPE
3911  016652  001405                          BEQ     9$              ;BRANCH IF DATA CARTRIDGE
3912  016654  104401  043402                  TYPE    ,MSG56          ;ALIGNMENT CARTRIDGE USED
3913  016660  005237  001542                  INC     BSERR           ;SET BSE ERROR FLAG
3914  016664  000426                          BR      10$
3915
3916  016666  005237  005450          9$:     INC     TEMP3
3917  016672  023727  005450  000001          CMP     TEMP3,#1
3918  016700  001020                          BNE     10$
3919  016702  005037  005446                  CLR     TEMP2
3920  016706  012737  003370  005452          MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
3921  016714  013765  005452  000004          MOV     TEMP4,RKBA(R5)
3922  016722  012737  001012  005454          MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12
3923  016730  013765  005454  000006          MOV     TEMP5,RKDA(R5)
3924  016736  000137  016414                  JMP     1$              ;REPEAT
3925
3926  016742  005737  001216          10$:    TST     $PASS
3927
3928
3929  016746  001014                          BNE     13$             ;TYPE CART # ONLY ON 1'ST PASS
3930  016750  104401  037766                  TYPE    ,MSG17          ;CART SERIAL #
3931  016754  012746  002370                  MOV     #BSE22H,-(SP)
3932  016760  004737  035312                  JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
3933  016764  004737  035662                  JSR     PC,$SUPRS       ;TYPE SERIAL #
3934  016770  104401  001205                  TYPE    ,$CRLF
3935  016774  104401  001205                  TYPE    ,$CRLF
3936
3937  017000  004737  026772          13$:    JSR     PC,SUBCLR
3938  017004  104024                          ERROR   24              ;CERR AFTER SCLR
3939                                                                  ;RETURN TO CYL 0
3940
3941  017006  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
3942  017014  013737  001430  005444          MOV     T50,TEMP1       ;SETUP TIMEOUT
3943  017022  004737  025062                  JSR     PC,FRDY         ;FIND RDY
3944  017026  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
3945
3946  017030  013737  001440  005444          MOV     T50000,TEMP1    ;SETUP TIMEOUT
3947  017036  004737  025472                  JSR     PC,FATT2        ;FIND ATTN
3948  017042  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
3949
3950  017044  032737  100000  005406          BIT     #CERR,HCS1
3951  017052  001401                          BEQ     64$
3952  017054  104210                          ERROR   210             ;CERR AFTER SEEK CMD
3953
3954  017056                          64$:
3955
3956
3957                                  ;;*****************************************************************
3958                                  ;*TEST 17        WRITE LOCK TEST
```

```
3959                                    ;*
3960                                    ;*      THIS TEST VERIFIES THAT DATA WRITTEN ON A SECTOR CANNOT BE
3961                                    ;*      ALTERED ONCE THE WRITE LOCK SWITCH HAS BEEN ENABLED.
3962                                    ;*      IT ALSO CHECKS THAT WRITE PROTECT TAKES EFFECT ONLY AT
3963                                    ;*      SECTOR BOUNDRIES WHEN DOING CONTINUOUS WRITING ON CYL 0, HEAD 0
3964                                    ;*
3965                                    ;;*****************************************************************
3966  017056  000004                    TST17:  SCOPE
3967  017060  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
3968  017066  012706  001100                    MOV     #STACK,SP       ;RESTORE STACK PTR
3969  017072  005737  001542                    TST     BSERR           ;SEE IF ALIGN CART
3970  017076  001402                            BEQ     15$             ;BR IF NO
3971  017100  000137  022016                    JMP     12$             ;ELSE EXIT TEST
3972
3973  017104  004737  026772            15$:    JSR     PC,SUBCLR
3974  017110  104024                            ERROR   24              ;CERR AFTER SCLR
3975
3976  017112  104401  040341                    TYPE    ,MSG24          ;WRITE LOCK TEST
3977  017116  104401  043041                    TYPE    ,MSG52          ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
3978  017122  004737  030746                    JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
3979  017126  000137  022016                    JMP     12$             ;RETURN HERE IF CONT-E
3980
3981  017132  004737  026420                    JSR     PC,GSTAT
3982  017136  032737  004000  005434            BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK IS ON
3983  017144  001416                            BEQ     2$              ;BR IF NO
3984  017146  104401  043142            1$:     TYPE    ,MSG53          ;DISABLE WRITE LOCK
3985  017152  104401  041265                    TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
3986  017156  004737  031006                    JSR     PC,GETSP        ;GET SPACE
3987
3988  017162  004737  026420                    JSR     PC,GSTAT
3989  017166  032737  004000  005434            BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK IS OFF
3990  017174  001402                            BEQ     2$
3991  017176  104110                            ERROR   110             ;WRITE LOCK NOT DISABLED
3992  017200  000762                            BR      1$
3993
3994  017202  005037  001366            2$:     CLR     TOCYL           ;SETUP
3995  017206  005037  001402                    CLR     CALADD          ;FOR
3996  017212  005037  001510                    CLR     HEAD            ;FILL HEADER TABLE
3997  017216  005037  001516                    CLR     FORMAT          ;ROUTINE
3998
3999  017222  004737  027754            16$:    JSR     PC,FHDTAB       ;BUILD STD 22 SECTOR HEADER TABLE
4000
4001  017226  012765  001554  000004            MOV     #HDTAB,RKBA(R5)
4002  017234  012765  177676  000002            MOV     #-66.,RKWC(R5)
4003  017242  000337  001510                    SWAB    HEAD
4004  017246  013765  001510  000006            MOV     HEAD,RKDA(R5)   ;HEAD ADDR
4005  017254  000337  001510                    SWAB    HEAD
4006
4007
4008  017260  012765  000027  000000            MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
4009  017266  013737  001440  005444            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4010  017274  004737  025062                    JSR     PC,FRDY         ;FIND RDY
4011  017300  104200                            ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4012  017302  004737  026420                    JSR     PC,GSTAT        ;GET FRESH STATUS
4013  017306  032737  100000  005406            BIT     #CERR,HCS1
4014  017314  001401                            BEQ     64$
```

```
4015  017316  104201                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
4016  017320                        64$:
4017
4018
4019  017320  005237  001510                INC     HEAD
4020  017324  023727  001510  000003        CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
4021  017332  001333                        BNE     16$             ;BR IF NO
4022
4023  017334  004737  026772                JSR     PC,SUBCLR
4024  017340  104024                        ERROR   24              ;CERR AFTER SCLR
4025
4026  017342  005037  001422                CLR     SECTOR
4027  017346  013765  001422  000006  14$:  MOV     SECTOR,RKDA(R5)
4028
4029  017354  012765  001524  000004        MOV     #DATA0,RKBA(R5) ;WRITE ALL 0'S
4030  017362  052765  000020  000010        BIS     #BAI,RKCS2(R5)
4031  017370  012765  177400  000002        MOV     #-256.,RKWC(R5) ;CYL 0, TRK 0, SECTOR 0
4032
4033  017376  012765  000023  000000        MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
4034  017404  012737  001440  005444        MOV     T50000,TEMP1    ;SETUP TIMEOUT
4035  017412  004737  025062                JSR     PC,FRDY         ;FIND RDY
4036  017416  104011                        ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4037  017420  004737  026420                JSR     PC,GSTAT        ;GET FRESH STATUS
4038  017424  032737  100000  005406        BIT     #CERR,HCS1
4039  017432  001465                        BEQ     68$             ;BR IF NO ERRORS
4040
4041  017434  032737  000200  005422        BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
4042  017442  001421                        BEQ     66$             ;BR IF NO
4043  017444  004737  030412                JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4044  017450  000455                        BR      67$             ;RETURN HERE IF NO
4045
4046  017452  005237  001422                INC     SECTOR          ;RETURN HERE IF YES
4047  017456  023727  001422  000012        CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4048  017464  001003                        BNE     65$             ;BR IF NO
4049  017466  104040                        ERROR   40              ;ABORTING TEST DETECTED 10 BAD SECTORS
4050  017470  000137  022016                JMP     12$             ;BYPASS TEST
4051
4052  017474  012765  100000  000000  65$:  MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4053  017502  000137  017346                JMP     14$
4054
4055  017506  104012                66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
4056
4057  017510  012737  010340  005476        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4058  017516  005037  005500                CLR     E.B0            ;EXPECTED MSG B0
4059  017522  012737  001720  005502        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4060  017530  012737  000001  005504        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4061  017536  005037  005506                CLR     E.A2            ;EXPECTED MSG A2
4062  017542  012737  000002  005510        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4063  017550  012737  000003  005514        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4064
4065  017556  004737  025604                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4066  017562  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4067  017564  104052                        ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4068  017566  104023                        ERROR   23              ;MSH B0 ERROR
4069  017570  104053                        ERROR   53              ;MSG A1 ERROR
4070  017572  104025                        ERROR   25              ;MSG B1 ERROR
```

```
4071  017574  104401  045333                  TYPE    ,MSG72          ;ABORTING BALANCE OF TESTS
4072  017600  000137  024402                  JMP     $EOP
4073  017604  104047                  67$:    ERROR   47              ;BAD SECTOR NOT LISTED IN TABLE
4074  017606                          68$:
4075
4076  017606  004737  026772                  JSR     PC,SUBCLR
4077  017612  104024                          ERROR   24              ;CERR AFTER SCLR
4078
4079  017614  012765  001524  000004          MOV     #DATAD,RKBA(R5)
4080  017622  052765  000020  000010          BIS     #BAI,RKCS2(R5)
4081  017630  012765  177400  000002          MOV     #-256.,RKWC(R5)
4082  017636  013737  001524  001504          MOV     DATAD,WD2       ;EXPECTED WORD FOR TRUERR TYPEOUT
4083  017644  013765  001422  000006          MOV     SECTOR,RKDA(R5)
4084
4085  017652  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4086  017660  013737  001440  005444          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4087  017666  004737  025062                  JSR     PC,FRDY         ;FIND RDY
4088  017672  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4089  017674  004737  026420                  JSR     PC,GSTAT        ;GET FRESH STATUS
4090  017700  032737  100000  005406          BIT     #CERR,HCS1
4091  017706  001450                          BEQ     70$
4092  017710  032737  040000  005410          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
4093  017716  001405                          BEQ     69$
4094  017720  016537  000024  001502          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
4095  017726  104016                          ERROR   16              ;WCE AFTER WRITE CMD
4096  017730  000437                          BR      70$
4097
4098  017732  104022                  69$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4099
4100  017734  012737  010340  005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4101  017742  005037  005500                  CLR     E.B0            ;EXPECTED MSG B0
4102  017746  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4103  017754  012737  000001  005504          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4104  017762  005037  005506                  CLR     E.A2            ;EXPECTED MSG A2
4105  017766  012737  000002  005510          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4106  017774  012737  000003  005514          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4107
4108  020002  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4109  020006  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4110  020010  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
4111  020012  104031                          ERROR   31              ;MSH B0 ERROR
4112  020014  104060                          ERROR   60              ;MSG A1 ERROR
4113  020016  104032                          ERROR   32              ;MSG B1 ERROR
4114  020020  104401  045333                  TYPE    ,MSG72          ;ABORTING BALANCE OF TESTS
4115  020024  000137  024006                  JMP     ENDRV
4116
4117  020030                          70$:
4118
4119  020030  004737  026772          3$:     JSR     PC,SUBCLR
4120  020034  104024                          ERROR   24              ;CERR AFTER SCLR
4121
4122  020036  104401  043233                  TYPE    ,MSG54          ;ENABLE WRITE LOCK
4123  020042  104401  041265                  TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4124  020046  004737  031006                  JSR     PC,GETSP        ;GET SPACE
4125
4126  020052  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
4127  020060  004737  026420                    JSR     PC,GSTAT
4128  020064  032737  004000  005434            BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK IS ON
4129  020072  001002                            BNE     4$
4130  020074  104115                            ERROR   115             ;WRITE LOCK NOT ENABLED
4131  020076  000754                            BR      3$
4132
4133  020100  012765  001530  000004  4$:       MOV     #DATA1,RKBA(R5) ;ATTEMPT TO WRITE ALL 1'S WITH WRITE LOCK SET
4134  020106  052765  000020  000010            BIS     #BAI,RKCS2(R5)
4135  020114  012765  177400  000002            MOV     #-256.,RKWC(R5)  ; CYLINDER 0, HEAD 0
4136  020122  013765  001422  000006            MOV     SECTOR,RKDA(R5)
4137  020130  012765  000023  000000            MOV     #WRDATA,RKCS1(R5)
4138  020136  013737  001440  005444            MOV     T50000,TEMP1    ;SETUP TIMEOUT
4139  020144  004737  025062                    JSR     PC,FRDY         ;FIND RDY
4140  020150  104116                            ERROR   116             ;CONTR NOT RDY
4141
4142  020152  004737  026420                    JSR     PC,GSTAT        ;GET FRESH STATUS
4143  020156  032737  100000  005406            BIT     #CERR,HCS1
4144  020164  001001                            BNE     17$
4145  020166  104207                            ERROR   207             ;CERR NOT SET AFTER WRITE WITH WRL SET
4146  020170  032737  004000  005436  17$:      BIT     #D.WLE,HMR3     ;CHECK WRITE LOCK ERROR SET
4147  020176  001001                            BNE     5$              ;BR IF SET
4148  020200  104121                            ERROR   121             ;WRITE LOCK ERROR NOT SET
4149                                            ;AFTER WRITE DATA WITH WRITE LOCK SET
4150  020202  012737  054340  005476  5$:       MOV     #<D.DSC!D.SPIN!D.WRL!D.DRDY!D.VV!D.DRA>,E.A0  ;EXP A0
4151  020210  012737  004200  005500            MOV     #<D.WLE!D.FLT>,E.B0
4152  020216  012737  001720  005502            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4153  020224  012737  000001  005504            MOV     #1,E.B1
4154
4155  020232  004737  025604                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4156  020236  000000                            .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4157  020240  104123                            ERROR   123             ;MSG A0 ERROR AFTER WRITE WITH WRITE LOCK SET
4158  020242  104125                            ERROR   125             ;MSH B0 ERROR
4159  020244  104126                            ERROR   126             ;MSG A1 ERROR
4160  020246  104127                            ERROR   127             ;MSG B1 ERROR
4161
4162  020250  004737  026772                    JSR     PC,SUBCLR
4163  020254  104024                            ERROR   24              ;CERR AFTER SCLR
4164
4165  020256  012765  001524  000004            MOV     #DATA0,RKBA(R5) ;CHECK THAT NONE OF ORIG DATA
4166  020264  052765  000020  000010            BIS     #BAI,RKCS2(R5)  ;HAS CHANGED
4167  020272  012765  177400  000002            MOV     #-256.,RKWC(R5)  ;AS RESULT OF WRITE WITH WRITE LOCK SET
4168  020300  013737  001524  001504            MOV     DATA0,WD2       ;EXPECTED WORD FOR TRUERR TYPEOUT
```

```
4169  020306  013765  001422  000006          MOV     SECTOR,RKDA(R5)
4170
4171  020314  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4172  020322  013737  001440  005444          MOV     T50000,TEMP1            ;SETUP TIMEOUT
4173  020330  004737  025062                  JSR     PC,FRDY                 ;FIND RDY
4174  020334  104015                          ERROR   15                      ;NO RDY AFTER WRITE CHECK CMD
4175  020336  004737  026420                  JSR     PC,GSTAT                ;GET FRESH STATUS
4176  020342  032737  100000  005406          BIT     #CERR,HCS1
4177  020350  001450                          BEQ     72$
4178  020352  032737  040000  005410          BIT     #WCE,HCS2               ;SEE IF WRITE CHECK ERROR
4179  020360  001405                          BEQ     71$
4180  020362  016537  000024  001502          MOV     RKDB(R5),WD1            ;ACTUAL WORD FOR PRINTOUT
4181  020370  104016                          ERROR   16                      ;WCE AFTER WRITE CMD
4182  020372  000437                          BR      72$
4183
4184  020374  104022                  71$:    ERROR   22                      ;CERR AFTER WRITE CHECK CMD
4185
4186  020376  012737  010340  005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4187  020404  005037  005500                  CLR     E.B0                    ;EXPECTED MSG B0
4188  020410  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4189  020416  012737  000001  005504          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
4190  020424  005037  005506                  CLR     E.A2                    ;EXPECTED MSG A2
4191  020430  012737  000002  005510          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
4192  020436  012737  000003  005514          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
4193
4194  020444  004737  025604                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
4195  020450  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
4196  020452  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
4197  020454  104031                          ERROR   31                      ;MSH B0 ERROR
4198  020456  104060                          ERROR   60                      ;MSG A1 ERROR
4199  020460  104032                          ERROR   32                      ;MSG B1 ERROR
4200  020462  104401  045333                  TYPE    ,MSG72                  ;ABORTING BALANCE OF TESTS
4201  020466  000137  024006                  JMP     ENDRV
4202
4203  020472                          72$:
4204  020472  104401  044506                  TYPE    ,MSG64                  ;FOLLOWING TEST TEMPORARY BYPASSED
4205
4206                                  ;THE CHANGE OF WRL LOGIC RESULTS IN A DSC AND ATTN TO THE CONTROLLER
4207                                  ;WHICH RECOGNIZES IT AS AN AN ERROR
4208                                  ;THEREFORE, IN CONTINUOUS WRITING RKDA WILL ALWAYS = 1 WHEN THE
4209                                  ;WRL SWITCH IS ENABLED.
4210                                  ;THE FOLLOWING CODE WILL BE VALID IF THE PROPOSED ECO IS APPROVED TO
4211                                  ;ELIMINATE THE CHANGE OF WRL LOGIC
4212
4213  020476  000137  022016                  JMP     12$                     ;EXIT TEST
4214
4215  020502  004737  026772          6$:     JSR     PC,SUBCLR
4216  020506  104024                          ERROR   24                      ;CERR AFTER SCLR
4217
4218  020510  104401  043142                  TYPE    ,MSG53                  ;DISABLE WRITE LOCK
4219  020514  104401  041265                  TYPE    ,MSG37                  ;SPACE BAR WHEN DONE
4220  020520  004737  031006                  JSR     PC,GETSP                ;GET SPACE
4221  020524  004737  026420                  JSR     PC,GSTAT
4222  020530  032737  004000  005434          BIT     #D.WRL,HMR2             ;SEE IF WRITE LOCK OFF
4223  020536  001361                          BNE     6$
4224                                                                          ;TEST FOR WRITE LOCK AT SECTOR BOUNDRIES
```

```
                                                                ;FOR CONTINUOUS WRITING ON CYL 0, TRACK 0,1,2
020540  104401  043233                      TYPE    ,MSG54          ;SET WRITE LOCK
020544  013737  001436  001160              MOV     T5000,$TMP0
020552  005037  001162                      CLR     $TMP1           ;BIT0=0:WRITE 0'S; BIT0=1;WRITE 1'S
020556  005037  001366                      CLR     TOCYL

020562  004737  026772          7$:         JSR     PC,SUBCLR
020566  104024                              ERROR   24              ;CERR AFTER SCLR

020570  032737  000001  001162              BIT     #BIT0,$TMP1
020576  001004                              BNE     8$              ;BR IF WRITING 1'S
020600  012765  001524  000004              MOV     #DATA0,RKBA(R5) ;SETUP ALL 0'S
020606  000403                              BR      9$

020610  012765  001530  000004  8$:         MOV     #DATA1,RKBA(R5) ;SETUP ALL 1'S
020616  052765  000020  000010  9$:         BIS     #BAI,RKCS2(R5)
020624  012765  140400  000002              MOV     #-63.*256.,RKWC(R5)     ;WRITE TRACK 0,1,2  63 SECTORS
020632  005065  000006                      CLR     RKDA(R5)        ;BEGIN AT TRACK AND SEC 0
020636  013765  001366  000020              MOV     TOCYL,RKDC(R5)
020644  012765  000023  000000              MOV     #WRDATA,RKCS1(R5)
020652  013737  001440  005444              MOV     T50000,TEMP1
020660  004737  025062                      JSR     PC,FRDY         ;FIND CONTR RDY
020664  104011                              ERROR   11              ;CONTR NOT RDY

020666  032737  100000  005406              BIT     #CERR,HCS1
020674  001017                              BNE     13$
020676  005337  001160                      DEC     $TMP0
020702  001011                              BNE     10$
020704  104204                              ERROR   204             ;CERR NOT SET BY TIMEOUT
020706  104401  043142                      TYPE    ,MSG53          ;DISABLE WRITE LOCK
020712  104401  041265                      TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
020716  004737  031006                      JSR     PC,GETSP        ;INPUT SPACE
020722  000137  022016                      JMP     12$             ;EXIT TEST

020726  005237  001162          10$:        INC     $TMP1
020732  000713                              BR      7$              ;GO WRITE OPPOSITE DATA
020734  032737  000200  005422  13$:        BIT     #BSE,HER        ;SEE IF BAD SECTOR
020742  001411                              BEQ     21$             ;BR IF NO
020744  005237  001366                      INC     TOCYL           ;ELSE TRY ANOTHER CYL
020750  023727  001366  000012              CMP     TOCYL,#10.      ;SEE IF 10 CONSEC CYL BAD
020756  001301                              BNE     7$              ;BR IF NO & DO AGAIN
020760  104062                              ERROR   62              ;10 BAD CONSEC CYLINDERS
020762  000137  022016                      JMP     12$             ;EXIT TEST

020766  032737  004000  005436  21$:        BIT     #D.WLE,HMR3
020774  001001                              BNE     11$
020776  104205                              ERROR   205             ;NO WRL BY TIMEOUT

021000  104401  043142          11$:        TYPE    ,MSG53          ;DISABLE WRITE LOCK
021004  104401  041265                      TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
021010  004737  031006                      JSR     PC,GETSP

021014  013737  005416  001164              MOV     HDA,$TMP2       ;STORE RKDA OF ERROR
021022  013737  005416  001166              MOV     HDA,$TMP3       ;STORE FOR ERROR TYPEOUT

021030  023727  001164  000001              CMP     $TMP2,#1        ;SEE IF WRL ON TRK 0, SEC 1
```

```
4281  021036  001003                        BNE     22$             ;BR IF NO
4282  021040  104401  045531                TYPE    ,MSG100         ;ELSE WRL OCCURRED ON TRK 2, SEC 21
4283  021044  000616                        BR      6$              ;& NO NEW DATA XFER TOOK PLACE
4284
4285  021046  005737  001164        22$:    TST     $TMP2           ;SEE IF WRL ON TRK 0, SEC 0
4286  021052  001004                        BNE     23$             ;BR IN NO
4287  021054  104401  045531                TYPE    ,MSG100         ;ELSE WRL ON TRK 2, SEC 20
4288  021060  000137  020502                JMP     6$              ;& NO NEW DATA XFER TOOK PLACE
4289
4290  021064  023727  001164  001025  23$:    CMP     $TMP2,#1025     ;SEE IF WRL ON TRK 2, SEC 21
4291  021072  001004                        BNE     24$             ;BE IF NO
4292  021074  104401  045531                TYPE    ,MSG100         ;ELSE WRL ON TRK 2, SEC 19
4293  021100  000137  020502                JMP     6$              ;& NO NEW DATA XFER TOOK PLACE
4294
4295  021104  023727  001164  001024  24$:    CMP     $TMP2,#1024     ;SEE IF WRL ON TRK 2, SEC 20
4296  021112  001004                        BNE     25$             ;BR IF NO
4297  021114  104401  045531                TYPE    ,MSG100         ;ELSE WRL ON TRK 2, SEC 18
4298  021120  000137  020502                JMP     6$              ;& NO OLD DATA TO CHECK AGAINST
4299
4300  021124  023727  001164  000400  25$:    CMP     $TMP2,#400      ;SEE IF WRL AT TRK 1, SEC 0
4301  021132  001004                        BNE     18$             ;BR IF NO
4302  021134  012765  000025  000006        MOV     #21.,RKDA(R5)   ;ELSE SECTOR AT WRL IS TRK 0, SEC 21
4303  021142  000415                        BR      20$
4304
4305  021144  023727  001164  001000  18$:    CMP     $TMP2,#1000     ;SEE IF WRL AT TRK 2, SEC 0
4306  021152  001004                        BNE     19$             ;BR IF NO
4307  021154  012765  000425  000006        MOV     #425,RKDA(R5)   ;ELSE SECTOR AT WRL IS TRK 1, SEC 21
4308  021162  000405                        BR      20$
4309
4310  021164  005337  001164        19$:    DEC     $TMP2           ;GET SECTOR AT WRL
4311  021170  013765  001164  000006        MOV     $TMP2,RKDA(R5)
4312
4313  021176  016537  000006  001166  20$:    MOV     RKDA(R5),$TMP3  ;FOR ERROR PRINTOUT
4314
4315
4316
4317  021204  004737  026772                JSR     PC,SUBCLR
4318  021210  104024                        ERROR   24              ;CERR AFTER SCLR
4319
4320  021212  005737  001164                TST     $TMP2           ;SEE IF TRK/SECTOR 0
4321  021216  001414                        BEQ     80$             ;REPEAT,NO NEW DATA XFER TOOK PLACE
4322  021220  023727  001164  001023        CMP     $TMP2,#1023     ;SEE IF TRK 2,SECTOR 19
4323  021226  001410                        BEQ     80$             ;REPEAT,NO OLD DATA TO CHECK AGAINST
4324  021230  032737  000001  001162        BIT     #BIT0,$TMP1
4325  021236  001006                        BNE     73$             ;BR IF WRITING 1'S WHEN WLE OCCURRED
4326  021240  012765  001530  000004        MOV     #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4327  021246  000405                        BR      74$
4328
4329  021250  000137  020502          80$:    JMP     6$
4330
4331  021254  012765  001524  000004  73$:    MOV     #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4332  021262  052765  000020  000010  74$:    BIS     #BAI,RKCS2(R5)
4333  021270  012765  177400  000002        MOV     #-256.,RKWC(R5)
4334  021276  013765  001166  000006        MOV     $TMP3,RKDA(R5)  ;REFRESH RKDA
4335  021304  017537  000004  001504        MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
4336  021312  013765  001366  000020        MOV     TOCYL,RKDC(R5)
```

```
4337
4338  021320  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4339  021326  013737  001440  005444          MOV     T50000,TEMP1            ;SETUP TIMEOUT
4340  021334  004737  025062                  JSR     PC,FRDY                 ;FIND RDY
4341  021340  104015                          ERROR   15                      ;NO RDY AFTER WRITE CHECK CMD
4342  021342  004737  026420                  JSR     PC,GSTAT                ;GET FRESH STATUS
4343  021346  032737  100000  005406          BIT     #CERR,HCS1
4344  021354  001450                          BEQ     82$
4345  021356  032737  040000  005410          BIT     #WCE,HCS2               ;SEE IF WRITE CHECK ERROR
4346  021364  001405                          BEQ     81$
4347  021366  016537  000024  001502          MOV     RKDB(R5),WD1            ;ACTUAL WORD FOR PRINTOUT
4348  021374  104134                          ERROR   134                     ;WCE AFTER WRITE CMD
4349  021376  000437                          BR      82$
4350
4351  021400  104022                  81$:    ERROR   22                      ;CERR AFTER WRITE CHECK CMD
4352
4353  021402  012737  010340  005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4354  021410  005037  005500                  CLR     E.B0                    ;EXPECTED MSG B0
4355  021414  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4356  021422  012737  000001  005504          MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
4357  021430  005037  005506                  CLR     E.A2                    ;EXPECTED MSG A2
4358  021434  012737  000002  005510          MOV     #2,E.B2                 ;MSG ID FOR EXPECTED MSG B2
4359  021442  012737  000003  005514          MOV     #3,E.B3                 ;MSG ID FOR EXPECTED MSG B3
4360
4361  021450  004737  025604                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
4362  021454  000003                          .WORD   T.A2!T.B2!0                     ;& MSGS SPECIFIED HERE
4363  021456  104057                          ERROR   57                      ;MSG A0 ERROR AFTER WRITE CHECK CMD
4364  021460  104031                          ERROR   31                      ;MSH B0 ERROR
4365  021462  104060                          ERROR   60                      ;MSG A1 ERROR
4366  021464  104032                          ERROR   32                      ;MSG B1 ERROR
4367  021466  104401  045333                  TYPE    ,MSG72                  ;ABORTING BALANCE OF TESTS
4368  021472  000137  024006                  JMP     ENDRV
4369
4370  021476                          82$:
4371  021476  000240                          NOP
4372  021500  000240                          NOP
4373
4374  021502  023727  001164  000400          CMP     $TMP2,#400              ;SEE IF WRL AT TRK 1, SECTOR 0
4375  021510  001004                          BNE     75$                     ;BR IF NO
4376  021512  012765  000025  000006          MOV     #21.,RKDA(R5)           ;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4377  021520  000415                          BR      77$
4378  021522  023727  001164  001000  75$:    CMP     $TMP2,#1000             ;SEE IF WRL AT TRK 2,SECTOR 0
4379  021530  001004                          BNE     76$                     ;BR IF NO
4380  021532  012765  000425  000006          MOV     #425,RKDA(R5)           ;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4381  021540  000405                          BR      77$
4382
4383  021542  005337  001164          76$:    DEC     $TMP2                   ;GET SECTOR BEFORE WRL
4384  021546  013765  001164  000006          MOV     $TMP2,RKDA(R5)
4385  021554  016537  000006  001166  77$:    MOV     RKDA(R5),$TMP3          ;FOR ERROR PRINTOUT
4386  021562  032737  000001  001162          BIT     #BIT0,$TMP1
4387  021570  001004                          BNE     78$                     ;BR IF WRITING 1'S WHEN WLE OCCURRED
4388  021572  012765  001524  000004          MOV     #DATA0,RKBA(R5)         ;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4389  021600  000403                          BR      79$
4390
4391  021602  012765  001530  000004  78$:    MOV     #DATA1,RKBA(R5)         ;WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4392  021610  052765  000020  000010  79$:    BIS     #BAI,RKCS2(R5)
```

```
 4393  021616  012765  177400  000002          MOV     #-256.,RKWC(R5)
 4394  021624  017537  000004  001504          MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
 4395  021632  013765  001366  000020          MOV     TOCYL,RKDC(R5)
 4396
 4397  021640  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
 4398  021646  013737  001440  005444          MOV     T50000,TEMP1    ;SETUP TIMEOUT
 4399  021654  004737  025062                  JSR     PC,FRDY         ;FIND RDY
 4400  021660  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
 4401  021662  004737  026420                  JSR     PC,GSTAT        ;GET FRESH STATUS
 4402  021666  032737  100000  005406          BIT     #CERR,HCS1
 4403  021674  001450                          BEQ     84$
 4404  021676  032737  040000  005410          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
 4405  021704  001405                          BEQ     83$
 4406  021706  016537  000024  001502          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
 4407  021714  104135                          ERROR   135             ;WCE AFTER WRITE CMD
 4408  021716  000437                          BR      84$
 4409
 4410  021720  104022                  83$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
 4411
 4412  021722  012737  010340  005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4413  021730  005037  005500                  CLR     E.B0            ;EXPECTED MSG B0
 4414  021734  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4415  021742  012737  000001  005504          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4416  021750  005037  005506                  CLR     E.A2            ;EXPECTED MSG A2
 4417  021754  012737  000002  005510          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4418  021762  012737  000003  005514          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4419
 4420  021770  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4421  021774  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4422  021776  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 4423  022000  104031                          ERROR   31              ;MSH B0 ERROR
 4424  022002  104060                          ERROR   60              ;MSG A1 ERROR
 4425  022004  104032                          ERROR   32              ;MSG B1 ERROR
 4426  022006  104401  045333                  TYPE    ,MSG72          ;ABORTING BALANCE OF TESTS
 4427  022012  000137  024006                  JMP     ENDRV
 4428
 4429  022016                          84$:
 4430
 4431  022016                          12$:
 4432
 4433                                  ;;*****************************************************************
 4434                                  ;*TEST 20       AC LOW DETECTION PART 2
 4435                                  ;*
 4436                                  ;*      THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
 4437                                  ;*      AT SECTOR BOUNDRIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
 4438                                  ;*      THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
 4439                                  ;*      WHEN THE INTERFACE SHUTS DOWN.
 4440                                  ;*
 4441                                  ;;*****************************************************************
 4442  022016  000004                  TST20:  SCOPE
 4443  022020  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4444  022026  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4445
 4446  022032  004737  026772                  JSR     PC,SUBCLR
 4447  022036  104024                          ERROR   24              ;CERR AFTER SCLR
 4448
```

```
4449  022040  104401  040371          TYPE    ,MSG26          ;AC LOW-PART 2
4450  022044  104401  043041          TYPE    ,MSG52          ;CONT-E IO BYPASS TEST
4451                                                          ;OR SPACE TO CONINUE
4452  022050  004737  030746          JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
4453  022054  000137  024006          JMP     12$             ;RETURN HERE FOR CONT-E
4454                                                          ;RETURN HERE FOR SPACE
4455  022060  005737  001542          TST     BSERR           ;TEST FOR ALIGN CARTRIDGE
4456  022064  001402                  BEQ     1$              ;BR IF NOT ALIGN CART.
4457  022066  000137  022524          JMP     2$
4458
4459  022072  004737  026420  1$:     JSR     PC,GSTAT
4460  022076  032737  004000  005434  BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK
4461  022104  001417                  BEQ     11$             ;BR IF NO
4462
4463  022106  104401  043142          TYPE    ,MSG53          ;DISABLE WRITE LOCK
4464  022112  104401  041265          TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4465  022116  004737  031006          JSR     PC,GETSP        ;GET SPACE
4466
4467  022122  004737  026420          JSR     PC,GSTAT
4468  022126  032737  004000  005434  BIT     #D.WRL,HMR2     ;SEE IF STILL WRITE LOCK
4469  022134  001403                  BEQ     11$             ;BR IF NO
4470  022136  104110                  ERROR   110             ;WRITE LOCK NOT DISABLED
4471  022140  000137  024006          JMP     12$             ;EXIT TEST
4472
4473  022144  005037  001366  11$:     CLR     TOCYL           ;SETUP
4474  022150  005037  001402          CLR     CALADD          ;FOR
4475  022154  005037  001510          CLR     HEAD            ;FILL HEADER TABLE
4476  022160  005037  001516          CLR     FORMAT          ;ROUTINE
4477
4478  022164  004737  027754  13$:     JSR     PC,FHDTAB       ;BUILD STD 22 SECTOR HEADER TABLE
4479
4480  022170  012765  001554  000004  MOV     #HDTAB,RKBA(R5)
4481  022176  012765  177676  000002  MOV     #-66.,RKWC(R5)
4482  022204  000337  001510          SWAB    HEAD
4483  022210  013765  001510  000006  MOV     HEAD,RKDA(R5)   ;HEAD ADDR
4484  022216  000337  001510          SWAB    HEAD
4485
4486  022222  012765  000027  000000  MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
4487  022230  013737  001440  005444  MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4488  022236  004737  025062          JSR     PC,FRDY         ;FIND RDY
4489  022242  104200                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4490  022244  004737  026420          JSR     PC,GSTAT        ;GET FRESH STATUS
4491  022250  032737  100000  005406  BIT     #CERR,HCS1
4492  022256  001401                  BEQ     64$
4493  022260  104201                  ERROR   201             ;CERR AFTER WRITE HEADER CMD
4494  022262                  64$:
4495
4496  022262  005237  001510          INC     HEAD
4497  022266  023727  001510  000003  CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
4498  022274  001333                  BNE     13$             ;BR IF NO
4499  022276  005037  001366          CLR     TOCYL
4500
4501  022302  004737  026772  16$:     JSR     PC,SUBCLR
4502  022306  104024                  ERROR   24              ;CERR AFTER SCLR
4503
4504  022310  012765  001530  000004  MOV     #DATA1,RKBA(R5) ;RETURN HERE FOR SPACE
```

```
4505  022316  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;WRITE INITIAL BACKGROUND OF 1'S
4506  022324  012765  137000  000002          MOV     #-66.*256.,RKWC(R5)     ;TRACK 0,1,2  ALL SECTORS
4507  022332  013765  001366  000020          MOV     TOCYL,RKDC(R5)
4508  022340  012765  000023  000000          MOV     #WRDATA,RKCS1(R5)  ;WRITE DATA CMD
4509  022346  013737  001440  005444          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4510  022354  004737  025062                  JSR     PC,FRDY         ;FIND RDY
4511  022360  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4512  022362  004737  026420                  JSR     PC,GSTAT        ;GET FRESH STATUS
4513  022366  032737  100000  005406          BIT     #CERR,HCS1
4514  022374  001453                          BEQ     2$
4515  022376  032737  000200  005422          BIT     #BSE,HER        ;SEE IF BAD SECTOR
4516  022404  001034                          BNE     17$
4517
4518  022406  012737  010340  005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4519  022414  005037  005500                  CLR     E.B0            ;EXPECTED MSG B0
4520  022420  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4521  022426  012737  000001  005504          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4522  022434  005037  005506                  CLR     E.A2            ;EXPECTED MSG A2
4523  022440  012737  000002  005510          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4524  022446  012737  000003  005514          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4525
4526  022454  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4527  022460  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4528  022462  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4529  022464  104023                          ERROR   23              ;MSH B0 ERROR
4530  022466  104053                          ERROR   53              ;MSG A1 ERROR
4531  022470  104025                          ERROR   25              ;MSG B1 ERROR
4532  022472  000137  024006                  JMP     12$
4533
4534  022476  005237  001366          17$:    INC     TOCYL
4535  022502  023727  001366  000012          CMP     TOCYL,#10.      ;TRIED 10 CYLINDERS?
4536  022510  001003                          BNE     18$             ;BR IF NO
4537  022512  104062                          ERROR   62              ;CANNOT WRITE ON 10 CONSEC CYL
4538  022514  000137  024006                  JMP     12$
4539
4540  022520  000137  022302          18$:    JMP     16$
4541
4542  022524  004737  026772          2$:     JSR     PC,SUBCLR
4543  022530  104024                          ERROR   24              ;CERR AFTER SCLR
4544
4545  022532  104401  042613                  TYPE    ,MSG49          ;TURN OFF AC
4546  022536  104401  043606                  TYPE    ,MSG57          ;VERIFY BATTERY RETRACT FUNCTIONAL
4547
4548  022542  005737  001542                  TST     BSERR           ;SEE IF ALIGN CART USED
4549  022546  001405                          BEQ     15$             ;BR IF NO
4550  022550  104401  041265                  TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4551  022554  004737  031006                  JSR     PC,GETSP        ;GET SPACE
4552  022560  000526                          BR      9$              ;SKIP ALL WRITING
4553
4554  022562  013737  001440  001160  15$:    MOV     T50000,$TMP0
4555  022570  005037  001162                  CLR     $TMP1           ;BIT0=0;WRITE 0'S; BIT0=1:WRITE 1'S
4556
4557  022574  004737  026772          4$:     JSR     PC,SUBCLR
4558  022600  104024                          ERROR   24              ;CERR AFTER SCLR
4559
4560  022602  032737  000001  001162          BIT     #BIT0,$TMP1
```

```
4561  022610  001004                          BNE     5$                      ;BR IF WRITING 1'S
4562  022612  012765  001524  000004          MOV     #DATA0,RKBA(R5) ;SETUP ALL 0'S
4563  022620  000403                          BR      6$
4564
4565  022622  012765  001530  000004  5$:     MOV     #DATA1,RKBA(R5) ;SETUP ALL 1'S
4566  022630  052765  000020  000010  6$:     BIS     #BAI,RKCS2(R5)
4567  022636  012765  140400  000002          MOV     #-63.*256.,RKWC(R5)     ;TRACK 0,1,2  63 SECTORS
4568  022644  005065  000006                  CLR     RKDA(R5)                ;BEGIN AT TRK AND SECTOR 0
4569  022650  013765  001366  000020          MOV     TOCYL,RKDC(R5)
4570  022656  012765  000023  000000          MOV     #WRDATA,RKCS1(R5)
4571  022664  013737  001440  005444          MOV     T50000,TEMP1
4572  022672  004737  025062                  JSR     PC,FRDY                 ;FIND CONTR RDY
4573  022676  104011                          ERROR   11                      ;CONTR NOT RDY
4574
4575  022700  004737  026420                  JSR     PC,GSTAT
4576  022704  032737  100000  005406          BIT     #CERR,HCS1
4577  022712  001017                          BNE     3$
4578  022714  005337  001160                  DEC     $TMP0
4579  022720  001011                          BNE     7$
4580  022722  104146                          ERROR   146                     ;CERR NOT SET BY TIMEOUT
4581  022724  104401  042666                  TYPE    ,MSG50                  ;TURN AC BACK ON
4582  022730  104401  045145                  TYPE    ,MSG69                  ;DEPRESS SPACE AFTER 'READY' LIGHT ON
4583  022734  004737  031006                  JSR     PC,GETSP                ;GET SPACE
4584  022740  000137  024006                  JMP     12$                     ;EXIT TEST
4585
4586  022744  005237  001162          7$:     INC     $TMP1
4587  022750  000711                          BR      4$                      ;GO WRITE OPPOSITE DATA
4588
4589  022752  032737  000100  005436  3$:     BIT     #D.ACLO,HMR3
4590  022760  001001                          BNE     10$
4591  022762  104147                          ERROR   147                     ;AC LOW NOT SET
4592
4593  022764  005237  005370          10$:    INC     UNLD                    ;FOR VALID HALT
4594  022770  012737  070140  005476          MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4595  022776  012737  010300  005500          MOV     #<D.SPLS!D.ACLO!D.FLT>,E.B0
4596  023004  012737  044720  005502          MOV     #<D.UNLD!D.REV!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4597  023012  012737  000001  005504          MOV     #1,E.B1
4598
4599  023020  004737  025604                  JSR     PC,CHKMSG               ;CHECK MSGS A0, B0, A1, B1
4600  023024  000000                          .WORD   0!0!0                   ;& MSGS SPECIFIED HERE
4601  023026  104035                          ERROR   35                      ;MSG A0 ERROR AFTER AC SWITCH OFF FROM HEADS LOADED
4602  023030  104203                          ERROR   203                     ;MSH B0 ERROR
4603  023032  104036                          ERROR   36                      ;MSG A1 ERROR
4604  023034  104037                          ERROR   37                      ;MSG B1 ERROR
4605
4606  023036  013737  005416  001164  8$:     MOV     HDA,$TMP2               ;SAVE RKDA
4607  023044  013737  005416  001166          MOV     HDA,$TMP3               ;SAVE FOR TYPEOUT
4608
4609  023052  104401  042666                  TYPE    ,MSG50                  ;TURN AC BACK ON
4610  023056  104401  045145                  TYPE    ,MSG69                  ;DEPRESS SPACE AFTER 'READY' LIGHT ON
4611  023062  004737  031006                  JSR     PC,GETSP                ;GET SPACE
4612
4613  023066  004737  026772                  JSR     PC,SUBCLR
4614  023072  104024                          ERROR   24                      ;CERR AFTER SCLR
4615
4616  023074  032737  000100  005436          BIT     #D.ACLO,HMR3
```

```
4617  023102  001401                        BEQ     9$
4618  023104  104152                        ERROR   152             ;ACLO NOT RESET AFTER POWER UP
4619
4620  023106  005037  005370          9$:   CLR     UNLD            ;FOR VALID HALT
4621  023112  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
4622  023120  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE #
4623  023126  012765  000003  000000        MOV     #PACK,RKCS1(R5) ;PACK CMD
4624  023134  013737  001426  005444        MOV     T10,TEMP1
4625  023142  004737  025062                JSR     PC,FRDY         ;FIND CONTR RDY
4626  023146  104116                        ERROR   116             ;CONTR NOT RDY
4627
4628  023150  032737  000100  005434        BIT     #D.VV,HMR2
4629  023156  001001                        BNE     65$
4630  023160  104027                        ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
4631  023162                          65$:
4632  023162  005737  001542                TST     BSERR           ;SEE IF ALIGN CART USED
4633  023166  001402                        BEQ     14$             ;BR IF NO
4634  023170  000137  024006                JMP     12$             ;ELSE EXIT TEST
4635
4636  023174                          14$:
4637
4638  023174  004737  026772                JSR     PC,SUBCLR
4639  023200  104024                        ERROR   24              ;CERR AFTER SCLR
4640
4641  023202  005737  001164                TST     $TMP2           ;SEE IF TRK/SECTOR 0
4642  023206  001414                        BEQ     73$             ;REPEAT,NO NEW DATA XFER TOOK PLACE
4643  023210  023727  001164  001023        CMP     $TMP2,#1023     ;SEE IF TRK 2,SECTOR 19
4644  023216  001410                        BEQ     73$             ;REPEAT,NO OLD DATA TO CHECK AGAINST
4645  023220  032737  000001  001162        BIT     #BIT0,$TMP1
4646  023226  001006                        BNE     66$             ;BR IF WRITING 1'S WHEN WLE OCCURRED
4647  023230  012765  001530  000004        MOV     #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4648  023236  000405                        BR      67$
4649
4650  023240  000137  022524          73$:  JMP     2$
4651
4652  023244  012765  001524  000004  66$:  MOV     #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4653  023252  052765  000020  000010  67$:  BIS     #BAI,RKCS2(R5)
4654  023260  012765  177400  000002        MOV     #-256.,RKWC(R5)
4655  023266  013765  001166  000006        MOV     $TMP3,RKDA(R5)  ;REFRESH RKDA
4656  023274  017537  000004  001504        MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
4657  023302  013765  001366  000020        MOV     T0CYL,RKDC(R5)
4658
4659  023310  012765  000031  000000        MOV     #<WRTCHK>.RKCS1(R5)     ;WRITE CHECK CMD
4660  023316  013737  001440  005444        MOV     T50000,TEMP1    ;SETUP TIMEOUT
4661  023324  004737  025062                JSR     PC,FRDY         ;FIND RDY
4662  023330  104015                        ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4663  023332  004737  026420                JSR     PC,GSTAT        ;GET FRESH STATUS
4664  023336  032737  100000  005406        BIT     #CERR,HCS1
4665  023344  001450                        BEQ     75$
4666  023346  032737  040000  005410        BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
4667  023354  001405                        BEQ     74$
4668  023356  016537  000024  001502        MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
4669  023364  104136                        ERROR   136             ;WCE AFTER WRITE CMD
4670  023366  000437                        BR      75$
4671
4672  023370  104022                  74$:  ERROR   22              ;CERR AFTER WRITE CHECK CMD
```

```
4673
4674  023372  012737  010340  005476        MOV   #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4675  023400  005037  005500                CLR   E.B0                ;EXPECTED MSG B0
4676  023404  012737  001720  005502        MOV   #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4677  023412  012737  000001  005504        MOV   #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
4678  023420  005037  005506                CLR   E.A2                ;EXPECTED MSG A2
4679  023424  012737  000002  005510        MOV   #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
4680  023432  012737  000003  005514        MOV   #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
4681
4682  023440  004737  025604                JSR   PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
4683  023444  000003                        .WORD T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
4684  023446  104057                        ERROR 57                  ;MSG A0 ERROR AFTER WRITE CHECK CMD
4685  023450  104031                        ERROR 31                  ;MSH B0 ERROR
4686  023452  104060                        ERROR 60                  ;MSG A1 ERROR
4687  023454  104032                        ERROR 32                  ;MSG B1 ERROR
4688  023456  104401  045333                TYPE  ,MSG72              ;ABORTING BALANCE OF TESTS
4689  023462  000137  024006                JMP   ENDRV
4690
4691  023466                        75$:
4692  023466  000240                        NOP
4693  023470  000240                        NOP
4694
4695  023472  023727  001164  000400        CMP   $TMP2,#400          ;SEE IF WRL AT TRK 1, SECTOR 0
4696  023500  001004                        BNE   68$                 ;BR IF NO
4697  023502  012765  000025  000006        MOV   #21.,RKDA(R5)       ;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4698  023510  000415                        BR    70$
4699  023512  023727  001164  001000  68$:  CMP   $TMP2,#1000         ;SEE IF WRL AT TRK 2,SECTOR 0
4700  023520  001004                        BNE   69$                 ;BR IF NO
4701  023522  012765  000425  000006        MOV   #425,RKDA(R5)       ;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4702  023530  000405                        BR    70$
4703
4704  023532  005337  001164        69$:  DEC   $TMP2                 ;GET SECTOR BEFORE WRL
4705  023536  013765  001164  000006        MOV   $TMP2,RKDA(R5)
4706  023544  016537  000006  001166  70$:  MOV   RKDA(R5),$TMP3      ;FOR ERROR PRINTOUT
4707  023552  032737  000001  001162        BIT   #BIT0,$TMP1
4708  023560  001004                        BNE   71$                 ;BR IF WRITING 1'S WHEN WLE OCCURRED
4709  023562  012765  001524  000004        MOV   #DATA0,RKBA(R5)     ;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4710  023570  000403                        BR    72$
4711
4712  023572  012765  001530  000004  71$:  MOV   #DATA1,RKBA(R5)     ;WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4713  023600  052765  000020  000010  72$:  BIS   #BAI,RKCS2(R5)
4714  023606  012765  177400  000002        MOV   #-256.,RKWC(R5)
4715  023614  017537  000004  001504        MOV   @RKBA(R5),WD2       ;EXPECTED WORD FOR TRUERR TYPEOUT
4716  023622  013765  001366  000020        MOV   TOCYL,RKDC(R5)
4717
4718  023630  012765  000031  000000        MOV   #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4719  023636  013737  001440  005444        MOV   T50000,TEMP1        ;SETUP TIMEOUT
4720  023644  004737  025062                JSR   PC,FRDY             ;FIND RDY
4721  023650  104015                        ERROR 15                  ;NO RDY AFTER WRITE CHECK CMD
4722  023652  004737  026420                JSR   PC,GSTAT            ;GET FRESH STATUS
4723  023656  032737  100000  005406        BIT   #CERR,HCS1
4724  023664  001450                        BEQ   77$
4725  023666  032737  040000  005410        BIT   #WCE,HCS2           ;SEE IF WRITE CHECK ERROR
4726  023674  001405                        BEQ   76$
4727  023676  016537  000024  001502        MOV   RKDB(R5),WD1        ;ACTUAL WORD FOR PRINTOUT
4728  023704  104137                        ERROR 137                 ;WCE AFTER WRITE CMD
```

```
4729  023706  000437                          BR      77$
4730
4731  023710  104022                  76$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4732
4733  023712  012737  010340  005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4734  023720  005037  005500                  CLR     E.B0            ;EXPECTED MSG B0
4735  023724  012737  001720  005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4736  023732  012737  000001  005504          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4737  023740  005037  005506                  CLR     E.A2            ;EXPECTED MSG A2
4738  023744  012737  000002  005510          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4739  023752  012737  000003  005514          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4740
4741  023760  004737  025604                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4742  023764  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4743  023766  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
4744  023770  104031                          ERROR   31              ;MSH B0 ERROR
4745  023772  104060                          ERROR   60              ;MSG A1 ERROR
4746  023774  104032                          ERROR   32              ;MSG B1 ERROR
4747  023776  104401  045333                  TYPE    ,MSG72          ;ABORTING BALANCE OF TESTS
4748  024002  000137  024006                  JMP     ENDRV
4749
4750  024006                          77$:
4751
4752  024006                          12$:
4753
4754  024006                          ENDRV:
4755
4756                                  ;;*****************************************************************
4757                                  ;*TEST 21       END OF PROGRAM
4758                                  ;*
4759                                  ;*      THIS IS NOT A TEST BUT A LINKAGE TO PERFORM ALL THE
4760                                  ;*      ABOVE TESTS FOR THE NEXT DRIVE PRESENT.
4761                                  ;*      THE NEXT TEST IS ENTERED ONLY AFTER ALL DRIVES PRESENT
4762                                  ;*      HAVE BEEN TESTED.
4763                                  ;*      DO NOT LOOP ON THIS 'TEST'.
4764                                  ;*
4765                                  ;;*****************************************************************
4766  024006  000004                  TST21:  SCOPE
4767  024010  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4768  024016  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4769
4770  024022  005237  001220                  INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
4771  024026  023737  005526  001220          CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
4772  024034  001402                          BEQ     TST22           ;;GO TO NEXT TEST IF YES
4773  024036  000137  012276                  JMP     NUDRV           ;IF NOT, GO BACK & TEST NEXT DRIVE PRESENT.
4774
4775                                  ;;*****************************************************************
4776                                  ;*TEST 22       MULTIPLE DRIVE DETECTION TEST
4777                                  ;*
4778                                  ;*      THIS TEST IS PERFORMED ONLY ONCE AT THE END OF PASS 1
4779                                  ;*      AND IS BYPASSED IF ONLY ONE DRIVE IS PRESENT.
4780                                  ;*
4781                                  ;*      THE MULTIPLE DRIVE DETECTION LOGIC IS TESTED BY THE FOLLOWING METHOD:
4782                                  ;*
4783                                  ;*              A. HEADS MUST BE LOADED (SECTOR PULSES REQ'D)
4784                                  ;*              B. THE OPERATOR IS INSTRUCTED TO INSERT THE SAME UNIT SELECT
```

```
                                        :*                PLUG NUMBER (1 PAIR AT A TIME) ON ANY 2 DRIVES
                                        :*                TO BE TESTED
                                        :*             C. THE OPERATOR THEN DEPRESSES THE SPACE BAR TO CONTINUE THE TEST
                                        :*                OR A CONT-E TO EXIT THE TEST
                                        :*
                                        :*      THE PROGRAM VERIFIES THAT MULTIPLE DRIVE SELECT & DRIVE UNSAFE
                                        :*      BOTH SET & THAT THE DRIVE UNLOADS
                                        :*
                                        :*      THE OPERATOR IS ASKED TO VERIFY THAT HEADS UNLOAD FROM BOTH DRIVES
                                        :*
                                        ;;*****************************************************************
024042  000004                  TST22:  SCOPE
024044  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
024052  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR

024056  005737  001216                  TST     $PASS
024062  001402                          BEQ     1$              ;DO TEST ONLY IN 1ST PASS
024064  000137  024370                  JMP     11$             ;ELSE EXIT TEST

024070  023727  005526  000001  1$:     CMP     DRIVS,#1
024076  001004                          BNE     2$              ;BR IF MORE THAN 1 DRIVE PRESENT
024100  104401  044217                  TYPE    ,MSG62          ;BYPASS TEST, ONLY 1 DRIVE PRESENT
024104  000137  024370                  JMP     11$             ;ELSE EXIT TEST

024110  104401  040462          2$:     TYPE    ,MSG28          ;MULT DRV DETECTION TEST
024114  104401  043041                  TYPE    ,MSG52          ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
024120  004737  030746                  JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
024124  000137  024370                  JMP     11$             ;RETURN HERE FOR CONT-E

024130  104401  043654                  TYPE    ,MSG58          ;LOAD HEADS ON ALL DRIVES
024134  104401  041265                  TYPE    ,MSG37          ;PRESS SPACE WHEN SONE
024140  004737  031006                  JSR     PC,GETSP        ;GET SPACE

024144  004737  026772          3$:     JSR     PC,SUBCLR
024150  104024                          ERROR   24              ;CERR AFTER SCLR

024152  104401  043736                  TYPE    ,MSG59          ;INSERT SAME UNIT SEL PLUG # IN 2 DRIVES
024156  104401  041265                  TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
024162  004737  031006                  JSR     PC,GETSP        ;GET SPACE

024166  005000                          CLR     R0              ;DRIVE # COUNTER

024170  012765  100000  000000  6$:     MOV     #CCLR,RKCS1(R5)
024176  010065  000010                  MOV     R0,RKCS2(R5)    ;DRIVE #
024202  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;SELECT DRIVE CMD TO GET STATUS
024210  013737  001426  005444          MOV     T10,TEMP1
024216  004737  025062                  JSR     PC,FRDY         ;FIND CONTR RDY
024222  104117                          ERROR   117             ;NO CONTR RDY

024224  012737  011610  005444          MOV     #5000.,TEMP1
024232  004737  025552                  JSR     PC,DLY          ;REQ 2 MS DELAY BEFORE MDS DETECTED

024236  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
024244  010065  000010                  MOV     R0,RKCS2(R5)
024250  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
024256  013737  001426  005444          MOV     T10,TEMP1
```

```
4841  024264  004737  025062          JSR     PC,FRDY
4842  024270  104117                  ERROR   117             ;NO CONTR RDY
4843
4844  024272  032737  001000  005410  BIT     #MDS,HCS2       ;SEE IF THAT DRIVE HAS MDS
4845  024300  001006                  BNE     7$              ;BR IF YES
4846  024302  005200                  INC     R0              ;ELSE TRY ANOTHER DRIVE
4847  024304  020027  000010          CMP     R0,#8.          ;SEE IF ALL DRIVES TESTED
4848  024310  001327                  BNE     6$              ;BR IF NO
4849  024312  104176                  ERROR   176             ;CANNOT FIND MDS
4850  024314  000411                  BR      10$             ;TRY AGAIN
4851
4852  024316  104401  044151  7$:     TYPE    ,MSG61          ;MULT DRIVES FOUND ON DRIVE #
4853  024322  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
4854                                                          ;;TYPE UNIT NO
4855  024324  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
4856  024326     001                  .BYTE   1               ;;TYPE 1 DIGIT(S)
4857  024327     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
4858  024330  104401  045105          TYPE    ,MSG68          ;VERIFY BOTH DRIVES UNLOADED
4859  024334  005237  005370          INC     UNLD            ;FOR VALID HALT
4860  024340  104401  044041  10$:     TYPE    ,MSG60          ;INSERT CORRECT UNIT SEL PLUG & LOAD HEADS
4861  024344  104401  044313          TYPE    ,MSG63          ;DEPRESS CONT-E OR SPACE BAR WHEN DONE
4862  024350  004737  030746          JSR     PC,CCSP         ;INPUT CONT-E OR SPACE
4863  024354  000137  024370          JMP     11$             ;RETURN HERE FOR CONTROL-E (EXIT)
4864  024360  005037  005370          CLR     UNLD            ;RETURN HERE FOR SPACE (DO AGAIN)
4865  024364  000137  024144          JMP     3$
4866  024370  005037  005370  11$:     CLR     UNLD
4867
4868  024374  004737  026772          JSR     PC,SUBCLR
4869  024400  104024                  ERROR   24              ;CERR AFTER SCLR
```

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

024402                          $EOP:
024402  000004                          SCOPE
024404  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
024410  005037  001174                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
024414  005237  001216                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
024420  042737  100000  001216          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
024426  005327                          DEC     (PC)+           ;;LOOP?
024430  000001                  $EOPCT: .WORD   1
024432  003022                          BGT     $DOAGN          ;;YES
024434  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
024436  000001                  $ENDCT: .WORD   1
024440  024430                          $EOPCT
024442  104401  024507                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
024446  013746  001216                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
024452  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
024454  104401  024504                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
024460  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
024464  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
024466  000005                          RESET                   ;;CLEAR THE WORLD
024470  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
024472  000240                          NOP                     ;;SAVE ROOM
024474  000240                          NOP                     ;;FOR
024476  000240                          NOP                     ;;ACT11
024500                          $DOAGN:
024500  000137                          JMP     @(PC)+          ;;RETURN
024502  010656                  $RTNAD: .WORD   ST5
024504      377       377   000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
024507      015  042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS #/
024514  050040  051501  020123
024522  000043
```

```
                                .SBTTL  SUBROUTINES

                                ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                ;
024524  012700  005516          CLRFLG: MOV     #DDUMP,R0
024530  012701  177757                  MOV     #-17.,R1
024534  005020                  1$:     CLR     (R0)+
024536  005201                          INC     R1
024540  001375                          BNE     1$
024542  000207                          RTS     PC

                                ;
                                ;TYPE PROGRAM ID IF FTITLE=0
                                ;
024544  005737  001360          TITLE:  TST     FTITLE
024550  001024                          BNE     1$
024552  005237  001360                  INC     FTITLE
024556  104401  036220                  TYPE    ,MSG1                   ;PROGRAM ID
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
024562  005737  000042                  TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
024566  001012                          BNE     64$                     ;;BRANCH IF YES
024570  123727  001230  000001          CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
024576  001406                          BEQ     64$                     ;;BRANCH IF YES
024600  023727  001140  000176          CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
024606  001005                          BNE     65$                     ;;BRANCH IF NO
024610  104406                          GTSWR                           ;;GET SOFT-SWR SETTINGS
024612  000403                          BR      65$
024614  112737  000001  001134  64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
024622                          65$:
024622  000207                  1$:     RTS     PC

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

024624  104411                  GDRVS:  RDLIN
024626  012600                          MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
024630  012701  177770                  MOV     #-8.,R1                 ;SET UP COUNT
024634  112002                  1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
024636  042702  177400                  BIC     #177400,R2              ;MASK HI BYTE
024642  012703  005530                  MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
024646  012704  000060                  MOV     #60,R4

024652  020402                  2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
024654  001415                          BEQ     3$                      ;BRANCH IF YES
024656  005723                          TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
024660  005204                          INC     R4
024662  020427  000070                  CMP     R4,#70
024666  001371                          BNE     2$                      ;S/B 0-7 OR TERMINATOR
024670  005702                          TST     R2
024672  001022                          BNE     4$
024674  020127  177770                  CMP     R1,#-8.
024700  001426                          BEQ     6$                      ;DEFAULT ALL DRIVES
```

```
 4964  024702  005037  005556          7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
 4965  024706  000207                          RTS     PC              ;FOUND TERMINATOR, EXIT
 4966
 4967  024710  005213                  3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
 4968  024712  005237  005526                  INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
 4969  024716  112002                          MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
 4970  024720  042702  177400                  BIC     #177400,R2      ;MASK
 4971  024724  022702  000054                  CMP     #54,R2          ;IS IT A COMMA?
 4972  024730  001407                          BEQ     5$              ;YES, GO TO NEXT WORD.
 4973  024732  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
 4974  024734  001001                          BNE     4$              ;IF NOT, SOMETHING WRONG.
 4975  024736  000761                          BR      7$              ;FOUND TERMINATOR, EXIT
 4976
 4977  024740  104401  045700          4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
 4978  024744  000137  010064                  JMP     PRGSRT          ;START ALL OVER
 4979
 4980  024750  005201                  5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
 4981  024752  001330                          BNE     1$              ;DRIVES TYPED IN.
 4982  024754  000771                          BR      4$              ;IF MORE, HAVE ERROR.
 4983
 4984  024756  005237  005556          6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
 4985  024762  000207                          RTS     PC              ;EXIT.
 4986
 4987                                  ;
 4988                                  ;ROUTINE TO INPUT RKBAS OR DEFAULT.
 4989                                  ;
 4990
 4991  024764  104412                  GBA:    RDOCT
 4992  024766  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 4993  024770  005700                          TST     R0
 4994  024772  001403                          BEQ     1$              ;BRANCH IF DEFAULT.
 4995  024774  010037  001264                  MOV     R0,$BASE
 4996  025000  000207                          RTS     PC
 4997  025002  012737  177440  001264  1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
 4998  025010  000207                          RTS     PC
 4999
 5000                                  ;
 5001                                  ;ROUTINE TO INPUT RKVEC OR DEFAULT
 5002                                  ;
 5003
 5004  025012  104412                  GINT:   RDOCT
 5005  025014  012600                          MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 5006  025016  005700                          TST     R0
 5007  025020  001405                          BEQ     1$              ;BRANCH IF DEFAULT
 5008  025022  010037  001334                  MOV     R0,RKVEC
 5009  025026  004737  025044          2$:     JSR     PC,SETINT
 5010  025032  000207                          RTS     PC
 5011  025034  012737  000210  001334  1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
 5012  025042  000771                          BR      2$
 5013
 5014                                  ;
 5015                                  ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
 5016                                  ;
 5017
 5018  025044  013700  001334          SETINT: MOV     RKVEC,R0
 5019  025050  012720  031520                  MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
                                        MOV     RKPRI,(R0)      ;PRS TO RKVEC+2
                                        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
                                ;
                        FRDY:   BIT     #RDY,RKCS1(R5)
                                BNE     1$
                                DEC     TEMP1
                                BNE     FRDY
                                JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
                                JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
                                RTS     PC              ;NO RDY, EXIT
                        1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
                                JSR     PC,HOLD
                                JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
                                RTS     PC
                                ;
                                ;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
                                ;
                        FRDY1:  BIT     #RDY,RKCS1(R5)
                                BNE     1$
                                DEC     TEMP1
                                BNE     FRDY1
                                MOV     RKMR2(R5),HMR2
                                MOV     RKMR3(R5),HMR3
                                JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
                                RTS     PC              ;NO RDY, EXIT
                        1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
                                MOV     RKMR2(R5),HMR2
                                MOV     RKMR3(R5),HMR3
                                JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
                                RTS     PC

                                ;
                                ;STORE ALL RK611 REGISTERS IN HOLDING REGS
                                ;
                        HOLD:   MOV     RKCS1(R5),HCS1
                                MOV     RKCS2(R5),HCS2
                                MOV     RKWC(R5),HWC
                                MOV     RKBA(R5),HBA
                                MOV     RKDA(R5),HDA
                                MOV     RKDS(R5),HDS
                                MOV     RKER(R5),HER
                                MOV     RKASOF(R5),HASOF
                                MOV     RKDC(R5),HDC
                                MOV     RKMR1(R5),HMR1
                                MOV     RKMR2(R5),HMR2
                                MOV     RKMR3(R5),HMR3
```

```
5076  025326  016537  000030  005440          MOV     RKECPS(R5),HPOS
5077  025334  016537  000032  005442          MOV     RKECPT(R5),HPAT
5078  025342  000207                          RTS     PC
5079
5080                                  ;
5081                                  ;ROUTINE TO CHECK FOR CORRECT ATTN
5082                                  ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
5083                                  ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
5084                                  ;
5085  025344  010446                  TSTATN: MOV     R4,-(SP)        ;SAV R4
5086  025346  013704  001222                  MOV     $UNIT,R4
5087  025352  136437  005376  005425          BITB    ATTN(R4),HASOF+1
5088  025360  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
5089  025362  012604                          MOV     (SP)+,R4        ;RESTOR R4
5090  025364  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
5091  025370  000207                          RTS     PC
5092  025372  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
5093  025374  000207                          RTS     PC
5094
5095
5096                                  ;
5097                                  ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
5098                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
5099                                  ;RETURN IF NO ATTN (ERROR CONDITION)
5100                                  ;RETURN +2 IF ATTN FOUND
5101                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
5102                                  ;
5103
5104  025376  010446                  FATT1:  MOV     R4,-(SP)        ;SAV R4
5105  025400  012737  177777  005444  3$:     MOV     #-1,TEMP1
5106  025406  013704  001222                  MOV     $UNIT,R4
5107  025412  136465  005376  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
5108  025420  001014                          BNE     2$
5109  025422  005337  005444                  DEC     TEMP1
5110  025426  001371                          BNE     1$
5111  025430  005337  005446                  DEC     TEMP2
5112  025434  001361                          BNE     3$
5113  025436  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
5114  025442  004737  026420                  JSR     PC,GSTAT        ;GET LATEST STATUS
5115  025446  012604                          MOV     (SP)+,R4        ;RESTOR R4
5116  025450  000207                          RTS     PC
5117
5118  025452  005065  000026          2$:     CLR     RKMR1(R5)
5119  025456  004737  026420                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
5120  025462  012604                          MOV     (SP)+,R4        ;RESTOR R4
5121  025464  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
5122  025470  000207                          RTS     PC
5123
5124                                  ;
5125                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
5126                                  ;ENTER WITH COUNT IN TEMP1
5127                                  ;RETURN IF NO ATTN (ERROR)
5128                                  ;RETURN +2 IF ATTN FOUND
5129                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
5130                                  ;
5131
```

```
5132  025472  010446                  FATT2:  MOV     R4,-(SP)        ;SAV R4
5133  025474  013704  001222          2$:     MOV     $UNIT,R4
5134  025500  136465  005376  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
5135  025506  001011                          BNE     1$
5136  025510  005337  005444                  DEC     TEMP1
5137  025514  001367                          BNE     2$
5138  025516  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
5139  025522  004737  026420                  JSR     PC,GSTAT        ;GET LATEST STATUS.
5140  025526  012604                          MOV     (SP)+,R4        ;RESTOR R4
5141  025530  000207                          RTS     PC
5142  025532  005065  000026          1$:     CLR     RKMR1(R5)
5143  025536  004737  026420                  JSR     PC,GSTAT
5144  025542  012604                          MOV     (SP)+,R4        ;RESTOR R4
5145  025544  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
5146  025550  000207                          RTS     PC
5147                                  ;
5148                                  ;ENTER WITH A COUNT IN TEMP1
5149                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
5150                                  ;WHEN COUNT IS 0. BASED ON AN 11/05
5151                                  ;
5152  025552  005737  005444          DLY:    TST     TEMP1           ;5.6 US
5153  025556  001403                          BEQ     1$              ;2.5 US
5154  025560  005337  005444                  DEC     TEMP1           ;6.8 US
5155  025564  000772                          BR      DLY             ;2.5 US
5156  025566  000207                  1$:     RTS     PC              ;3.8 US
5157                                  ;
5158                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
5159                                  ;
5160
5161  025570  104401  037705          BYP:    TYPE    ,MSG14          ;BYPASS DRIVE
5162  025574  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
5163                                                                  ;;TYPE DR#
5164  025576  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
5165  025600     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
5166  025601     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
5167  025602  000207                          RTS     PC
5168                                  ;
5169                                  ;THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
5170                                  ;
5171  025604  017637  000000  001552  CHKMSG: MOV     @(SP),CHKFLG    ;PASS MSGS TO BE TESTED
5172  025612  062716  000002                  ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
5173  025616  004737  026462                  JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS
5174
5175  025622  053737  001222  005476          BIS     $UNIT,E.A0      ;SET UNIT #
5176  025630  053737  001222  005502          BIS     $UNIT,E.A1
5177  025636  053737  001222  005506          BIS     $UNIT,E.A2
5178  025644  053737  001222  005512          BIS     $UNIT,E.A3
5179
5180  025652  013746  005444                  MOV     TEMP1,-(SP)     ;SAVE TEMP1
5181
5182  025656  013737  005476  005444          MOV     E.A0,TEMP1
5183  025664  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A0
5184  025670  013737  005444  005476          MOV     TEMP1,E.A0
5185
5186  025676  013737  005502  005444          MOV     E.A1,TEMP1
5187  025704  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A1
```

```
5188  025710  013737  005444  005502          MOV     TEMP1,E.A1
5189
5190  025716  013737  005506  005444          MOV     E.A2,TEMP1
5191  025724  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
5192  025730  013737  005444  005506          MOV     TEMP1,E.A2
5193
5194  025736  013737  005500  005444          MOV     E.B0,TEMP1
5195  025744  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
5196  025750  013737  005444  005500          MOV     TEMP1,E.B0
5197
5198  025756  013737  005504  005444          MOV     E.B1,TEMP1
5199  025764  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
5200  025770  013737  005444  005504          MOV     TEMP1,E.B1
5201
5202  025776  013737  005510  005444          MOV     E.B2,TEMP1
5203  026004  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
5204  026010  013737  005444  005510          MOV     TEMP1,E.B2
5205
5206  026016  013737  005514  005444          MOV     E.B3,TEMP1
5207  026024  004737  030642                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
5208  026030  013737  005444  005514          MOV     TEMP1,E.B3
5209
5210  026036  012637  005444                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
5211  026042  013737  001176  001172          MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE
5212
5213  026050  023737  005456  005476          CMP     H.A0,E.A0       ;TEST MSG A0
5214  026056  001411                          BEQ     2$              ;BR IF OK
5215  026060  012737  026072  001176          MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
5216  026066  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR.
5217  026070  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
5218
5219  026072  032777  001000  153040  1$:     BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
5220  026100  001107                          BNE     20$             ;& BR IF LOOP ON ERROR
5221  026102  062716  000002          2$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
5222
5223  026106  023737  005460  005500          CMP     H.B0,E.B0       ;TEST MSG B0
5224  026114  001411                          BEQ     5$              ;BR IF OK
5225  026116  012737  026130  001176          MOV     #4$,$ESCAPE     ;ELSE SETUP ESCAPE
5226  026124  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR
5227  026126  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
5228
5229  026130  032777  001000  153002  4$:     BIT     #SW9,@SWR       ;RETURN HERE FROM MAINLINE ERROR
5230  026136  001070                          BNE     20$             ;& BR IF LOOP ON ERROR
5231  026140  062716  000002          5$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
5232
5233  026144  023737  005462  005502          CMP     H.A1,E.A1       ;TEST MSG A1
5234  026152  001411                          BEQ     8$              ;BR IF OK
5235  026154  012737  026166  001176          MOV     #7$,$ESCAPE
5236  026162  011646                          MOV     (SP),-(SP)
5237  026164  000207                          RTS     PC
5238
5239  026166  032777  001000  152744  7$:     BIT     #SW9,@SWR
5240  026174  001051                          BNE     20$
5241  026176  062716  000002          8$:     ADD     #2,(SP)
5242
5243  026202  023737  005464  005504          CMP     H.B1,E.B1       ;TEST MSG B1
```

```
5244  026210  001411                        BEQ     11$                     ;BR IF OK
5245  026212  012737  026224  001176        MOV     #10$,$ESCAPE
5246  026220  011646                        MOV     (SP),-(SP)
5247  026222  000207                        RTS     PC
5248
5249  026224  032777  001000  152706  10$:  BIT     #SW9,@SWR
5250  026232  001032                        BNE     20$
5251  026234  062716  000002          11$:  ADD     #2,(SP)
5252
5253  026240  032737  000001  001552  12$:  BIT     #T.A2,CHKFLG            ;TEST MSG A2?
5254  026246  001402                        BEQ     13$                     ;BR IF NO
5255  026250  004737  027256                JSR     PC,RCYLD                ;PUT INFO CYLDIF, DO NOT CHECK
5256  026254  032737  000002  001552  13$:  BIT     #T.B2,CHKFLG            ;TEST MSG B2?
5257  026262  001402                        BEQ     14$                     ;BR IF NO
5258  026264  004737  027330                JSR     PC,RCYLA                ;PUT INFO IN CYLADD, DO NOT CHECK
5259
5260  026270  032737  000004  001552  14$:  BIT     #T.B3,CHKFLG            ;TEST MSG B3?
5261  026276  001404                        BEQ     15$
5262  026300  004737  027366                JSR     PC,RSEC                 ;PUT INFO IN SECTOR, DO NOT CHECK
5263  026304  004737  027424                JSR     PC,RHEAD                ;PUT INFO IN HEADA, DO NOT CHECK
5264
5265  026310  013737  001172  001176  15$:  MOV     $TMP5,$ESCAPE           ;RESTORE ESCAPE
5266  026316  000207                        RTS     PC
5267
5268  026320  012706  001100          20$:  MOV     #STACK,SP               ;RESET STACK PTR
5269  026324  013737  001172  001176        MOV     $TMP5,$ESCAPE           ;RESTORE ESCAPE
5270  026332  000177  152552                JMP     @$LPERR
5271                                  ;
5272                                  ;THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
5273                                  ;I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
5274                                  ;
5275  026336  005737  001550          CKCERR: TST     BYPCERR
5276  026342  001025                        BNE     4$
5277  026344  032737  100000  005406        BIT     #CERR,HCS1
5278  026352  001001                        BNE     1$                      ;BR IF CERR
5279  026354  000207                        RTS     PC
5280
5281  026356  032737  004000  005406  1$:   BIT     #CTO,HCS1
5282  026364  001402                        BEQ     2$                      ;BR IF NOT CTO
5283  026366  104211                        ERROR   211                     ;CTO ERROR, MSG A & B INVALID
5284  026370  000207                        RTS     PC
5285
5286  026372  032737  010000  005410  2$:   BIT     #NED,HCS2
5287  026400  001401                        BEQ     3$                      ;BR IF NOT NED
5288  026402  104212                        ERROR   212                     ;NED ERROR, MSG A & B INVALID
5289
5290  026404  032737  001000  005410  3$:   BIT     #MDS,HCS2
5291  026412  001401                        BEQ     4$
5292  026414  104213                        ERROR   213                     ;MDS ERROR, MSG A & B INVALID
5293
5294  026416  000207                  4$:   RTS     PC
5295
5296                                  ;
5297                                  ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
5298                                  ;IT THEN WAITS FOR CONTROLLER READY
5299
```

```
                                        ;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
                                        ;
026420 013746 005444            GSTAT:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
026424 013765 001222 000010             MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
026432 012765 000001 000000             MOV     #SELDRV,RKCS1(R5)  ;GET STATUS WITH SELECT DRIVE CMD
026440 013737 001426 005444             MOV     T10,TEMP1
026446 004737 025062                    JSR     PC,FRDY         ;FIND RDY
026452 104117                           ERROR   117             ;RDY NOT SET BY END OF SELECT DRIVE CMD
026454 012637 005444                    MOV     (SP)+,TEMP1     ;RESTOR TEMP1.
026460 000207                           RTS     PC
                                        ;
                                        ;THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
                                        ;& ALL CONTROLLER REGISTERS.
                                        ;
026462 013746 005444            GSTAT1: MOV     TEMP1,-(SP)     ;SAVE TEMP1
026466 004737 025216                    JSR     PC,HOLD         ;GET ALL CONTR REG
026472 012765 100000 000000             MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
026500 013765 001222 000010             MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
026506 012765 000003 000026             MOV     #3,RKMR1(R5)    ;SELECT WORD 3
026514 012765 000001 000000             MOV     #SELDRV,RKCS1(R5)  ;GET STATUS
026522 013737 001426 005444             MOV     T10,TEMP1
026530 004737 025130                    JSR     PC,FRDY1                ;FIND RDY & STORE DRIVE REGS ONLY
026534 104117                           ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
026536 013737 005434 005472             MOV     HMR2,H.A3       ;STORE MSG A3
026544 013737 005436 005474             MOV     HMR3,H.B3       ;STORE MSG B3
                                        
026552 012765 100000 000000             MOV     #CCLR,RKCS1(R5)
026560 013765 001222 000010             MOV     $UNIT,RKCS2(R5)
026566 012765 000002 000026             MOV     #2,RKMR1(R5)    ;SELECT WORD 2
026574 012765 000001 000000             MOV     #SELDRV,RKCS1(R5)
026602 013737 001426 005444             MOV     T10,TEMP1
026610 004737 025130                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
026614 104117                           ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
026616 013737 005434 005466             MOV     HMR2,H.A2       ;STORE MSG A2
026624 013737 005436 005470             MOV     HMR3,H.B2       ;STORE MSG B2
                                        
026632 012765 100000 000000             MOV     #CCLR,RKCS1(R5)
026640 013765 001222 000010             MOV     $UNIT,RKCS2(R5)
026646 012765 000001 000026             MOV     #1,RKMR1(R5)    ;SELECT WORD 1
026654 012765 000001 000000             MOV     #SELDRV,RKCS1(R5)
026662 013737 001426 005444             MOV     T10,TEMP1
026670 004737 025130                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
026674 104117                           ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
026676 013737 005434 005462             MOV     HMR2,H.A1       ;STORE MSG A1
026704 013737 005436 005464             MOV     HMR3,H.B1       ;STORE MSG B1
                                        
026712 012765 100000 000000             MOV     #CCLR,RKCS1(R5)
026720 013765 001222 000010             MOV     $UNIT,RKCS2(R5)
026726 012765 000001 000000             MOV     #SELDRV,RKCS1(R5)  ;SELECT WORD 0
026734 013737 001426 005444             MOV     T10,TEMP1
026742 004737 025130                    JSR     PC,FRDY1        ;FIND RDY & STORE DRIVE REGS ONLY
026746 104117                           ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
026750 013737 005434 005456             MOV     HMR2,H.A0       ;STORE MSG A0
026756 013737 005436 005460             MOV     HMR3,H.B0       ;STORE MSG B0
```

```
                                        MOV     (SP)+,TEMP1     ;RESTORE TEMP1
                                        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
                                ;
026772  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
027000  013737  001426  005444          MOV     T10,TEMP1
027006  004737  025062                  JSR     PC,FRDY         ;FIND RDY
027012  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
027014  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
027022  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
027026  004737  026420                  JSR     PC,GSTAT        ;GET STATUS
027032  032737  100000  005406          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
027040  001401                          BEQ     1$
027042  000207                          RTS     PC
027044  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
027050  000207                          RTS     PC

                                ;
                                ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                ;
027052  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
027060  004737  026420                  JSR     PC,GSTAT
027064  013737  005436  001422          MOV     HMR3,SECTOR
027072  042737  177017  001422          BIC     #^C<M.SECT>,SECTOR
027100  006237  001422                  ASR     SECTOR          ;RIGHT JUSTIFY
027104  006237  001422                  ASR     SECTOR          ;SECTOR
027110  006237  001422                  ASR     SECTOR          ;INFO
027114  006237  001422                  ASR     SECTOR
027120  000207                          RTS     PC

                                ;
                                ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                ;
027122  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
027130  004737  026420                  JSR     PC,GSTAT
027134  013737  005434  001376          MOV     HMR2,CYLDIF

027142  042737  160017  001376          BIC     #^C<M.CDIF>,CYLDIF
027150  006237  001376                  ASR     CYLDIF          ;RIGHT JUSTIFY
027154  006237  001376                  ASR     CYLDIF          ;CYL DIFF/OFFSET
027160  006237  001376                  ASR     CYLDIF          ;INFO
027164  006237  001376                  ASR     CYLDIF
027170  023727  001376  000777          CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
027176  001002                          BNE     1$              ;BR IF NOT
027200  005037  001376                  CLR     CYLDIF          ;CLR IF YES
027204  000207                  1$:     RTS     PC

                                ;
                                ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
                                ;
027206  012765  000002  000026  RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
027214  004737  026420                  JSR     PC,GSTAT
```

```
5412  027220  013737  005436  001400          MOV     HMR3,CYLADD
5413  027226  042737  160017  001400          BIC     #↑C<M.CADD>,CYLADD
5414  027234  006237  001400                  ASR     CYLADD          ;RIGHT JUSTIFY
5415  027240  006237  001400                  ASR     CYLADD          ;CYL ADDR
5416  027244  006237  001400                  ASR     CYLADD          ;INFO
5417  027250  006237  001400                  ASR     CYLADD
5418  027254  000207                          RTS     PC
5419                                  ;
5420                                  ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
5421                                  ;
5422  027256  013737  005466  001376  RCYLD:  MOV     H.A2,CYLDIF
5423  027264  042737  160017  001376          BIC     #↑C<M.CDIF>,CYLDIF  ;CLEAR UNWANTED INFO
5424  027272  006237  001376                  ASR     CYLDIF  ;RIGHT JUSTIFY
5425  027276  006237  001376                  ASR     CYLDIF
5426  027302  006237  001376                  ASR     CYLDIF
5427  027306  006237  001376                  ASR     CYLDIF
5428  027312  023727  001376  000777          CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
5429  027320  001002                          BNE     1$              ;BR IF NO
5430  027322  005037  001376                  CLR     CYLDIF          ;ELSE CLEAR
5431  027326  000207                  1$:     RTS     PC
5432                                  ;
5433                                  ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
5434                                  ;
5435  027330  013737  005470  001400  RCYLA:  MOV     H.B2,CYLADD
5436  027336  042737  160017  001400          BIC     #↑C<M.CADD>,CYLADD  ;CLEAR UNWANTED INFO
5437  027344  006237  001400                  ASR     CYLADD          ;RIGHT JUSTIFY
5438  027350  006237  001400                  ASR     CYLADD
5439  027354  006237  001400                  ASR     CYLADD
5440  027360  006237  001400                  ASR     CYLADD
5441  027364  000207                          RTS     PC
5442                                  ;
5443                                  ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
5444                                  ;
5445  027366  013737  005474  001422  RSEC:   MOV     H.B3,SECTOR
5446  027374  042737  177017  001422          BIC     #↑C<M.SECT>,SECTOR  ;CLEAR UNWANTED INFO
5447  027402  006237  001422                  ASR     SECTOR          ;RIGHT JUSTIFY
5448  027406  006237  001422                  ASR     SECTOR
5449  027412  006237  001422                  ASR     SECTOR
5450  027416  006237  001422                  ASR     SECTOR
5451  027422  000207                          RTS     PC
5452                                  ;
5453                                  ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
5454                                  ;
5455  027424  013737  005474  001512  RHEAD:  MOV     H.B3,HEADA
5456  027432  042737  170777  001512          BIC     #↑C<M.HEAD>,HEADA  ;CLEAR UNWANTED INFO
5457  027440  006237  001512                  ASR     HEADA           ;RIGHT JUSTIFY IT
5458  027444  000337  001512                  SWAB    HEADA
5459  027450  000207                          RTS     PC
5460                                  ;
5461                                  ;FIND SECTOR 17
5462                                  ;RETURN IF NOT FOUND
5463                                  ;RETURN +4 IF FOUND
5464                                  ;
5465  027452  013737  001436  005444  FSEC17: MOV     T5000,TEMP1     ;SETUP TIMEOUT
5466  027460  004737  027052          1$:     JSR     PC,RDSEC        ;READ SECTOR
5467  027464  023727  001422  000021          CMP     SECTOR,#17.     ;TEST FOR SECTOR 17
```

```
5468  027472  001014                          BNE     2$              ;BR IF NOT 17
5469
5470  027474  004737  027052                  JSR     PC,RDSEC
5471  027500  023727  001422  000021          CMP     SECTOR,#17.
5472  027506  001412                          BEQ     3$              ;BR IF READ SAME TWICE
5473  027510  004737  027052                  JSR     PC,RDSEC        ;ELSE TRY 1 MORE TIME
5474  027514  023727  001422  000021          CMP     SECTOR,#17.
5475  027522  001404                          BEQ     3$              ;BR IF 17
5476
5477  027524  005337  005444          2$:     DEC     TEMP1
5478  027530  001353                          BNE     1$              ;TRY AGAIN
5479  027532  000207                          RTS     PC
5480
5481  027534  062716  000004          3$:     ADD     #4,(SP)         ;SKIP OVER ERROR
5482  027540  000207                          RTS     PC
5483                                  ;
5484                                  ;FIND DESIRED CYL DIFF
5485                                  ;RETURN IF NOT FOUND
5486                                  ;RETURN+6 IF FOUND
5487                                  ;
5488
5489  027542  013737  001436  005444  FCYL:   MOV     T5000,TEMP1     ;SETUP TIMEOUT
5490  027550  004737  027122          1$:     JSR     PC,RDCYLD
5491  027554  023737  001376  005446          CMP     CYLDIF,TEMP2    ;TEST FOR CYL DIFF
5492  027562  001014                          BNE     2$              ;BR IF NOT FOUND
5493
5494  027564  004737  027122                  JSR     PC,RDCYLD
5495  027570  023737  001376  005446          CMP     CYLDIF,TEMP2
5496  027576  001412                          BEQ     3$              ;BR IF READ SAME TWICE
5497  027600  004737  027122                  JSR     PC,RDCYLD       ;ELSE TRY 1 MORE TIME
5498  027604  023737  001376  005446          CMP     CYLDIF,TEMP2
5499  027612  001404                          BEQ     3$
5500
5501  027614  005337  005444          2$:     DEC     TEMP1
5502  027620  001353                          BNE     1$              ;TRY AGAIN
5503  027622  000207                          RTS     PC
5504
5505  027624  062716  000006          3$:     ADD     #6,(SP)         ;SKIP OVER ERROR
5506  027630  000207                          RTS     PC
5507
5508                                  ;
5509                                  ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
5510                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
5511                                  ;RETURN IF NOT FOUND
5512                                  ;RETURN+2 IF FOUND - SKIP OVER ERROR
5513                                  ;
5514  027632  012737  177777  005444  FHDHM:  MOV     #-1,TEMP1       ;ALL 1'S
5515  027640  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
5516  027646  004737  026420          1$:     JSR     PC,GSTAT
5517  027652  032737  000040  005434          BIT     #D.HDHM,HMR2
5518  027660  001007                          BNE     2$
5519  027662  005337  005444                  DEC     TEMP1
5520  027666  001367                          BNE     1$
5521
5522  027670  005337  005446                  DEC     TEMP2
5523  027674  001356                          BNE     FHDHM
```

```
5524   027676  000207                          RTS     PC
5525   027700  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
5526   027704  000207                          RTS     PC
5527                                   ;
5528                                   ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE SMS
5529                                   ;RETURN IF NOT FOUND
5530                                   ;RETURN+2 IF FOUND: SKIP OVER ERROR
5531                                   ;
5532   027706  012737  000372  005444  FLOAD:  MOV     #250.,TEMP1
5533   027714  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
5534   027722  004737  026420          1$:     JSR     PC,GSTAT
5535   027726  032737  010000  005434          BIT     #D.LOAD,HMR2
5536   027734  001004                          BNE     2$
5537   027736  005337  005444                  DEC     TEMP1
5538   027742  001367                          BNE     1$
```

```
                                        RTS     PC
                                2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
                                        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'
                                ;
                                FHDTAB: MOV     R0,-(SP)        ;SAV R0
                                        MOV     R1,-(SP)        ;SAV R1
                                        MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
                                        CLR     R1              ;SECTOR COUNTER
                                        MOV     HEAD,HD1
                                        ASL     HD1
                                        ASL     HD1
                                        ASL     HD1
                                        ASL     HD1
                                        ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
                                        MOV     FORMAT,FMT1
                                        SWAB    FMT1
                                        ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER

                                1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
                                        MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
                                        BIS     HD1,(R0)        ;               -HEAD NO
                                        BIS     FMT1,(R0)       ;               -FORMAT
                                        JSR     PC,SECFLG       ;GET SECTOR FLAGS

                                        MOV     CALADD,TEMP1
                                        MOV     (R0),TEMP2
                                        BIC     CALADD,TEMP2
                                        BIC     (R0)+,TEMP1
                                        BIS     TEMP1,TEMP2
                                        MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK

                                        INC     R1              ;SECTOR CTR
                                        CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
                                        BNE     1$              ;BR IF NO

                                        MOV     (SP)+,R1        ;RESTOR R1
                                        MOV     (SP)+,R0        ;RESTOR R0
                                        RTS     PC
                                ;
                                ;THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
                                ;TEST & SETS BITS 14 & 15 APPROPRIATLY.
                                ;
                                SECFLG: MOV     R2,-(SP)        ;SAVE R2
                                        TST     FORMAT
                                        BNE     1$              ;BR IF 20 SECTOR FORMAT
                                        MOV     #BSE22H+8.,R2
                                        JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
                                        BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR

                                        MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
                                        JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
```

```
5595  030170  052710  040000          BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
5596
5597  030174  012602                  MOV     (SP)+,R2        ;ELSE RETURN HERE
5598  030176  000207                  RTS     PC
5599
5600  030200  012602          1$:     MOV     (SP)+,R2        ;RESTORE R2
5601  030202  000207                  RTS     PC
5602
5603                          ;
5604                          ; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES.
5605                          ;ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
5606                          ;RETURN IF NO COMPARE
5607                          ;RETURN+4 IF COMPARE
5608                          ;
5609  030204  010346          FLGTST: MOV     R3,-(SP)        ;SAVE R3
5610
5611  030206  021227  177777  1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
5612  030212  001002                  BNE     2$              ;BR IF NO
5613  030214  012603                  MOV     (SP)+,R3        ;RESTORE R3
5614  030216  000207                  RTS     PC
5615
5616  030220  022237  001402  2$:     CMP     (R2)+,CALADD    ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
5617  030224  001403                  BEQ     3$
5618  030226  062702  000002          ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
5619  030232  000765                  BR      1$
5620
5621  030234  013703  001510  3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
5622  030240  000303                  SWAB    R3
5623  030242  050103                  BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
5624  030244  022203                  CMP     (R2)+,R3        ;SEE IF SECTOR/HEAD COMPARE
5625                                                          ;& INCR PTR TO NEXT CYL WORD
5626  030246  001401                  BEQ     4$              ;BR IF COMPARE
5627  030250  000756                  BR      1$              ;ELSE TRY NEXT CYL
5628
5629  030252  012603          4$:     MOV     (SP)+,R3        ;RESTORE R3
5630  030254  062716  000004          ADD     #4,(SP)         ;INCREMENT RET ADDR
5631  030260  000207                  RTS     PC
5632
5633                          ;
5634                          ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
5635                          ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
5636                          ;
5637  030262  010046          SORT:   MOV     R0,-(SP)        ;SAVE R0
5638  030264  010146                  MOV     R1,-(SP)        ;SAVE R1
5639  030266  004737  027052          JSR     PC,RDSEC
5640  030272  062737  000001  001422  ADD     #1,SECTOR
5641  030300  004737  030370          JSR     PC,MULT6        ;MULT SECTOR BY 6
5642
5643  030304  012700  000204          MOV     #132.,R0
5644  030310  163700  001422          SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
5645  030314  010037  001422          MOV     R0,SECTOR
5646  030320  062737  001760  001422  ADD     #RHTAB,SECTOR   ;SAVE INDEX
5647
5648  030326  062700  001760          ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
5649  030332  012701  002164          MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
5650
```

```
5651
5652  030336  012021                  1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
5653  030340  020027  002164                  CMP     R0,#RHTAB+132.
5654  030344  001374                          BNE     1$
5655
5656  030346  012700  001760                  MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
5657  030352  012021                  2$:     MOV     (R0)+,(R1)+
5658  030354  020037  001422                  CMP     R0,SECTOR
5659  030360  001374                          BNE     2$
5660
5661  030362  012601                          MOV     (SP)+,R1        ;RESTOR R1
5662  030364  012600                          MOV     (SP)+,R0        ;RESTOR R0
5663  030366  000207                          RTS     PC
5664
5665                                  ;
5666                                  ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
5667                                  ;
5668  030370  006337  001422          MULT6:  ASL     SECTOR          ;2 X SECTOR
5669  030374  013746  001422                  MOV     SECTOR,-(SP)
5670  030400  006337  001422                  ASL     SECTOR          ;4 X SECTOR
5671  030404  062637  001422                  ADD     (SP)+,SECTOR    ;(4 X 5)+(2 X 5) = 6 X SECTOR
5672  030410  000207                          RTS     PC
5673
5674
5675                                  ;
5676                                  ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
5677                                  ;CMD.  IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
5678                                  ;CYLINDER AT CYL 410, TRACK 2.
5679                                  ;
5680                                  ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
5681                                  ;RETURN+2 IF LISTED, SKIP OVER ERROR
5682  030412  010446                  ;
                                TRUERR: MOV     R4,-(SP)                ;SAVE R4
5683
5684  030414  032737  010000  005406          BIT     #CFMT,HCS1      ;CHECK FORMAT
5685  030422  001014                          BNE     3$              ;BR FOR 20 SECTOR FORMAT
5686                                                                  ;NOTE, 20 SECTOR FORMAT NOT
5687                                                                  ;DONE IN THIS PROGRAM
5688
5689  030424  012704  002400                  MOV     #BSE22H+8.,R4
5690  030430  004737  030464                  JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
5691  030434  000407                          BR      3$              ;RETURN HERE IF YES
5692
5693  030436  012704  003400                  MOV     #BSE22S+8.,R4   ;ELSE RETURN HERE
5694  030442  004737  030464                  JSR     PC,TERR1        ;& SEE IF ON SOFTWARE DETECTED TABLE
5695  030446  000402                          BR      3$              ;RETURN HERE IF YES
5696
5697  030450  012604                  1$:     MOV     (SP)+,R4        ;RESTORE R4
5698  030452  000207                          RTS     PC              ;RETURN WITHOUT JUMPING OVER ERROR
5699
5700
5701  030454  012604                  3$:     MOV     (SP)+,R4        ;RESTORE R4
5702  030456  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR ON RETURN
5703  030462  000207                          RTS     PC
5704
5705                                  ;
5706                                  ;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
```

```
5707                                    ;THE BSE TABLE FOR THE ABOVE SUBROUTINE.
5708                                    ;RETURN IF FOUND ON TABLE
5709                                    ;RETURN+2 IF NOT FOUND
5710                                    ;
5711  030464  021427  177777    TERR1:  CMP     (R4),#-1        ;SEE IF ALL 1'S
5712  030470  001405                    BEQ     1$              ;BR IF YES, NOT ON TABLE
5713  030472  022437  001370            CMP     (R4)+,CCYL      ;SEE IF CYL MATCH
5714  030476  001405                    BEQ     2$              ;BR IF YES
5715  030500  005724                    TST     (R4)+           ;ELSE ADV TO NEXT CYL WORD
5716  030502  000770                    BR      TERR1           ;& TRY AGAIN.
5717
5718  030504  062716  000002    1$:     ADD     #2,(SP)
5719  030510  000207                    RTS     PC
5720
5721  030512  022437  005416    2$:     CMP     (R4)+,HDA       ;SEE IF SECTOR & TRACK MATCH
5722  030516  001401                    BEQ     3$              ;BR IF YES
5723  030520  000761                    BR      TERR1           ;OR TRY AGAIN
5724
5725  030522  000207    3$:     RTS     PC
5726
5727
5728
5729
5730                                    ;
5731                                    ;
5732                                    ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
5733                                    ;
5734  030524  005037  001412    CLKON:  CLR     TIMUP
5735  030530  005737  005552            TST     PCLKF
5736  030534  001004                    BNE     1$              ;BRANCH IF P-CLOCK PRESENT
5737  030536  012777  000100  150602    MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
5738  030544  000207                    RTS     PC
5739  030546  012777  177777  150566 1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
5740  030554  012777  000135  150556    MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
5741  030562  000207                    RTS     PC              ;LINE FREQ & RUN
5742                                    ;
5743                                    ;KW11-L & KW11-P INTERRUPT HANDLER
5744                                    ;
5745  030564  005037  001412    CLOCK:  CLR     TIMUP
5746  030570  005337  001406            DEC     COUNT
5747  030574  001010                    BNE     1$
5748  030576  013737  001404  001406    MOV     HZ,COUNT
5749  030604  005337  001410            DEC     SEC
5750  030610  001002                    BNE     1$
5751  030612  005237  001412            INC     TIMUP           ;SORRY, TIME IS UP
5752  030616  000002    1$:     RTI
5753                                    ;
5754                                    ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
5755                                    ;
5756  030620  005737  005552    CLKOF:  TST     PCLKF
5757  030624  001003                    BNE     1$              ;BRACH IF P-CLOCK PRESENT
5758  030626  005077  150514            CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
5759  030632  000207                    RTS     PC
5760  030634  005077  150500    1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
5761  030640  000207                    RTS     PC
5762
```

```
                                        ;
                                        ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MESSAGES
                                        ;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
                                        ;
030642  010046                  $BPAR:  MOV     R0,-(SP)        ;SAVE R0
030644  010146                          MOV     R1,-(SP)        ;SAVE R1
030646  012700  000021                  MOV     #17.,R0         ;SHIFT COUNTER
030652  005001                          CLR     R1              ;COUNT # OF 1'S IN TEMP1
030654  000241                          CLC                     ;CLEAR CARRY
                                
030656  006137  005444          1$:     ROL     TEMP1
030662  103001                          BCC     2$              ;BR IF CARRY CLEAR
030664  005201                          INC     R1              ;COUNT # OF 1'S
030666  005300                  2$:     DEC     R0              ;SHIFT COUNTER
030670  001372                          BNE     1$
                                
030672  032701  000001                  BIT     #BIT0,R1
030676  001003                          BNE     3$              ;BR IF ODD # IN R0
030700  052737  100000  005444          BIS     #M.PAR,TEMP1    ;SET PARITY BIT
030706  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
030710  012600                          MOV     (SP)+,R0        ;RESTORE R0
030712  000207                          RTS     PC
                                
                                        ;
                                        ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                        ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                        ;  IE: MY LOOP MACRO
                                        ;
030714  032777  001000  150216  $SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
030722  001406                          BEQ     1$              ;BR IF NO
030724  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
030730  001403                          BEQ     1$              ;BR IF NO
030732  013716  001110                  MOV     $LPERR,(SP)
030736  000002                          RTI
                                
030740  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
030744  000002                          RTI
                                
                                
                                        ;
                                        ;ROUTINE TO INPUT A 'SPACE' OR 'CONTROL-E' FROM TTY
                                        ;RETURN IF CONTROL-E
                                        ;RETURN +4 IF SPACE
                                        ;
030746  005777  150174          CCSP:   TST     @$TKB           ;CLEAR DONE FLAG
030752  104410                          RDCHR                   ;READ CHAR FROM TTY
030754  012600                          MOV     (SP)+,R0        ;GET CHAR OFF STACK
030756  020027  000040                  CMP     R0,#$PBAR       ;SEE IF SPACE
030762  001406                          BEQ     1$              ;BR IF YES.
```

```
5819
5820  030764  020027  000005          CMP     R0,#5           ;SEE IF CONTROL-E
5821  030770  001405                  BEQ     2$              ;BR IF YES
5822  030772  104401  040652          TYPE    ,MSG31          ;"?"
5823  030776  000763                  BR      CCSP            ;TRY AGAIN
5824
5825  031000  062716  000004  1$:     ADD     #4,(SP)
5826  031004  000207          2$:     RTS     PC
5827
5828                                  ;
5829                                  ;ROUTINE TO INPUT A 'SPACE' FROM TTY
5830                                  ;
5831  031006  005777  150134  GETSP:  TST     @$TKB           ;CLEAR DONE FLAG
5832  031012  104410                  RDCHR                   ;READ CHAR OFF TTY
5833  031014  012600                  MOV     (SP)+,R0        ;GET CHAR OFF STACK
5834  031016  020027  000040          CMP     R0,#SPBAR       ;SEE IF SPACE
5835  031022  001403                  BEQ     1$              ;EXIT IF YES
5836  031024  104401  040652          TYPE    ,MSG31          ;?
5837  031030  000766                  BR      GETSP           ;TRY AGAIN
5838  031032  000207          1$:     RTS     PC
5839
5840                                  ;
5841                                  ;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
5842                                  ;IT IS USED TO ALLOW THE OPERATOR TO HALT THE CPU WHILE INSURING
5843                                  ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
5844                                  ;THE CPU.
5845                                  ;
5846  031034  022626          STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT
5847
5848  031036  004737  026772  STOP1:  JSR     PC,SUBCLR
5849  031042  104024                  ERROR   24              ;CERR AFTER
5850
5851  031044  005737  005370          TST     UNLD            ;SEE IF HEADS UNLOADED
5852  031050  001440                  BEQ     3$              ;BR IF NO
5853  031052  005737  000042          TST     42              ;SEE IF MANUAL OR AUTO MODE
5854  031056  001403                  BEQ     1$              ;BR IF MANUAL MODE
5855  031060  104401  045367          TYPE    ,MSG74          ;PGM ABORT PENDING
5856  031064  000402                  BR      2$
5857  031066  104401  045435  1$:     TYPE    ,MSG75          ;HALT PENDING
5858  031072                  2$:
5859
5860  031072  004737  026772          JSR     PC,SUBCLR
5861  031076  104024                  ERROR   24              ;CERR AFTER SCLR
5862
5863  031100  032737  010000  005434  BIT     #D.SPIN,HMR2    ;SEE IF SPINDLE ALREADY ON
5864  031106  001021                  BNE     64$             ;BR IF YES
5865  031110  104401  040524          TYPE    ,MSG29          ;PLEASE WAIT, HEADS BEING LOADED
5866
5867  031114  012765  000011  000000  MOV     #SRTSPL,RKCS1(R5)  ;START SPINDLE CMD
5868  031122  013737  001426  005444  MOV     T10,TEMP1
5869  031130  004737  025062          JSR     PC,FRDY         ;FIND CONTR RDY
5870  031134  104143                  ERROR   143             ;CONTR RDY NOT SET AFTER CMD
5871
5872  031136  013737  001434  005446  MOV     T100,TEMP2
5873  031144  004737  025376          JSR     PC,FATT1        ;FIND ATTN
5874  031150  104144                  ERROR   144             ;NO ATTN AFTER CMD
```

```
5875  031152                          64$:
5876
5877  031152  005737  005372          3$:     TST     BADHDR          ;SEE IF HEADERS VALID
5878  031156  001466                          BEQ     4$              ;BR IF YES
5879  031160  005237  005374                  INC     HPEND
5880  031164  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5881  031172  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
5882  031200  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
5883                                                                  ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5884                                                                  ;IN RKMR2 & RKMR3 RESP.
5885  031206  013737  001426  005444          MOV     T10,TEMP1       ;SETUP TIMEOUT
5886  031214  004737  025062                  JSR     PC,$RDY         ;FIND RDY
5887  031220  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
5888
5889  031222  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5890  031230  004737  026420                  JSR     PC,GSTAT
5891  031234  032737  020000  005434          BIT     #D.RTZ,HMR2
5892  031242  001001                          BNE     65$
5893  031244  104214                          ERROR   214             ;RTZ NOT SET DURING RECAL CMD
5894  031246  013737  001426  005446  65$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
5895  031254  004737  025376                  JSR     PC,$ATT1        ;FIND ATTN
5896  031260  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
5897
5898  031262  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5899  031270  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5900  031276  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5901  031304  013737  001426  005444          MOV     T10,TEMP1       ;SETUP TIMEOUT
5902  031312  004737  025062                  JSR     PC,$RDY         ;FIND RDY
5903  031316  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5904  031320  004737  025344                  JSR     PC,TSTATN       ;TEST FOR ATTN
5905  031324  000401                          BR      66$
5906  031326  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5907  031330                          66$:
5908
5909
5910  031330  000137  031370                  JMP     FORM            ;WRITE VALID FORMATS
5911
5912  031334  005737  000042          4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
5913  031340  001406                          BEQ     5$              ;BR IF MANUAL MODE
5914  031342  104401  045472                  TYPE    ,MSG76          ;PGM ABORTED
5915  031346  005037  024430                  CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
5916  031352  000137  024402                  JMP     $EOP
5917
5918  031356  104401  045514          5$:     TYPE    ,MSG77          ;CPU HALTED
5919  031362  000000                          HALT
5920  031364  000137  010656                  JMP     ST5             ;START OVER IF CONTINUE PRESSED
5921
5922  031370                          FORM:
5923                                  .SBTTL  UNEXPECTED TIMEOUT HANDLER
5924
5925                                  ;
5926                                  ;THIS ROUTINE IS ENTERED IF THERE IS
5927                                  ;       A.  NON EXISTANT MEMORY (NO SSYN)
5928                                  ;       B.  BOUNDRY ERROR
5929                                  ;       C.  STACK OVERFLOW
5930                                  ;
```

```
                                ;
031370  011600          BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
031372  005740                  TST     -(R0)           ;GET PC BEFORE UPDATE
031374  032777  020000  147536  BIT     #SW13,@SWR      ;INHIBIT ERR TYPOUT?
031402  001005                  BNE     1$              ;YES, DON'T TYPE
031404  104401  046044          TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
031410  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                ;;TYPE PC
031412  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
031414     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
031415     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
031416  032777  001000  147514  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
031424  001403                  BEQ     2$              ;NO, BRANCH
031426  022626                  CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
031430  000177  147452          JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
                                ;THAT GAVE BAD TIMEOUT
031434  032777  040000  147476  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
031442  001401                  BEQ     3$              ;NO BRANCH
031444  000002                  RTI                     ;YES
                                ;
031446  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.
                                ;
031450  022626                  CMP     (SP)+,(SP)+     ;RESTORE STACK
031452  000137  024402          JMP     $EOP            ;ABORT TESTS
                                ;
                        .SBTTL  MEMORY CHECK ENABLE TRAP
                                ;
031456  012737  031472  001176  MEMERR: MOV     #1$,$ESCAPE
031464  011637  001354          MOV     (SP),TRAPPC     ;STORE PC
031470  104202                  ERROR   202             ;UNEXP MEM PARITY ERROR
                                ;
031472  005037  001176  1$:     CLR     $ESCAPE
031476  032777  001000  147434  BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
031504  001001                  BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
031506  000002                  RTI                     ;ELSE RETURN
                                ;
031510  012706  001100  2$:     MOV     #STACK,SP       ;INIT STACK
031514  000177  147370          JMP     @$LPERR
                                ;
                        .SBTTL  RK06 INTERRUPT HANDLER
                                ;
031520  000240          INTER:  NOP
031522  000240                  NOP
031524  000240                  NOP
031526  011600                  MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
031530  005740                  TST     -(R0)           ;GET PC BEFORE UPDATE.
031532  104401  037354          TYPE    ,MSG6           ;INT AT PC=
031536  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                ;;TYPE PC
031540  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
031542     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
031543     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
 5987  031544  000000                          HALT
 5988  031546  000240                          NOP
 5989  031550  000240                          NOP
 5990  031552  000002                          RTI
 5991
 5992                                  .SBTTL  POWER DOWN AND UP ROUTINES
 5993
 5994                                  ;POWER DOWN ROUTINE
 5995
 5996  031554  012737  031566  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
 5997  031562  000000                          HALT
 5998  031564  000776                          BR      .-2             ;HANG UP.
 5999
 6000                                  ;POWER UP ROUTINE
 6001
 6002  031566  005037  031640          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
 6003  031572  005237  031640          1$:     INC     $PWRCT          ;WAIT FOR THE INCR
 6004  031576  001375                          BNE     1$              ;OF WORD
 6005  031600  012737  031554  000024          MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
 6006  031606  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;PRIORITY 7
 6007  031614  012737  000340  000036          MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
 6008  031622  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
 6009  031626  104401  037550                  TYPE    ,MSG11          ;REPORT POWER FAIL
 6010  031632  000005                          RESET
 6011  031634  000137  012434                  JMP     PFSRT
 6012
 6013  031640  000000          $PWRCT: 0                               ;WAIT COUNT FOR TTY
 6014
 6015                                  ;
 6016                                  ;DIVISION UTILITY ROUTINE
 6017                                  ;
 6018                                  ;R0-R1-R2-R3=DIVIDEND
 6019                                  ;R4-R5=DIVISOR
 6020                                  ;R0-R1=REMAINDER AFTER DIVISION
 6021                                  ;R2-R3=QUOTIENT AFTER DIVISION
 6022                                  ;ENTER WITH JSR PC,M.DPID
 6023                                  ;
 6024  031642  012746  000040  M.DPID: MOV     #40,-(SP)       ;COUNTER FOR DIVISION CYCLES
 6025  031646  010446                  MOV     R4,-(SP)        ;HI ORDER
 6026  031650  010546                  MOV     R5,-(SP)        ;LO ORDER TO THE STACK
 6027  031652  005466  000002          NEG     2(SP)           ;FORM NEGATIVE
 6028  031656  005416                  NEG     @SP             ;VERSION OF DIVISOR
 6029  031660  005666  000002          SBC     2(SP)
 6030  031664  061601                  ADD     @SP,R1
 6031  031666  005500                  ADC     R0              ;PERFORM INIT SUBT.
 6032  031670  066600  000002          ADD     2(SP),R0
 6033  031674  103445                  BCS     M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
 6034  031676  005046                  CLR     -(SP)           ;THIS IS A LONGER LASTING CARRY BIT
 6035  031700  006103          M.DP40: ROL     R3
 6036  031702  006102                  ROL     R2
 6037  031704  006101                  ROL     R1
 6038  031706  006100                  ROL     R0
 6039  031710  005716                  TST     @SP             ;TEST CARRY INDICATOR
 6040  031712  001410                  BEQ     M.DP41          ;IF TO CARRY THEN ADD, ELSE SUBT.
 6041  031714  005016                  CLR     @SP             ;CLEAR UP FOR NEXT TIME
 6042  031716  066601  000002          ADD     2(SP),R1
```

```
 6043  031722  005500                        ADC     R0              ;ADD -(DIVISOR)
 6044  031724  005516                        ADC     @SP             ;SET CARRY
 6045  031726  066600  000004                ADD     4(SP),R0
 6046  031732  000404                        BR      M.DP42
 6047
 6048  031734  060501              M.DP41:   ADD     R5,R1
 6049  031736  005500                        ADC     R0              ;ADD +(DIVISOR)
 6050  031740  005516                        ADC     @SP             ;SET CARRY
 6051  031742  060400                        ADD     R4,R0
 6052  031744  005516              M.DP42:   ADC     @SP             ;SET CARRY
 6053  031746  005716                        TST     @SP             ;TEST THE UPDATE INDICATOR
 6054  031750  001401                        BEQ     .+4             ;IF 0,FORGET IT
 6055  031752  005203                        INC     R3              ;NO CARRY POSSIBLE HERE
 6056  031754  005366  000006                DEC     6(SP)           ;DECREMENT CTR
 6057  031760  003347                        BGT     M.DP40          ;BR IF MORE TO DO
 6058  031762  006003                        ROR     R3
 6059  031764  103404                        BCS     M.DP44
 6060  031766  060501                        ADD     R5,R1
 6061  031770  005500                        ADC     R0
 6062  031772  060400                        ADD     R4,R0
 6063  031774  000241                        CLC
 6064
 6065  031776  006103              M.DP44:   ROL     R3
 6066  032000  062706  000010                ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
 6067  032004  000242                        CLV
 6068  032006  000207                        RTS     PC
 6069
 6070  032010  062706  000006      M.DP50:   ADD     #6,SP
 6071  032014  000262                        SEV
 6072  032016  000207                        RTS     PC
 6073
```

```
                                .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

032020                          $SCOPE:
032020  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
032022  032777  040000  147110  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
032030  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
032032  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
032034  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
032040  012737  032060  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
032046  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
032052  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
032056  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
032060  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
032062  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
032066  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
032070                          6$:;#####END OF CODE FOR THE XOR TESTER#####
032070  032777  C00400  147042          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
032076  001404                          BEQ     2$              ;;BR IF NO
032100  127737  147034  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
032106  001465                          BEQ     $OVER           ;;BR IF YES
032110  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
032114  001421                          BEQ     3$              ;;BR IF NO
032116  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
032124  101015                          BHI     3$              ;;BR IF NO
032126  032777  001000  147004          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
032134  001404                          BEQ     4$              ;;BR IF NO
032136  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
032144  000446                          BR      $OVER
032146  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
032152  005037  001174                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
032156  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
032160  032777  004000  146752  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
032166  001011                          BNE     1$              ;;BR IF YES
032170  005737  001216                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
032174  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
032176  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
032202  023737  001174  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
032210  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
032212  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
032220  013737  032276  001174          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
032226  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
032232  113737  001102  001214          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
  6130  032240  011637  001106                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
  6131  032244  011637  001110                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
  6132  032250  005037  001176                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
  6133  032254  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
  6134  032262  013777  001102  146652 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
  6135  032270  013716  001106                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
  6136  032274  000002                        RTI                     ;;FIXES PS
  6137  032276  003720                $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
  6138                                .SBTTL  ERROR HANDLER ROUTINE
  6139
  6140                                ;;**********************************************************************
  6141                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
  6142                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
  6143                                ;*AND GO TO TYPERR ON ERROR
  6144                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  6145                                ;*SW15=1         HALT ON ERROR
  6146                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
  6147                                ;*SW10=1         BELL ON ERROR
  6148                                ;*SW09=1         LOOP ON ERROR
  6149                                ;*CALL
  6150                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
  6151
  6152  032300                        $ERROR:
  6153  032300  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
  6154  032302  105237  001103        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
  6155  032306  001775                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
  6156  032310  013777  001102  146624        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
  6157  032316  032777  002000  146614        BIT     #BIT10,@SWR     ;;BELL ON ERROR?
  6158  032324  001402                        BEQ     1$              ;;NO - SKIP
  6159  032326  104401  001200                TYPE    ,$BELL          ;;RING BELL
  6160  032332  005237  001112        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
  6161  032336  011637  001116                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
  6162  032342  162737  000002  001116        SUB     #2,$ERRPC
  6163  032350  117737  146542  001114        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
  6164  032356  032777  020000  146554        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
  6165  032364  001004                        BNE     20$             ;;SKIP TYPEOUTS
  6166  032366  004737  061432                JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
  6167  032372  104401  001205                TYPE    ,$CRLF
  6168  032376                        20$:
  6169  032376  122737  000001  001230        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
  6170  032404  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
  6171  032406  113737  001114  032420        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
  6172  032414  004737  033224                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
  6173  032420     000                21$:    .BYTE   0
  6174  032421     000                        .BYTE   0
  6175  032422  000777                22$:    BR      22$             ;;APT ERROR LOOP
  6176  032424  005777  146510        2$:     TST     @SWR            ;;HALT ON ERROR
  6177  032430  100002                        BPL     3$              ;;SKIP IF CONTINUE
  6178  032432  000000                        HALT                    ;;HALT ON ERROR!
  6179  032434  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
  6180  032436  032777  001000  146474 3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
  6181  032444  001402                        BEQ     4$              ;;BR IF NO
  6182  032446  013716  001110                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
  6183  032452  005737  001176        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
  6184  032456  001402                        BEQ     5$              ;;BR IF NONE
  6185  032460  013716  001176                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
6186  032464
6187  032464  022737  024470  000042  5$:     CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
6188  032472  001001                          BNE     6$              ;;BRANCH IF NO
6189  032474  000000                          HALT                    ;;YES
6190  032476                              6$:
6191  032476  000002                          RTI                     ;;RETURN
6192                                    .SBTTL  TYPE ROUTINE
6193
6194                                    ;;**********************************************************************
6195                                    ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
6196                                    ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
6197                                    ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
6198                                    ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
6199                                    ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
6200                                    ;*
6201                                    ;*CALL:
6202                                    ;*1) USING A TRAP INSTRUCTION
6203                                    ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
6204                                    ;*OR
6205                                    ;*      TYPE
6206                                    ;*      MESADR
6207                                    ;*
6208
6209  032500  105737  001157  $TYPE:   TSTB    $TPFLG          ;;IS THERE A TERMINAL?
6210  032504  100002                          BPL     1$              ;;BR IF YES
6211  032506  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
6212  032510  000430                          BR      3$              ;;LEAVE
6213  032512  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
6214  032514  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
6215  032520  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
6216  032526  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
6217  032530  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
6218  032536  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
6219  032540  010037  032550                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
6220  032544  004737  033214                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
6221  032550  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
6222  032552  132737  000040  001231  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
6223  032560  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
6224  032562  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
6225  032564  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
6226  032566  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
6227  032570  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
6228  032572  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
6229  032576  000002                          RTI                     ;;RETURN
6230  032600  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
6231  032604  001430                          BEQ     8$
6232  032606  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
6233  032612  001006                          BNE     5$
6234  032614  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
6235  032616  104401                          TYPE                    ;;TYPE A CR AND LF
6236  032620  001205                          $CRLF
6237  032622  105037  032756                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
6238  032626  000755                          BR      2$              ;;GET NEXT CHARACTER
6239  032630  004737  032712          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
6240  032634  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
6241  032640  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
6242  032642  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
6243                                                                    ;;AND THE NULL CHAR.
6244  032646  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
6245  032652  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
6246  032654  004737  032712                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
6247  032660  105337  032756                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
6248  032664  000770                          BR      7$              ;;LOOP
6249
6250                                  ;HORIZONTAL TAB PROCESSOR
6251
6252  032666  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
6253  032672  004737  032712          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
6254  032676  132737  000007  032756          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
6255  032704  001372                          BNE     9$              ;;TAB STOP
6256  032706  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
6257  032710  000724                          BR      2$              ;;GET NEXT CHARACTER
6258  032712  105777  146232          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
6259  032716  100375                          BPL     $TYPEC
6260  032720  116677  000002  146224          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6261  032726  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
6262  032734  001003                          BNE     1$              ;;BRANCH IF NO
6263  032736  105037  032756                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
6264  032742  000406                          BR      $TYPEX          ;;EXIT
6265  032744  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
6266  032752  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
6267  032754  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
6268  032756  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
6269  032760  000207          $TYPEX: RTS     PC
6270
6271                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6272
6273                                  ;;**************************************************************
6274                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
6275                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
6276                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
6277                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
6278                                  ;*REPLACED WITH SPACES.
6279                                  ;*CALL:
6280                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
6281                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
6282
6283  032762                          $TYPDS:
6284  032762  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6285  032764  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6286  032766  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6287  032770  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
6288  032772  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
6289  032774  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
6290  033000  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
6291  033004  100004                          BPL     1$              ;;BR IF INPUT IS POS.
6292  033006  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
6293  033010  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
6294  033016  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
6295  033020  012703  033176                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
6296  033024  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
6297  033030  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
```

```
6298   033032  016001  033166          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
6299   033036  160105          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
6300   033040  002402                  BLT     4$              ;;BR IF DONE
6301   033042  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
6302   033044  000774                  BR      3$
6303   033046  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
6304   033050  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
6305   033052  001002                  BNE     5$              ;;FALL THROUGH IF 0
6306   033054  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
6307   033056  100407                  BMI     7$              ;;BR IF YES
6308   033060  106316          5$:     ASLB    (SP)            ;;MSD?
6309   033062  103003                  BCC     6$              ;;BR IF NO
6310   033064  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
6311   033072  052702  000060  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
6312   033076  052702  000040  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
6313   033102  110223                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
6314   033104  005720                  TST     (R0)+           ;;JUST INCREMENTING
6315   033106  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
6316   033112  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
6317   033114  003002                  BGT     8$              ;;GO TO EXIT
6318   033116  010502                  MOV     R5,R2           ;;GET THE LSD
6319   033120  000764                  BR      6$              ;;GO CHANGE TO ASCII
6320   033122  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
6321   033124  100003                  BPL     9$              ;;BR IF NO
6322   033126  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
6323   033134  105013          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
6324   033136  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
6325   033140  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
6326   033142  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
6327   033144  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
6328   033146  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
6329   033150  104401  033176          TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
6330   033154  016666  000002  000004  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
6331   033162  012616                  MOV     (SP)+,(SP)
6332   033164  000002                  RTI                     ;;RETURN TO USER
6333   033166  023420          $DTBL:  10000.
6334   033170  001750                  1000.
6335   033172  000144                  100.
6336   033174  000012                  10.
6337   033176  000004          $DBLK:  .BLKW   4
6338                           .SBTTL  APT COMMUNICATIONS ROUTINE
6339
6340                           ;;**************************************************************
6341   033206  112737  000001  033452  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
6342   033214  112737  000001  033450  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
6343   033222  000403                  BR      $ATYC
6344   033224  112737  000001  033452  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
6345   033232                          $ATYC:
6346   033232  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6347   033234  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6348   033236  105737  033450          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
6349   033242  001450                  BEQ     5$              ;;IF NOT:  BR
6350   033244  122737  000001  001230  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
6351   033252  001031                  BNE     3$              ;;IF NOT:  BR
6352   033254  132737  000100  001231  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
6353   033262  001425                  BEQ     3$              ;;IF NOT:  BR
```

```
 6354  033264  017600  000004                 MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 6355  033270  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 6356  033276  005737  001210         1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 6357  033302  001375                 BNE     1$              ;;IF NOT:  WAIT
 6358  033304  010037  001224                 MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 6359  033310  105720                 2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 6360  033312  001376                 BNE     2$
 6361  033314  163700  001224                 SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 6362  033320  006200                 ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 6363  033322  010037  001226                 MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 6364  033326  012737  000004  001210         MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 6365  033334  000413                 BR      5$
 6366  033336  017637  000004  033362 3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 6367  033344  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 6368  033352  013746  177776                 MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 6369  033356  004737  032500                 JSR     PC,$TYPE        ;;CALL TYPE MACRO
 6370  033362  000000                 4$:     .WORD   0
 6371  033364                         5$:
 6372  033364  105737  033452         10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 6373  033370  001416                 BEQ     12$             ;;IF NOT:  BR
 6374  033372  005737  001230                 TST     $ENV            ;;RUNNING UNDER APT?
 6375  033376  001413                 BEQ     12$             ;;IF NOT:  BR
 6376  033400  005737  001210         11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 6377  033404  001375                 BNE     11$             ;;IF NOT:  WAIT
 6378  033406  017637  000004  001212         MOV     @4(SP),$FATAL   ;;GET ERROR #
 6379  033414  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 6380  033422  005237  001210                 INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 6381  033426  105037  033452         12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 6382  033432  105037  033451                 CLRB    $LFLG           ;;CLEAR LOG FLAG
 6383  033436  105037  033450                 CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 6384  033442  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 6385  033444  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 6386  033446  000207                         RTS     PC              ;;RETURN
 6387  033450     000         $MFLG:  .BYTE   0               ;;MESSG. FLAG
 6388  033451     000         $LFLG:  .BYTE   0               ;;LOG FLAG
 6389  033452     000         $FFLG:  .BYTE   0               ;;FATAL FLAG
 6390          033454                 .EVEN
 6391          000200         APTSIZE=200
 6392          000001         APTENV=001
 6393          000100         APTSPOOL=100
 6394          000040         APTCSUP=040
 6395                         .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 6396
 6397                         ;;**************************************************************
 6398                         ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 6399                         ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 6400                         ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 6401                         ;*CALL:
 6402                         ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 6403                         ;*      TYPOS                   ;;CALL FOR TYPEOUT
 6404                         ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 6405                         ;*      .BYTE   M               ;;M=1 OR 0
 6406                         ;*                                      ;;1=TYPE LEADING ZEROS
 6407                         ;*                                      ;;0=SUPPRESS LEADING ZEROS
 6408                         ;*
 6409                         ;*$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

033454  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
033460  116637  000001  033677          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
033466  112637  033701                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
033472  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
033476  000406                          BR      $TYPON
033500  112737  000001  033677  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
033506  112737  000006  033701          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
033514  112737  000005  033676  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
033522  010346                          MOV     R3,-(SP)        ;;SAVE R3
033524  010446                          MOV     R4,-(SP)        ;;SAVE R4
033526  010546                          MOV     R5,-(SP)        ;;SAVE R5
033530  113704  033701                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
033534  005404                          NEG     R4
033536  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
033542  110437  033700                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
033546  113704  033677                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
033552  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
033556  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
033560  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
033562  000404                          BR      3$              ;;GO DO MSB
033564  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
033566  006105                          ROL     R5
033570  006105                          ROL     R5
033572  010503                          MOV     R5,R3
033574  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
033576  105337  033700                  DECB    $OMODE          ;;TYPE THIS DIGIT?
033602  100016                          BPL     7$              ;;BR IF NO
033604  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
033610  001002                          BNE     4$              ;;TEST FOR 0
033612  005704                          TST     R4              ;;SUPPRESS THIS 0?
033614  001403                          BEQ     5$              ;;BR IF YES
033616  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
033620  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
033624  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
033630  110337  033674                  MOVB    R3,8$           ;;SAVE FOR TYPING
033634  104401  033674                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
033640  105337  033676          7$:     DECB    $OCNT           ;;COUNT BY 1
033644  003347                          BGT     2$              ;;BR IF MORE TO DO
033646  002402                          BLT     6$              ;;BR IF DONE
033650  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
033652  000744                          BR      2$              ;;GO DO THE LAST DIGIT
033654  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
033656  012604                          MOV     (SP)+,R4        ;;RESTORE R4
033660  012603                          MOV     (SP)+,R3        ;;RESTORE R3
033662  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
033670  012616                          MOV     (SP)+,(SP)
```

```
                                        RTI                             ;;RETURN
                                8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
                                        .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
                                $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
                                $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
                                $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  TTY INPUT ROUTINE

                                ;;******************************************************************
                                .ENABL  LSB
                                $TKCNT: .WORD   0                       ;;NUMBER OF ITEMS IN QUEUE
                                $TKQIN: .WORD   0                       ;;INPUT POINTER
                                $TKQOUT: .WORD  0                       ;;OUTPUT POINTER
                                $TKQSRT: .BLKB  1                       ;;TTY KEYBOARD QUEUE
                                $TKQEND=.
                                .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
                                ;
033712  005037  033702          $TKINT: CLR     $TKCNT                  ;;CLEAR COUNT OF ITEMS IN QUEUE
033716  012737  033710  033704          MOV     #$TKQSRT,$TKQIN         ;;MOVE THE STARTING ADDRESS OF THE
033724  013737  033704  033706          MOV     $TKQIN,@#TKQOUT         ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
033732  012737  033762  000060          MOV     #$TKSRV,@#TKVEC         ;;INITIALIZE THE KEYBOARD VECTOR
033740  012737  000200  000062          MOV     #200,@#TKVEC+2          ;;"BR" LEVEL 4
033746  005777  145174                  TST     @$TKB                   ;;CLEAR DONE FLAG
033752  012777  000100  145164          MOV     #100,@$TKS              ;;ENABLE TTY KEYBOARD INTERRUPT
033760  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)
                                ;
033762  117746  145160          $TKSRV: MOVB    @$TKB,-(SP)             ;;PICKUP THE CHARACTER
033766  042716  177600                  BIC     #^C177,(SP)             ;;STRIP THE JUNK
033772  021627  000003                  CMP     (SP),#3                 ;;IS IT A CONTROL C?
033776  001007                          BNE     1$                      ;;BRANCH IF NO
034000  104401  035110                  TYPE    ,$CNTLC                 ;;TYPE A CONTROL-C (^C)
034004  004737  033712                  JSR     PC,$TKINT               ;;INIT THE KEYBOARD
034010  005726                          TST     (SP)+                   ;;CLEAN UP STACK
034012  000137  031034                  JMP     STOP                    ;;CONTROL C RESTART
034016  021627  000007          1$:     CMP     (SP),#7                 ;;IS IT A CONTROL G?
034022  001004                          BNE     2$                      ;;BRANCH IF NO
034024  022737  000176  001140          CMP     #SWREG,SWR              ;;IS SOFT-SWR SELECTED?
034032  001500                          BEQ     6$                      ;;GO TO SWR CHANGE

034034                          2$:
034034  022737  000001  033702          CMP     #1,$TKCNT               ;;IS THE QUEUE FULL?
```

```
6522  034042  001004                        BNE     3$              ;;BRANCH IF NO
6523  034044  104401  001200                TYPE    ,$BELL          ;;RING THE TTY BELL
6524  034050  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
6525  034052  000451                        BR      5$              ;;EXIT
6526  034054  021627  000023        3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
6527  034060  001021                        BNE     32$             ;;BRANCH IF NO
6528  034062  005077  145056                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
6529  034066  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
6530  034070  105777  145050        31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
6531  034074  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
6532  034076  117746  145044                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
6533  034102  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
6534  034106  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
6535  034112  001366                        BNE     31$             ;;BRANCH IF NO
6536  034114  012777  000100  145022        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
6537  034122  000002                        RTI                     ;;RETURN
6538  034124  005237  033702        32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
6539  034130  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
6540  034134  002405                        BLT     4$              ;;BRANCH IF YES
6541  034136  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
6542  034142  003002                        BGT     4$              ;;BRANCH IF YES
6543  034144  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
6544  034150  112677  177530        4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
6545  034154  005237  033704                INC     $TKQIN          ;;UPDATE THE POINTER
6546  034160  023727  033704  033711        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
6547  034166  001003                        BNE     5$              ;;BRANCH IF NO
6548  034170  012737  033710  033704        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
6549  034176  000002                5$:     RTI                     ;;RETURN
6550
6551                                        ;;**************************************************************
6552                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
6553                                        ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
6554                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
6555                                        ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
6556  034200  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
6557  034206  001124                        BNE     15$             ;;EXIT IF NOT
6558  034210  105777  144730                TSTB    @$TKS           ;;IS A CHAR WAITING?
6559  034214  100121                        BPL     15$             ;;IF NOT, EXIT
6560  034216  117746  144724                MOVB    @$TKB,-(SP)     ;;YES
6561  034222  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
6562  034226  021627  000007                CMP     (SP),#7         ;;IS IT A CONTROL-G?
6563  034232  001300                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
6564                                                                ;;AND EXIT
6565
6566                                        ;;**************************************************************
6567                                        ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
6568                                        ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
6569                                        ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
6570  034234  123727  001134  000001 6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
6571  034242  001674                        BEQ     2$              ;;BRANCH IF YES
6572  034244  005726                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
6573  034246  004737  033712                JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
6574  034252  005077  144666                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
6575  034256  112737  000001  001135        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
6576
6577  034264  104401  035122                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
```

```
 6578  034270  104401  035127          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 6579  034274  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 6580  034300  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 6581  034302  104401  035140                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 6582  034306  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 6583  034310  005046                          CLR     -(SP)           ;;THE NEW SWR
 6584  034312  105777  144626          7$:     TSTB    @$TKS           ;;CHAR THERE?
 6585  034316  100375                          BPL     7$              ;;IF NOT TRY AGAIN
 6586
 6587  034320  117746  144622                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 6588  034324  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 6589
 6590  034330  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
 6591  034334  001015                          BNE     9$              ;;BRANCH IF NOT
 6592  034336  104401  035110                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
 6593  034342  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
 6594  034346  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
 6595  034354  001003                          BNE     8$              ;;BRANCH IF NO
 6596  034356  012777  000100  144560          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
 6597  034364  000137  031034          8$:     JMP     STOP            ;;CONTROL-C RESTART
 6598
 6599
 6600  034370  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 6601  034374  001005                          BNE     10$             ;;BRANCH IF NOT
 6602  034376  104401  035115                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 6603  034402  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 6604  034406  000737                          BR      19$             ;;LET'S TRY IT AGAIN
 6605
 6606
 6607  034410  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
 6608  034414  001022                          BNE     16$             ;;BRANCH IF NO
 6609  034416  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 6610  034422  001403                          BEQ     11$             ;;BRANCH IF YES
 6611  034424  016677  000002  144506          MOV     2(SP),@SWR      ;;SAVE NEW SWR
 6612  034432  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
 6613  034436  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 6614  034442  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 6615  034450  001003                          BNE     15$             ;;BRANCH IF NOT
 6616  034452  012777  000100  144464          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 6617  034460  000002          15$:            RTI                     ;;RETURN
 6618  034462  004737  032712          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 6619  034466  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
 6620  034472  002420                          BLT     18$             ;;BRANCH IF YES
 6621  034474  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
 6622  034500  003015                          BGT     18$             ;;BRANCH IF YES
 6623  034502  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 6624  034506  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
 6625  034512  001403                          BEQ     17$             ;;BRANCH IF YES
 6626  034514  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
 6627  034516  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
 6628  034520  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
 6629  034522  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 6630  034526  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 6631  034532  000667                          BR      7$              ;;GET THE NEXT ONE
 6632  034534  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 6633  034540  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
                                ;
034542  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
034544  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
034552  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
034556  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
034560  012746  034566                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
034564  000002                          RTI                     ;;POP NEW PC AND PS
034566                          64$:
034566  005737  033702          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
034572  001775                          BEQ     1$
034574  005337  033702                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
034600  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
034606  005237  033706                  INC     $TKQOUT         ;;UPDATE THE POINTER
034612  023727  033706  033711          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
034620  001003                          BNE     2$              ;;BRANCH IF NO
034622  012737  033710  033706          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
034630  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

034632  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
034634  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
034636  012703  035066          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
034642  022703  035110          2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
034646  101456                          BLOS    4$              ;;BR IF YES
034650  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
034652  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
034654  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
034660  001022                          BNE     5$              ;;BR IF NO
034662  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
034664  001007                          BNE     6$              ;;BR IF NO
034666  112737  000134  035064          MOVB    #'\,9$          ;;TYPE A BACK SLASH
034674  104401  035064                  TYPE    ,9$
034700  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
034704  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
034706  020327  035066                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
034712  103434                          BLO     4$              ;;BR IF YES
034714  111337  035064                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
034720  104401  035064                  TYPE    ,9$             ;;GO TYPE
034724  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
034726  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
034730  001406                          BEQ     7$              ;;BR IF NO
```

```
 6690  034732  112737  000134  035064          MOVB    #'\,9$          ;;TYPE A BACK SLASH
 6691  034740  104401  035064                  TYPE    ,9$
 6692  034744  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
 6693  034746  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
 6694  034752  001003                          BNE     8$              ;;BR IF NO
 6695  034754  104401  035115                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
 6696  034760  000726                          BR      1$              ;;GO START OVER
 6697  034762  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
 6698  034766  001011                          BNE     3$              ;;BRANCH IF NO
 6699  034770  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
 6700  034772  104401  001205                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
 6701  034776  104401  035066                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
 6702  035002  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
 6703  035004  104401  001204          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 6704  035010  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 6705  035012  111337  035064          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 6706  035016  104401  035064                  TYPE    ,9$
 6707  035022  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 6708  035026  001305                          BNE     2$              ;;LOOP IF NOT RETURN
 6709  035030  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 6710  035034  104401  001206                  TYPE    ,$LF            ;;TYPE A LINE FEED
 6711  035040  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 6712  035042  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 6713  035044  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 6714  035046  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 6715  035054  012766  035066  000004          MOV     #$TTYIN,4(SP)
 6716  035062  000002                          RTI                     ;;RETURN
 6717  035064     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 6718  035065     000                          .BYTE   0               ;;TERMINATOR
 6719  035066  000022                  $TTYIN: .BLKB   22              ;;RESERVE 22 BYTES FOR TTY INPUT
 6720  035110  041536  005015     000  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
 6721  035115     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 6722  035122  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 6723  035127     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
 6724  035134  036440  000040
 6725  035140  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
 6726  035146  020075     000
 6727          035152                  .EVEN
 6728                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 6729
 6730                                  ;;*****************************************************************
 6731                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 6732                                  ;*CHANGE IT TO BINARY.
 6733                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
 6734                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
 6735                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
 6736                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
 6737                                  ;*CALL:
 6738                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 6739                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 6740                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 6741
 6742  035152  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 6743  035154  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
 6744  035162  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6745  035164  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
6746  035166  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6747  035170  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
6748  035172  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
6749  035174  010037  035300          MOV     R0,5$           ;;AND SAVE IT
6750  035200  005001                  CLR     R1              ;;CLEAR DATA WORD
6751  035202  005002                  CLR     R2
6752  035204  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
6753  035206  001420                  BEQ     3$              ;;IF ZERO GET OUT
6754  035210  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
6755  035214  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
6756  035216  122716  000067          CMPB    #'7,(SP)
6757  035222  002423                  BLT     4$
6758  035224  006301                  ASL     R1              ;;*2
6759  035226  006102                  ROL     R2
6760  035230  006301                  ASL     R1              ;;*4
6761  035232  006102                  ROL     R2
6762  035234  006301                  ASL     R1              ;;*8
6763  035236  006102                  ROL     R2
6764  035240  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
6765  035244  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
6766  035246  000756                  BR      2$              ;;LOOP
6767  035250  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
6768  035252  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
6769  035256  010237  035310          MOV     R2,$HIOCT
6770  035262  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
6771  035264  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
6772  035266  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
6773  035270  000002                  RTI                     ;;RETURN
6774  035272  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
6775  035274  105010                  CLRB    (R0)            ;;SET A TERMINATOR
6776  035276  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
6777  035300  000000          5$:     .WORD   0
6778  035302  104401  001204          TYPE    ,$QUES          ;;"?" "CR" & "LF"
6779  035306  000730                  BR      1$              ;;TRY AGAIN
6780  035310  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
6781                          .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
6782
6783                          ;;*****************************************************************
6784                          ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
6785                          ;*UNSIGNED OCTAL ASCIZ NUMBER.
6786                          ;*CALL
6787                          ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
6788                          ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
6789                          ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
6790
6791
6792  035312  104413          $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
6793  035314  016601  000002          MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
6794  035320  012705  035431          MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
6795  035324  012704  000014          MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
6796  035330  012703  177770          MOV     #^C7,R3         ;;MASK
6797  035334  012100                  MOV     (R1)+,R0        ;;LOWER WORD
6798  035336  012101                  MOV     (R1)+,R1        ;;HIGH WORD
6799  035340  005002                  CLR     R2              ;;TERMINATOR
6800  035342  110245          1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
6801  035344  010002                  MOV     R0,R2           ;;GET THIS DIGIT
```

```
6802  035346  005304                        DEC     R4              ;;COUNT THIS CHARACTER
6803  035350  003007                        BGT     3$              ;;BR IF NOT THE LAST DIGIT
6804  035352  001405                        BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
6805  035354  005205                        INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
6806  035356  010566  000002                MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
6807  035362  104414                        RESREG                  ;;RESTORE ALL REGISTERS
6808  035364  000207                        RTS     PC              ;;RETURN TO USER
6809  035366  006203                2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
6810  035370  006001                3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
6811  035372  006000                        ROR     R0              ;;      THE NEXT OCTAL DIGIT
6812  035374  006001                        ROR     R1
6813  035376  006000                        ROR     R0
6814  035400  006001                        ROR     R1
6815  035402  006000                        ROR     R0
6816  035404  040302                        BIC     R3,R2           ;;MASK OUT ALL JUNK
6817  035406  062702  000060                ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
6818  035412  000753                        BR      1$              ;;GO PUT IT IN THE DATA TABLE
6819  035414  000016                $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
6820                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
6821
6822                                  ;;**************************************************************
6823                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
6824                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
6825                                  ;*POSITIVE.
6826                                  ;*CALL
6827                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
6828                                  ;*      JSR     PC,@#$DB2D
6829                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
6830                                                                  ;;IS ON THE STACK
6831
6832
6833  035432  104413                $DB2D:  SAVREG                  ;;SAVE REGISTERS
6834  035434  016602  000002                MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
6835  035440  012700  035612                MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
6836  035444  010066  000002                MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
6837  035450  012201                        MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
6838  035452  012202                        MOV     (R2)+,R2
6839  035454  012737  000012  035530        MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
6840  035462  012704  035542                MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
6841  035466  012705  035544                MOV     #$TNPWR+2,R5
6842  035472  005003                1$:     CLR     R3              ;;CLEAR PARTIAL
6843  035474  161401                2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
6844  035476  005602                        SBC     R2
6845  035500  161502                        SUB     (R5),R2
6846  035502  002402                        BLT     3$              ;;BR IF TEN POWER TO LARGE
6847  035504  005203                        INC     R3              ;;ADD 1 TO PARTIAL
6848  035506  000772                        BR      2$              ;;LOOP
6849  035510  062401                3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
6850  035512  005502                        ADC     R2
6851  035514  062402                        ADD     (R4)+,R2
6852  035516  022525                        CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
6853  035520  052703  000060                BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
6854  035524  110320                        MOVB    R3,(R0)+        ;;SAVE IT
6855  035526  005327                        DEC     (PC)+           ;;DONE?
6856  035530  000000                4$:     .WORD   0
6857  035532  001357                        BNE     1$              ;;BR IF NO
```

```
 6858  035534  105020                    CLRB    (R0)+           ;;TERMINATOR
 6859  035536  104414                    RESREG                  ;;RESTORE REGISTERS
 6860  035540  000207                    RTS     PC              ;;RETURN
 6861  035542  145000            $TNPWR: 145000                  ;;1.0E09
 6862  035544  035632                    35632
 6863  035546  160400                    160400                  ;;1.0E08
 6864  035550  002765                    2765
 6865  035552  113200                    113200                  ;;1.0E07
 6866  035554  000230                    230
 6867  035556  041100                    041100                  ;;1.0E06
 6868  035560  000017                    17
 6869  035562  103240                    103240                  ;;1.0E05
 6870  035564  000001                    1
 6871  035566  023420                    23420                   ;;1.0E04
 6872  035570  000000                    0
 6873  035572  001750                    1750                    ;;1.0E03
 6874  035574  000000                    0
 6875  035576  000144                    144                     ;;1.0E02
 6876  035600  000000                    0
 6877  035602  000012                    12                      ;;1.0E01
 6878  035604  000000                    0
 6879  035606  000001                    1                       ;;1.0E00
 6880  035610  000000                    0
 6881  035612  000014            $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
 6882                            .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
 6883
 6884                            ;;**************************************************************
 6885                            ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
 6886                            ;*UNSIGNED DECIMAL ASCIZ NUMBER.
 6887                            ;*CALL
 6888                            ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
 6889                            ;*      JSR     PC,@#$$B2D      ;;CALL
 6890                            ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
 6891
 6892
 6893  035626  016637  000002  035656  $$B2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
 6894  035634  012746  035656                  MOV     #1$,-(SP)       ;;SET POINTER
 6895  035640  004737  035432                  JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
 6896  035644  062716  000005                  ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
 6897  035650  012666  000002                  MOV     (SP)+,2(SP)     ;;PICKUP POINTER
 6898  035654  000207                          RTS     PC              ;;RETURN
 6899  035656  000000  000000          1$:     .WORD   0,0
 6900                            .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
 6901
 6902                            ;;**************************************************************
 6903                            ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
 6904                            ;*LEADING NUMBERS.
 6905                            ;*CALL
 6906                            ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
 6907                            ;*      JSR     PC,@#$SUPRS
 6908
 6909
 6910  035662  010046            $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
 6911  035664  016600  000004            MOV     4(SP),R0        ;;PICKUP THE POINTER
 6912  035670  105710            1$:     TSTB    (R0)            ;;TERMINATEOR?
 6913  035672  001403                    BEQ     2$              ;;BR IF YES
```

```
 6914  035674  122720  000060          CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
 6915  035700  001773                  BEQ     1$              ;;BR IF YES
 6916  035702  005300                  DEC     R0              ;;BACKUP BY "1"
 6917  035704  010037  035712          MOV     R0,3$           ;;SAVE FOR TYPING
 6918  035710  104401                  TYPE                    ;;GO TYPE
 6919  035712  000000          3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
 6920  035714  012600                  MOV     (SP)+,R0        ;;RESTORE R0
 6921  035716  012616                  MOV     (SP)+,(SP)      ;;RESTORE THE STACK
 6922  035720  000207                  RTS     PC              ;;RETURN
 6923                          .SBTTL  INTEGER MULTIPLY ROUTINE
 6924
 6925                          ;;**************************************************************************
 6926                          ;*CALL
 6927                          ;*      MOV     MULTIPLER,-(SP)
 6928                          ;*      MOV     MULTIPLICAND,-(SP)
 6929                          ;*      JSR     PC,@#$MULT
 6930                          ;*      RETURN  ;;PRODUCT IS ON THE STACK
 6931                          ;*
 6932                          ;*      STACK   PRODUCT
 6933                          ;*      -----   -------
 6934                          ;*      TOP     LSB'S
 6935                          ;*      +2      MSB'S
 6936
 6937  035722                  $MULT:
 6938  035722  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6939  035724  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 6940  035726  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 6941  035730  005046                  CLR     -(SP)           ;;CLEAR THE SIGN KEY
 6942  035732  016601  000012          MOV     12(SP),R1       ;;GET THE MULTIPLICAND
 6943  035736  100002                  BPL     1$              ;;BR IF PLUS
 6944  035740  005216                  INC     (SP)            ;;SET THE SIGN KEY
 6945  035742  005401                  NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
 6946  035744  016602  000014  1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
 6947  035750  100002                  BPL     2$              ;;BR IF PLUS
 6948  035752  005316                  DEC     (SP)            ;;UPDATE THE SIGN KEY
 6949  035754  005402                  NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
 6950  035756  012746  000021  2$:     MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
 6951  035762  005000                  CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
 6952  035764  103001          3$:     BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
 6953  035766  060200                  ADD     R2,R0
 6954  035770  006000          4$:     ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
 6955  035772  006001                  ROR     R1              ;;THE MULTIPLICAND
 6956  035774  005316                  DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
 6957  035776  001372                  BNE     3$              ;;BR IF NO
 6958  036000  022616                  CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
 6959  036002  001403                  BEQ     5$              ;;GO TO EXIT IF NO
 6960  036004  005400                  NEG     R0              ;;YES--SO MAKE IT SO
 6961  036006  005401                  NEG     R1
 6962  036010  005600                  SBC     R0
 6963  036012  005726          5$:     TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
 6964  036014  010066  000012          MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
 6965  036020  010166  000010          MOV     R1,10(SP)       ;;LSB'S
 6966  036024  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 6967  036026  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 6968  036030  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
 6969  036032  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

036034                          $SAVREG:
036034  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
036036  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
036040  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
036042  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
036044  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
036046  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
036050  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
036054  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
036060  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
036064  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
036070  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
036072                          $RESREG:
036072  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
036076  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
036102  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
036106  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
036112  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
036114  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
036116  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
036120  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
036122  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
036124  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
036126  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.

036130  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
036132  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
036136  005740                          TST     -(R0)           ;;BACKUP BY 2
```

```
7026  036140  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
7027  036142  006300                  ASL     R0              ;;POSITION FOR INDEXING
7028  036144  016000  036164          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
7029  036150  000200                  RTS     R0              ;;GO TO ROUTINE
7030
7031
7032                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
7033
7034  036152  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
7035  036154  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
7036  036162  000002                          RTI                     ;;RESTORE THE PSW
7037
7038                                  .SBTTL  TRAP TABLE
7039
7040                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
7041                                  ;*BY THE "TRAP" INSTRUCTION.
7042
7043                                  ;       ROUTINE
7044                                  ;       -------
7045  036164  036152                  $TRPAD: .WORD   $TRAP2
7046  036166  032500                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
7047  036170  033500                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
7048  036172  033454                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
7049  036174  033514                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
7050  036176  032762                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
7051
7052  036200  034270                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
7053
7054  036202  034200                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
7055  036204  034542                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
7056  036206  034632                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
7057  036210  035152                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
7058  036212  036034                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
7059  036214  036072                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
7060  036216  030714                          SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
7061
```

```
                                .SBTTL  SERVICE MESSAGES

036220  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
036226  051525  051040  030113
036234  020066  051104  053111
036242  020105  044504  043501
036250  047516  052123  041511
036256  005015  050011  051101          .ASCII  <CR><LF>/       PART 3/
036264  020124     063
036267     015  046412  044501          .ASCII  <CR><LF>/MAINDEC-11-DZR6J-C-PB/<CR><LF>
036274  042116  041505  030455
036302  026461  055104  033122
036310  026512  026503  041120
036316  005015
036320  005015  025011  025052          .ASCII  <CR><LF>/       ***** CAUTION *****/<CR><LF>
036326  025052  041440  052501
036334  044524  047117  025040
036342  025052  025052  005015
036350  005015  044124  051511          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY BY TYPING CONTROL-C/
036356  050040  047522  051107
036364  046501  051440  047510
036372  046125  020104  041040
036400  020105  040510  052114
036406  042105  047440  046116
036414  020131  054502  052040
036422  050131  047111  020107
036430  047503  052116  047522
036436  026514     103
036441     015  047412  044124          .ASCII  <CR><LF>/OTHERWISE CARTRIDGE FORMATTING AND, OR THE DEVICE/
036446  051105  044527  042523
036454  041440  051101  051124
036462  042111  042507  043040
036470  051117  040515  052124
036476  047111  020107  047101
036504  026104  047440  020122
036512  044124  020105  042504
036520  044526  042503
036524  005015  040515  020131          .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/<CR><LF>
036532  042502  046040  043105
036540  020124  047111  040440
036546  020116  047125  042504
036554  042524  046522  047111
036562  042105  051440  040524
036570  042524  005015
036574  005015  047111  052111          .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
036602  040511  046114  026131
036610  042040  044522  042526
036616  020123  047524  041040
036624  020105  042524  052123
036632  042105  051440  047510
036640  046125  020104  040510
036646  042526  006472     012
036653     015  040412  020056          .ASCII  <CR><LF>/A. HEADS MANUALLY LOADED/
036660  042510  042101  020123
```

```
 7118  036666  040515  052516  046101
 7119  036674  054514  046040  040517
 7120  036702  042504     104
 7121  036705     015  041012  020056          .ASCII  <CR><LF>/B. CORRECT PORT SELECTED/
 7122  036712  047503  051122  041505
 7123  036720  020124  047520  052122
 7124  036726  051440  046105  041505
 7125  036734  042524     104
 7126  036737     015  041412  020056          .ASCII  <CR><LF>/C. WRITE LOCK DISABLED/
 7127  036744  051127  052111  020105
 7128  036752  047514  045503  042040
 7129  036760  051511  041101  042514
 7130  036766     104
 7131  036767     015  042012  020056          .ASCII  <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
 7132  036774  051104  053111  020105
 7133  037002  042522  042101  020131
 7134  037010  047111  044504  040503
 7135  037016  047524  020122  047117
 7136  037024  005015
 7137  037026  005015  051104  053111          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
 7138  037034  051505  047040  052117
 7139  037042  052040  020117  042502
 7140  037050  052040  051505  042524
 7141  037056  020104  052515  052123
 7142  037064  044040  053101     105
 7143  037071     015  041012  052117          .ASCIZ  <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
 7144  037076  020110  047520  052122
 7145  037104  020123  042504  042523
 7146  037112  042514  052103  042105
 7147  037120  005015     000
 7148
 7149  037123     015  041012  020105  MSG2:   .ASCIZ  <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
 7150  037130  052523  042522  052040
 7151  037136  020117  052520  020124
 7152  037144  041523  040522  041524
 7153  037152  020110  040520  045503
 7154  037160  044440  020116  051104
 7155  037166  053111  020105  000060
 7156  037174  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
 7157  037202  024105  024523  052040
 7158  037210  020117  042502  052040
 7159  037216  051505  042524  035104
 7160  037224  000040
 7161  037226  005015  054524  042520  MSG4:   .ASCIZ  <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440:  /
 7162  037234  041040  051525  020123
 7163  037242  042101  051104  051505
 7164  037250  020123  043111  047040
 7165  037256  052117  030440  033467
 7166  037264  032064  035060  020040
 7167  037272     000
 7168  037273     015  052012  050131  MSG5:   .ASCIZ  <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210:  /
 7169  037300  020105  047503  052116
 7170  037306  047522  046114  051105
 7171  037314  044440  052116  051105
 7172  037322  052522  052120  053040
 7173  037330  041505  047524  020122
```

```
 7174  037336  043111  047040  052117
 7175  037344  031040  030061  020072
 7176  037352  000040
 7177  037354  005015  047111  042524  MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
 7178  037362  051122  050125  020124
 7179  037370  041517  052503  051122
 7180  037376  042105  040440  020124
 7181  037404  041520  000075
 7182  037410  005015  051104  053111  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
 7183  037416  020105  020060  044527
 7184  037424  046114  047040  052117
 7185  037432  041040  020105  042524
 7186  037440  052123  042105     000
 7187  037445     015  044412  052116  MSG8:   .ASCIZ  <CR><LF>/INTERLOCKS TEST/<CR><LF>
 7188  037452  051105  047514  045503
 7189  037460  020123  042524  052123
 7190  037466  005015     000
 7191  037471     015  051012  046505  MSG9:   .ASCIZ  <CR><LF>/REMOVE UNIT SELECT PLUG/
 7192  037476  053117  020105  047125
 7193  037504  052111  051440  046105
 7194  037512  041505  020124  046120
 7195  037520  043525     000
 7196  037523     015  005012  044527  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
 7197  037530  046114  052040  051505
 7198  037536  020124  051104  053111
 7199  037544  051505  000072
 7200  037550  005015  050012  053517  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
 7201  037556  051105  052440  020120
 7202  037564  042522  052123  051101
 7203  037572  020124  047524  052040
 7204  037600  051505  020124  006461
 7205  037606  000012
 7206  037610  047516  042516  005015  MSG12:  .ASCIZ  /NONE/<CR><LF>
 7207  037616     000
 7208  037617     015  047012  020117  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
 7209  037624  020114  051117  050040
 7210  037632  041440  047514  045503
 7211  037640  020123  051120  051505
 7212  037646  047105     124
 7213  037651     015  040412  046114          .ASCIZ  <CR><LF>/ALL TIMING TESTS BYPASSED/
 7214  037656  052040  046511  047111
 7215  037664  020107  042524  052123
 7216  037672  020123  054502  040520
 7217  037700  051523  042105     000
 7218  037705     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
 7219  037712  051501  044523  043516
 7220  037720  042040  044522  042526
 7221  037726  000040
 7222  037730  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
 7223  037736  042526  000040
 7224  037742  005015  051104  053111  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL NO. /
 7225  037750  020105  042523  044522
 7226  037756  046101  047040  027117
 7227  037764  000040
 7228  037766  005015  040503  052122  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
 7229  037774  044522  043504  020105
```

```
7230  040002  042523  044522  046101
7231  040010  047040  027117  000040
7232  040016  005015  047125  052111  MSG18:  .ASCIZ  <CR><LF>/UNIT SELECT PLUG TEST/<CR><LF>
7233  040024  051440  046105  041505
7234  040032  020124  046120  043525
7235  040040  052040  051505  006524
7236  040046  000012
7237  040050  005015  047520  052122  MSG19:  .ASCIZ  <CR><LF>/PORT SELECTION TESTS/<CR><LF>
7238  040056  051440  046105  041505
7239  040064  044524  047117  052040
7240  040072  051505  051524  005015
7241  040100      000
7242  040101      015  051012  047125  MSG20:  .ASCIZ  <CR><LF>/RUN-STOP SWITCH TEST/<CR><LF>
7243  040106  051455  047524  020120
7244  040114  053523  052111  044103
7245  040122  052040  051505  006524
7246  040130  000012
7247  040132  005015  041501  046040  MSG21:  .ASCIZ  <CR><LF>/AC LOW DETECTION TEST-PART 1/<CR><LF>
7248  040140  053517  042040  052105
7249  040146  041505  044524  047117
7250  040154  052040  051505  026524
7251  040162  040520  052122  030440
7252  040170  005015      000
7253  040173      015  047012  047117  MSG22:  .ASCIZ  <CR><LF>/NON-EXECUTABLE FUNCTION (NXF) DETECTION TEST/<CR><LF>
7254  040200  042455  042530  052503
7255  040206  040524  046102  020105
7256  040214  052506  041516  044524
7257  040222  047117  024040  054116
7258  040230  024506  042040  052105
7259  040236  041505  044524  047117
7260  040244  052040  051505  006524
7261  040252  000012
7262  040254  005015  047516  041440  MSG23:  .ASCIZ  <CR><LF>/NO CLOCK INTERRUPTS PRESENT, ABORTING TIMING TESTS/
7263  040262  047514  045503  044440
7264  040270  052116  051105  052522
7265  040276  052120  020123  051120
7266  040304  051505  047105  026124
7267  040312  040440  047502  052122
7268  040320  047111  020107  044524
7269  040326  044515  043516  052040
7270  040334  051505  051524      000
7271  040341      015  053412  044522  MSG24:  .ASCIZ  <CR><LF>/WRITE LOCK TEST/<CR><LF>
7272  040346  042524  046040  041517
7273  040354  020113  042524  052123
7274  040362  005015      000
7275  040365      040  051525      000  MSG25:  .ASCIZ  / US/
7276  040371      015  040412  020103  MSG26:  .ASCIZ  <CR><LF>/AC LOW DETECTION TEST-PART 2/<CR><LF>
7277  040376  047514  020127  042504
7278  040404  042524  052103  047511
7279  040412  020116  042524  052123
7280  040420  050055  051101  020124
7281  040426  006462  000012
7282  040432  005015  040412  046114  MSG27:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
7283  040440  042040  044522  042526
7284  040446  020123  042524  052123
7285  040454  042105  005015  000012
```

```
7286  040462  005015  052515  052114  MSG28:  .ASCIZ  <CR><LF>/MULTIPLE DRIVE DETECTION TEST/<CR><LF>
7287  040470  050111  042514  042040
7288  040476  044522  042526  042040
7289  040504  052105  041505  044524
7290  040512  047117  052040  051505
7291  040520  006524  000012
7292  040524  005015  046120  040505  MSG29:  .ASCIZ  <CR><LF>/PLEASE WAIT, HEADS BEING LOADED/<CR><LF>
7293  040532  042523  053440  044501
7294  040540  026124  044040  040505
7295  040546  051504  041040  044505
7296  040554  043516  046040  040517
7297  040562  042504  006504  000012
7298  040570  005015  042526  044522  MSG30:  .ASCIZ  <CR><LF>/VERIFY DOOR CAN NOW BE OPENED & LEAVE IT OPEN/<CR><LF>
7299  040576  054506  042040  047517
7300  040604  020122  040503  020116
7301  040612  047516  020127  042502
7302  040620  047440  042520  042516
7303  040626  020104  020046  042514
7304  040634  053101  020105  052111
7305  040642  047440  042520  006516
7306  040650  000012
7307  040652  005015  006477  000012  MSG31:  .ASCIZ  <CR><LF>/?/<CR><LF>
7308  040660  005015  042522  047515  MSG32:  .ASCIZ  <CR><LF>/REMOVE DISK PACK & CLOSE DOOR/
7309  040666  042526  042040  051511
7310  040674  020113  040520  045503
7311  040702  023040  041440  047514
7312  040710  042523  042040  047517
7313  040716  000122
7314  040720  005015  042504  051120  MSG33:  .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE DOOR OPEN/
7315  040726  051505  020123  051047
7316  040734  047125  051455  047524
7317  040742  023520  051440  044527
7318  040750  041524  020110  047524
7319  040756  023440  052522  023516
7320  040764  053440  044510  042514
7321  040772  042040  047517  020122
7322  041000  050117  047105      000
7323  041005      015  053012  051105  MSG34:  .ASCIZ  <CR><LF>/VERIFY SPINDLE DOES NOT START & HEADS DO NOT LOAD/<CR><LF>
7324  041012  043111  020131  050123
7325  041020  047111  046104  020105
7326  041026  047504  051505  047040
7327  041034  052117  051440  040524
7328  041042  052122  023040  044040
7329  041050  040505  051504  042040
7330  041056  020117  047516  020124
7331  041064  047514  042101  005015
7332  041072      000
7333  041073      015  042012  050105  MSG35:  .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE CARTRIDGE REMOVED/
7334  041100  042522  051523  023440
7335  041106  052522  026516  052123
7336  041114  050117  020047  053523
7337  041122  052111  044103  052040
7338  041130  020117  051047  047125
7339  041136  020047  044127  046111
7340  041144  020105  040503  052122
7341  041152  044522  043504  020105
```

```
7342  041160  042522  047515  042526
7343  041166  000104
7344  041170  005015  047111  042523  MSG36:  .ASCIZ  <CR><LF>/INSERT DISK PACK, DEPRESS 'RUN-STOP' TO 'RUN' & CLOSE DOOR/
7345  041176  052122  042040  051511
7346  041204  020113  040520  045503
7347  041212  020054  042504  051120
7348  041220  051505  020123  051047
7349  041226  047125  051455  047524
7350  041234  023520  052040  020117
7351  041242  051047  047125  020047
7352  041250  020046  046103  051517
7353  041256  020105  047504  051117
7354  041264     000
7355  041265     015  042012  050105  MSG37:  .ASCIZ  <CR><LF>/DEPRESS SPACE BAR WHEN FINISHED/<CR><LF>
7356  041272  042522  051523  051440
7357  041300  040520  042503  041040
7358  041306  051101  053440  042510
7359  041314  020116  044506  044516
7360  041322  044123  042105  005015
7361  041330     000
7362  041331     015  044412  051516  MSG38:  .ASCII  <CR><LF>/INSERT UNIT SELECT PLUGS, IN ANY ORDER/
7363  041336  051105  020124  047125
7364  041344  052111  051440  046105
7365  041352  041505  020124  046120
7366  041360  043525  026123  044440
7367  041366  020116  047101  020131
7368  041374  051117  042504     122
7369  041401     015  052012  042510          .ASCIZ  <CR><LF>/THE PROGRAM WILL ECHO THE UNIT SELECT PLUG NUMBER/<CR><LF>
7370  041406  050040  047522  051107
7371  041414  046501  053440  046111
7372  041422  020114  041505  047510
7373  041430  052040  042510  052440
7374  041436  044516  020124  042523
7375  041444  042514  052103  050040
7376  041452  052514  020107  052516
7377  041460  041115  051105  005015
7378  041466     000
7379  041467     015  042012  050105  MSG39:  .ASCIZ  <CR><LF>/DEPRESS CONTROL-E TO EXIT TEST/<CR><LF>
7380  041474  042522  051523  041440
7381  041502  047117  051124  046117
7382  041510  042455  052040  020117
7383  041516  054105  052111  052040
7384  041524  051505  006524  000012
7385  041532  005015  047526  052514  MSG40:  .ASCII  <CR><LF>/VOLUME VALID NOT SET/
7386  041540  042515  053040  046101
7387  041546  042111  047040  052117
7388  041554  051440  052105
7389  041560  005015  040515  042513          .ASCIZ  <CR><LF>/MAKE SURE ORIGINAL UNIT SELECT PLUG IS INSERTED/<CR><LF>
7390  041566  051440  051125  020105
7391  041574  051117  043511  047111
7392  041602  046101  052440  044516
7393  041610  020124  042523  042514
7394  041616  052103  050040  052514
7395  041624  020107  051511  044440
7396  041632  051516  051105  042524
7397  041640  006504  000012
```

```
7398  041644  005015  042504  042523  MSG41:  .ASCIZ  <CR><LF>/DESELECT PORT IN USE & SELECT OPPOSITE PORT/<CR><LF>
7399  041652  042514  052103  050040
7400  041660  051117  020124  047111
7401  041666  052440  042523  023040
7402  041674  051440  046105  041505
7403  041702  020124  050117  047520
7404  041710  044523  042524  050040
7405  041716  051117  006524  000012
7406  041724  005015  042504  042523  MSG42:  .ASCIZ  <CR><LF>/DESELECT WRONG PORT & SELECT CORRECT PORT/<CR><LF>
7407  041732  042514  052103  053440
7408  041740  047522  043516  050040
7409  041746  051117  020124  020046
7410  041754  042523  042514  052103
7411  041762  041440  051117  042522
7412  041770  052103  050040  051117
7413  041776  006524  000012
7414  042002  005015  042504  042523  MSG43:  .ASCIZ  <CR><LF>/DESELECT BOTH PORTS/<CR><LF>
7415  042010  042514  052103  041040
7416  042016  052117  020110  047520
7417  042024  052122  006523  000012
7418  042032  005015  042523  042514  MSG44:  .ASCIZ  <CR><LF>/SELECT CORRECT PORT/<CR><LF>
7419  042040  052103  041440  051117
7420  042046  042522  052103  050040
7421  042054  051117  006524  000012
7422  042062  005015  047503  051122  MSG45:  .ASCIZ  <CR><LF>/CORRECT PORT NOT SELECTED, TRY AGAIN/<CR><LF>
7423  042070  041505  020124  047520
7424  042076  052122  047040  052117
7425  042104  051440  046105  041505
7426  042112  042524  026104  052040
7427  042120  054522  040440  040507
7428  042126  047111  005015     000
7429  042133     015  042012  020117  MSG46:  .ASCII  <CR><LF>/DO A MANUAL DRIVE LOAD/
7430  042140  020101  040515  052516
7431  042146  046101  042040  044522
7432  042154  042526  046040  040517
7433  042162     104
7434  042163     015  052012  042510          .ASCII  <CR><LF>/THE SPINDLE SHOULD START AND THE 'STOP' INDICATOR SHOULD GO OFF
7435  042170  051440  044520  042116
7436  042176  042514  051440  047510
7437  042204  046125  020104  052123
7438  042212  051101  020124  047101
7439  042220  020104  044124  020105
7440  042226  051447  047524  023520
7441  042234  044440  042116  041511
7442  042242  052101  051117  051440
7443  042250  047510  046125  020104
7444  042256  047507  047440  043106
7445  042264  005015  051047  040505          .ASCIZ  <CR><LF>/'READY' SHOULD GO ON IN APPROX. 1 MIN/<CR><LF>
7446  042272  054504  020047  044123
7447  042300  052517  042114  043440
7448  042306  020117  047117  044440
7449  042314  020116  050101  051120
7450  042322  054117  020056  020061
7451  042330  044515  006516  000012
7452  042336  005015  042526  044522  MSG47:  .ASCIZ  <CR><LF>/VERIFY DOOR CANNOT BE OPENED (DO NOT FORCE)/<CR><LF>
7453  042344  054506  042040  047517
```

```
7454  042352  020122  040503  047116
7455  042360  052117  041040  020105
7456  042366  050117  047105  042105
7457  042374  024040  047504  047040
7458  042402  052117  043040  051117
7459  042410  042503  006451  000012
7460  042416  005015  042504  051120  MSG48:  .ASCII  <CR><LF>/DEPRESS THE 'RUN-STOP' SWITCH TO 'STOP'/
7461  042424  051505  020123  044124
7462  042432  020105  051047  047125
7463  042440  051455  047524  023520
7464  042446  051440  044527  041524
7465  042454  020110  047524  023440
7466  042462  052123  050117     047
7467  042467     015  053012  051105          .ASCII  <CR><LF>/VERIFY THE HEADS UNLOAD, THE 'READY' LIGHT GOES OFF/
7468  042474  043111  020131  044124
7469  042502  020105  042510  042101
7470  042510  020123  047125  047514
7471  042516  042101  020054  044124
7472  042524  020105  051047  040505
7473  042532  054504  020047  044514
7474  042540  044107  020124  047507
7475  042546  051505  047440  043106
7476  042554  005015  020046  044124          .ASCIZ  <CR><LF>/& THE 'STOP' LIGHT GOES ON/<CR><LF>
7477  042562  020105  051447  047524
7478  042570  023520  046040  043511
7479  042576  052110  043440  042517
7480  042604  020123  047117  005015
7481  042612     000
7482  042613     015  052012  051125  MSG49:  .ASCIZ  <CR><LF>/TURN OFF AC POWER FROM BEHIND THE RK06/<CR><LF>
7483  042620  020116  043117  020106
7484  042626  041501  050040  053517
7485  042634  051105  043040  047522
7486  042642  020115  042502  044510
7487  042650  042116  052040  042510
7488  042656  051040  030113  006466
7489  042664  000012
7490  042666  005015  053523  052111  MSG50:  .ASCIZ  <CR><LF>/SWITCH AC POWER BACK ON/
7491  042674  044103  040440  020103
7492  042702  047520  042527  020122
7493  042710  040502  045503  047440
7494  042716  000116
7495  042720  005015  047515  042515  MSG51:  .ASCII  <CR><LF>/MOMENTARILY REMOVE & INSERT THE SAME UNIT SELECT PLUG/
7496  042726  052116  051101  046111
7497  042734  020131  042522  047515
7498  042742  042526  023040  044440
7499  042750  051516  051105  020124
7500  042756  044124  020105  040523
7501  042764  042515  052440  044516
7502  042772  020124  042523  042514
7503  043000  052103  050040  052514
7504  043006     107
7505  043007     015  052012  020117          .ASCIZ  <CR><LF>/TO RESET VOLUME VALID/<CR><LF>
7506  043014  042522  042523  020124
7507  043022  047526  052514  042515
7508  043030  053040  046101  042111
7509  043036  005015     000
```

```
7510  043041  015     042012  050105  MSG52:  .ASCII  <CR><LF>/DEPRESS SPACE TO DO TEST/
7511  043046  042522  051523  051440
7512  043054  040520  042503  052040
7513  043062  020117  047504  052040
7514  043070  051505  124
7515  043073  015     047412  020122          .ASCIZ  <CR><LF>/OR CONTROL-E TO BYPASS ENTIRE TEST/<CR><LF>
7516  043100  047503  052116  047522
7517  043106  026514  020105  047524
7518  043114  041040  050131  051501
7519  043122  020123  047105  044524
7520  043130  042522  052040  051505
7521  043136  006524  000012
7522  043142  005015  044504  040523  MSG53:  .ASCIZ  <CR><LF>/DISABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES OFF/<CR><LF>
7523  043150  046102  020105  044124
7524  043156  020105  051127  052111
7525  043164  020105  047514  045503
7526  043172  051440  044527  041524
7527  043200  026110  053040  051105
7528  043206  043111  020131  044514
7529  043214  044107  020124  047507
7530  043222  051505  047440  043106
7531  043230  005015  000
7532  043233  015     042412  040516  MSG54:  .ASCIZ  <CR><LF>/ENABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES ON/<CR><LF>
7533  043240  046102  020105  044124
7534  043246  020105  051127  052111
7535  043254  020105  047514  045503
7536  043262  051440  044527  041524
7537  043270  026110  053040  051105
7538  043276  043111  020131  044514
7539  043304  044107  020124  047507
7540  043312  051505  047440  006516
7541  043320  000012
7542  043322  005015  054105  052111  MSG55:  .ASCIZ  <CR><LF>/EXIT TEST WITH ORIGINAL UNIT SELECT PLUG NO. /
7543  043330  052040  051505  020124
7544  043336  044527  044124  047440
7545  043344  044522  044507  040516
7546  043352  020114  047125  052111
7547  043360  051440  046105  041505
7548  043366  020124  046120  043525
7549  043374  047040  027117  000040
7550  043402  005015  046101  043511  MSG56:  .ASCII  <CR><LF>/ALIGNMENT CARTRIDGE USED/
7551  043410  046516  047105  020124
7552  043416  040503  052122  044522
7553  043424  043504  020105  051525
7554  043432  042105
7555  043434  005015  051120  043517          .ASCII  <CR><LF>/PROGRAM WILL BYPASS THE WRITE LOCK TEST/
7556  043442  040522  020115  044527
7557  043450  046114  041040  050131
7558  043456  051501  020123  044124
7559  043464  020105  051127  052111
7560  043472  020105  047514  045503
7561  043500  052040  051505  124
7562  043505  015     040412  042116          .ASCIZ  <CR><LF>/AND READ-WRITE DATA PORTION OF AC LOW DETECTION TEST-PART 2/<CR
7563  043512  051040  040505  026504
7564  043520  051127  052111  020105
7565  043526  040504  040524  050040
```

```
 7566  043534  051117  044524  047117
 7567  043542  047440  020106  041501
 7568  043550  046040  053517  042040
 7569  043556  052105  041505  044524
 7570  043564  047117  052040  051505
 7571  043572  026524  040520  052122
 7572  043600  031040  005015  000012
 7573  043606  005015  042526  044522  MSG57:  .ASCIZ  <CR><LF>/VERIFY BATTERY RETRACT FUNCTIONAL/<CR><LF>
 7574  043614  054506  041040  052101
 7575  043622  042524  054522  051040
 7576  043630  052105  040522  052103
 7577  043636  043040  047125  052103
 7578  043644  047511  040516  006514
 7579  043652  000012
 7580  043654  005015  047514  042101  MSG58:  .ASCIZ  <CR><LF>/LOAD HEADS ON ALL DRIVES TO BE TESTED FOR MDS/<CR><LF>
 7581  043662  044040  040505  051504
 7582  043670  047440  020116  046101
 7583  043676  020114  051104  053111
 7584  043704  051505  052040  020117
 7585  043712  042502  052040  051505
 7586  043720  042524  020104  047506
 7587  043726  020122  042116  006523
 7588  043734  000012
 7589  043736  005015  047111  042523  MSG59:  .ASCIZ  <CR><LF>/INSERT SAME UNIT SELECT PLUG NUMBER IN ANY 2 DRIVES TO BE TESTE
 7590  043744  052122  051440  046501
 7591  043752  020105  047125  052111
 7592  043760  051440  046105  041505
 7593  043766  020124  046120  043525
 7594  043774  047040  046525  042502
 7595  044002  020122  047111  040440
 7596  044010  054516  031040  042040
 7597  044016  044522  042526  020123
 7598  044024  047524  041040  020105
 7599  044032  042524  052123  042105
 7600  044040     000
 7601  044041     015  044412  051516  MSG60:  .ASCII  <CR><LF>/INSERT CORRECT UNIT SELECT PLUGS & LOAD HEADS/
 7602  044046  051105  020124  047503
 7603  044054  051122  041505  020124
 7604  044062  047125  052111  051440
 7605  044070  046105  041505  020124
 7606  044076  046120  043525  020123
 7607  044104  020046  047514  042101
 7608  044112  044040  040505  051504
 7609  044120  005015  047117  050040          .ASCIZ  <CR><LF>/ON PREVIOUS 2 DRIVES/<CR><LF>
 7610  044126  042522  044526  052517
 7611  044134  020123  020062  051104
 7612  044142  053111  051505  005015
 7613  044150     000
 7614  044151     015  046412  046125  MSG61:  .ASCIZ  <CR><LF>/MULTIPLE DRIVES FOUND ON DRIVE NO. /
 7615  044156  044524  046120  020105
 7616  044164  051104  053111  051505
 7617  044172  043040  052517  042116
 7618  044200  047440  020116  051104
 7619  044206  053111  020105  047516
 7620  044214  020056     000
 7621  044217     015  041012  050131  MSG62:  .ASCII  <CR><LF>/BYPASSING MULT. DRIVE SELECT TEST/
```

```
7622  044224  051501  044523  043516
7623  044232  046440  046125  027124
7624  044240  042040  044522  042526
7625  044246  051440  046105  041505
7626  044254  020124  042524  052123
7627  044262  005015  047117  054514          .ASCIZ  <CR><LF>/ONLY 1 DRIVE PRESENT/<CR><LF>
7628  044270  030440  042040  044522
7629  044276  042526  050040  042522
7630  044304  042523  052116  005015
7631  044312     000
7632  044313     015  042012  050105  MSG63:  .ASCII  <CR><LF>/DEPRESS SPACE BAR TO DO ANOTHER 2 DRIVES/
7633  044320  042522  051523  051440
7634  044326  040520  042503  041040
7635  044334  051101  052040  020117
7636  044342  047504  040440  047516
7637  044350  044124  051105  031040
7638  044356  042040  044522  042526
7639  044364     123
7640  044365     015  047412  020122          .ASCII  <CR><LF>/OR CONTROL-E TO EXIT TEST/
7641  044372  047503  052116  047522
7642  044400  026514  020105  047524
7643  044406  042440  044530  020124
7644  044414  042524  052123
7645  044420  005015  044450  051516          .ASCIZ  <CR><LF>/(INSERT CORRECT UNIT SELECT PLUGS BEFORE EXITING)/<CR><LF>
7646  044426  051105  020124  047503
7647  044434  051122  041505  020124
7648  044442  047125  052111  051440
7649  044450  046105  041505  020124
7650  044456  046120  043525  020123
7651  044464  042502  047506  042522
7652  044472  042440  044530  044524
7653  044500  043516  006451  000012
7654  044506  005015  044124  020105  MSG64:  .ASCII  <CR><LF>/THE SECOND PORTION OF THIS TEST IS TEMPORARILY BYPASSED/
7655  044514  042523  047503  042116
7656  044522  050040  051117  044524
7657  044530  047117  047440  020106
7658  044536  044124  051511  052040
7659  044544  051505  020124  051511
7660  044552  052040  046505  047520
7661  044560  040522  044522  054514
7662  044566  041040  050131  051501
7663  044574  042523     104
7664  044577     015  051412  042505          .ASCIZ  <CR><LF>/SEE COMMENTS IN THE LISTING/<CR><LF>
7665  044604  041440  046517  042515
7666  044612  052116  020123  047111
7667  044620  052040  042510  046040
7668  044626  051511  044524  043516
7669  044634  005015     000
7670  044637     015  042012  050105  MSG65:  .ASCIZ  <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'STOP'/
7671  044644  042522  051523  023440
7672  044652  052522  026516  052123
7673  044660  050117  020047  053523
7674  044666  052111  044103  052040
7675  044674  020117  051447  047524
7676  044702  023520     000
7677  044705     015  042012  050105  MSG66:  .ASCII  <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN'/
```

```
 7678  044712  042522  051523  023440
 7679  044720  052522  026516  052123
 7680  044726  050117  020047  053523
 7681  044734  052111  044103  052040
 7682  044742  020117  051047  047125
 7683  044750     047
 7684  044751     015  023012  042040          .ASCIZ  <CR><LF>/& DEPRESS SPACE WHEN 'READY' LIGHT GOES ON/<CR><LF>
 7685  044756  050105  042522  051523
 7686  044764  051440  040520  042503
 7687  044772  053440  042510  020116
 7688  045000  051047  040505  054504
 7689  045006  020047  044514  044107
 7690  045014  020124  047507  051505
 7691  045022  047440  006516  000012
 7692  045030  005015  042504  042523  MSG67:  .ASCIZ  <CR><LF>/DESELECT PORT SWITCH ON ALL OTHER DRIVES/<CR><LF>
 7693  045036  042514  052103  050040
 7694  045044  051117  020124  053523
 7695  045052  052111  044103  047440
 7696  045060  020116  046101  020114
 7697  045066  052117  042510  020122
 7698  045074  051104  053111  051505
 7699  045102  005015     000
 7700  045105     015  053012  051105  MSG68:  .ASCIZ  <CR><LF>/VERIFY BOTH DRIVES UNLOADED/<CR><LF>
 7701  045112  043111  020131  047502
 7702  045120  044124  042040  044522
 7703  045126  042526  020123  047125
 7704  045134  047514  042101  042105
 7705  045142  005015     000
 7706  045145     015  042012  050105  MSG69:  .ASCIZ  <CR><LF>/DEPRESS SPACE WHEN 'READY' LIGHT GOES ON/<CR><LF>
 7707  045152  042522  051523  051440
 7708  045160  040520  042503  053440
 7709  045166  042510  020116  051047
 7710  045174  040505  054504  020047
 7711  045202  044514  044107  020124
 7712  045210  047507  051505  047440
 7713  045216  006516  000012
 7714  045222  005015  042526  044522  MSG70:  .ASCIZ <CR><LF>/VERIFY HEADS LOAD/<CR><LF>
 7715  045230  054506  044040  040505
 7716  045236  051504  046040  040517
 7717  045244  006504  000012
 7718  045250  005015  042523  042514  MSG71:  .ASCIZ  <CR><LF>/SELECT CORRECT PORT SWITCH ON ALL OTHER DRIVES/<CR><LF>
 7719  045256  052103  041440  051117
 7720  045264  042522  052103  050040
 7721  045272  051117  020124  053523
 7722  045300  052111  044103  047440
 7723  045306  020116  046101  020114
 7724  045314  052117  042510  020122
 7725  045322  051104  053111  051505
 7726  045330  005015     000
 7727  045333     015  040412  047502  MSG72:  .ASCIZ  <CR><LF>/ABORTING BALANCE OF TESTS/
 7728  045340  052122  047111  020107
 7729  045346  040502  040514  041516
 7730  045354  020105  043117  052040
 7731  045362  051505  051524     000
 7732
 7733  045367     015  050012  047522  MSG74:  .ASCIZ  <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
```

```
7734  045374  051107  046501  040440
7735  045402  047502  052122  050040
7736  045410  047105  044504  043516
7737  045416  027056  050056  042514
7738  045424  051501  020105  040527
7739  045432  052111     000
7740  045435     015  044012  046101  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING...PLEASE WAIT/
7741  045442  020124  042520  042116
7742  045450  047111  027107  027056
7743  045456  046120  040505  042523
7744  045464  053440  044501  000124
7745  045472  005015  051120  043517  MSG76:  .ASCIZ  <CR><LF>/PROGRAM ABORTED/
7746  045500  040522  020115  041101
7747  045506  051117  042524  000104
7748  045514  005015  050103  020125  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
7749  045522  040510  052114  042105
7750  045530     000
7751  045531     015  051412  051117  MSG100: .ASCII  <CR><LF>/SORRY, WRITE LOCK SHOULD NOT BE DEPRESSED/
7752  045536  054522  020054  051127
7753  045544  052111  020105  047514
7754  045552  045503  051440  047510
7755  045560  046125  020104  047516
7756  045566  020124  042502  042040
7757  045574  050105  042522  051523
7758  045602  042105
7759  045604  005015  044127  046111          .ASCII  <CR><LF>/WHILE ON TRACK 2 SECTORS 19, 20 OR 21/
7760  045612  020105  047117  052040
7761  045620  040522  045503  031040
7762  045626  051440  041505  047524
7763  045634  051522  030440  026071
7764  045642  031040  020060  051117
7765  045650  031040     061
7766  045653     015  050012  042514          .ASCIZ  <CR><LF>/PLEASE TRY AGAIN/<CR><LF>
7767  045660  051501  020105  051124
7768  045666  020131  043501  044501
7769  045674  006516  000012
7770
7771                                  .SBTTL  ERROR MESSAGES
7772
7773  045700  005015  051105  047522  EM1:    .ASCIZ  <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
7774  045706  026122  047440  046116
7775  045714  020131  020060  044124
7776  045722  052522  033440  040440
7777  045730  046114  053517  042105
7778  045736  020054  051124  020131
7779  045744  043501  044501  006516
7780  045752  000012
7781  045754  051104  053111  020105  EM2:    .ASCIZ  /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
7782  045762  020043  047111  051040
7783  045770  041513  031123  041440
7784  045776  047101  047516  020124
7785  046004  042502  051040  040505
7786  046012  020104  040502  045503
7787  046020  041440  051117  042522
7788  046026  052103  054514  044440
7789  046034  020116  045522  051115
```

```
7790  046042  000062
7791  046044  005015  041101  051117  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
7792  046052  020124  042524  052123
7793  046060  027123  027056  047125
7794  046066  054105  042520  052103
7795  046074  042105  052040  046511
7796  046102  020105  052517  020124
7797  046110  052101  050040  036503
7798  046116     000
7799  046117     015  040412  047502  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
7800  046124  052122  052040  051505
7801  046132  051524  027056  052456
7802  046140  042516  050130  041505
7803  046146  042524  020104  047111
7804  046154  042524  051122  050125
7805  046162  020124  052101  050040
7806  046170  036503     000
7807  046173     115  051504  051440  EM5:    .ASCIZ  /MDS SET IN RKCS2/
7808  046200  052105  044440  020116
7809  046206  045522  051503  000062
7810  046214  043125  020105  042523  EM6:    .ASCIZ  /UFE SET IN RKCS2/
7811  046222  020124  047111  051040
7812  046230  041513  031123     000
7813  046235     104  040522  044440  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
7814  046242  020116  045522  051504
7815  046250  023040  047040  042105
7816  046256  044440  020116  045522
7817  046264  051503  020062  042522
7818  046272  042523  035524  053440
7819  046300  047522  043516  050040
7820  046306  051117  020124  042523
7821  046314  042514  052103  042105
7822  046322  000077
7823  046324  051104  053111  020105  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
7824  046332  051120  051505  047105
7825  046340  020124  052502  020124
7826  046346  047516  020124  050123
7827  046354  041505  043111  042511
7828  046362  020104  054502  047440
7829  046370  042520  040522  047524
7830  046376  000122
7831  046400  051104  053111  020105  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
7832  046406  047516  020124  051120
7833  046414  051505  047105  020124
7834  046422  052502  020124  050123
7835  046430  041505  043111  042511
7836  046436  020104  054502  047440
7837  046444  042520  040522  047524
7838  046452  000122
7839  046454  041101  051117  020124  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
7840  046462  042524  052123  027123
7841  046470  027056  040503  047116
7842  046476  052117  051040  043105
7843  046504  051105  047105  042503
7844  046512  041440  047117  051124
7845  046520  046117  042514  020122
```

```
 7846  046526  042522  044507  052123
 7847  046534  051105     000
 7848  046537     104  040522  044440  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
 7849  046544  020116  045522  051504
 7850  046552  023040  047040  042105
 7851  046560  044440  020116  045522
 7852  046566  051503  020062  047502
 7853  046574  044124  051440  052105
 7854  046602     000
 7855  046603     103  047117  051124  EM12:   .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
 7856  046610  046117  042514  020122
 7857  046616  047516  020124  042522
 7858  046624  042101  020131  047111
 7859  046632  051040  041513  030523
 7860  046640     000
 7861  046641     116  020117  052101  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
 7862  046646  047124  044440  020116
 7863  046654  045522  051501  043117
 7864  046662     000
 7865  046663     127  047522  043516  EM14:   .ASCIZ  /WRONG ATTN IN RKASOF/
 7866  046670  040440  052124  020116
 7867  046676  047111  051040  040513
 7868  046704  047523  000106
 7869  046710  051104  054504  047040  EM15:   .ASCIZ  /DRDY NOT CLEARED IN RKMR2/
 7870  046716  052117  041440  042514
 7871  046724  051101  042105  044440
 7872  046732  020116  045522  051115
 7873  046740  000062
 7874  046742  051504  020103  047516  EM16:   .ASCIZ  /DSC NOT SET IN RKMR2/
 7875  046750  020124  042523  020124
 7876  046756  047111  051040  046513
 7877  046764  031122     000
 7878  046767     115  051505  040523  EM17:   .ASCIZ  /MESSAGE A0 ERROR/
 7879  046774  042507  040440  020060
 7880  047002  051105  047522  000122
 7881  047010  042515  051523  043501  EM18:   .ASCIZ  /MESSAGE B0 ERROR/
 7882  047016  020105  030102  042440
 7883  047024  051122  051117     000
 7884  047031     115  051505  040523  EM19:   .ASCIZ  /MESSAGE A1 ERROR/
 7885  047036  042507  040440  020061
 7886  047044  051105  047522  000122
 7887  047052  042515  051523  043501  EM20:   .ASCIZ  /MESSAGE B1 ERROR/
 7888  047060  020105  030502  042440
 7889  047066  051122  051117     000
 7890  047073     103  051105  020122  EM21:   .ASCIZ  /CERR SET IN RKCS1/
 7891  047100  042523  020124  047111
 7892  047106  051040  041513  030523
 7893  047114     000
 7894  047115     104  047517  020122  EM22:   .ASCIZ  /DOOR STATUS IN RKMR2 NOT CLEARED/
 7895  047122  052123  052101  051525
 7896  047130  044440  020116  045522
 7897  047136  051115  020062  047516
 7898  047144  020124  046103  040505
 7899  047152  042522  000104
 7900  047156  050123  047111  046104  EM23:   .ASCIZ  /SPINDLE ON SET IN RKMR2/
 7901  047164  020105  047117  051440
```

```
7902  047172  052105  044440  020116
7903  047200  045522  051115  000062
7904  047206  047526  052514  042515  EM24:   .ASCIZ  /VOLUME VALID NOT SET IN RKMR2/
7905  047214  053040  046101  042111
7906  047222  047040  052117  051440
7907  047230  052105  044440  020116
7908  047236  045522  051115  000062
7909  047244  040503  052122  044522  EM25:   .ASCIZ  /CARTRIDGE STATUS IN RKMR2 NOT CLEARED/
7910  047252  043504  020105  052123
7911  047260  052101  051525  044440
7912  047266  020116  045522  051115
7913  047274  020062  047516  020124
7914  047302  046103  040505  042522
7915  047310  000104
7916  047312  047526  052514  042515  EM26:   .ASCIZ  /VOLUME VALID NOT CLEARED IN RKMR2/
7917  047320  053040  046101  042111
7918  047326  047040  052117  041440
7919  047334  042514  051101  042105
7920  047342  044440  020116  045522
7921  047350  051115  000062
7922  047354  042516  020104  047516  EM27:   .ASCIZ  /NED NOT SET IN RKCS2/
7923  047362  020124  042523  020124
7924  047370  047111  051040  041513
7925  047376  031123     000
7926  047401     126  046117  046525  EM28:   .ASCIZ  /VOLUME VALID SET IN RKMR2/
7927  047406  020105  040526  044514
7928  047414  020104  042523  020124
7929  047422  047111  051040  046513
7930  047430  031122     000
7931  047433     104  041523  047040  EM29:   .ASCIZ  /DSC NOT SET IN RKMR2/
7932  047440  052117  051440  052105
7933  047446  044440  020116  045522
7934  047454  051115  000062
7935  047460  052101  047124  047040  EM30:   .ASCIZ  /ATTN NOT RESET IN RKASOF/
7936  047466  052117  051040  051505
7937  047474  052105  044440  020116
7938  047502  045522  051501  043117
7939  047510     000
7940  047511     116  042105  047040  EM31:   .ASCIZ  /NED NOT CLEARED IN RKCS2/
7941  047516  052117  041440  042514
7942  047524  051101  042105  044440
7943  047532  020116  045522  051503
7944  047540  000062
7945  047542  050123  047111  046104  EM32:   .ASCIZ  /SPINDLE ON NOT SET IN RKMR2/
7946  047550  020105  047117  047040
7947  047556  052117  051440  052105
7948  047564  044440  020116  045522
7949  047572  051115  000062
7950  047576  051104  053111  020105  EM33:   .ASCIZ  /DRIVE NOT READY IN RKMR2/
7951  047604  047516  020124  042522
7952  047612  042101  020131  047111
7953  047620  051040  046513  031122
7954  047626     000
7955  047627     104  047517  020122  EM34:   .ASCIZ  /DOOR STATUS BIT NOT SET IN RKMR2/
7956  047634  052123  052101  051525
7957  047642  041040  052111  047040
```

```
7958  047650  052117  051440  052105
7959  047656  044440  020116  045522
7960  047664  051115  000062
7961  047670  042510  042101  020123  EM35:   .ASCIZ  /HEADS HOME NOT SET IN RKMR2/
7962  047676  047510  042515  047040
7963  047704  052117  051440  052105
7964  047712  044440  020116  045522
7965  047720  051115  000062
7966  047724  054503  020114  042101  EM36:   .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
7967  047732  051104  044440  020116
7968  047740  045522  051115  020063
7969  047746  047516  020124  040523
7970  047754  042515  040440  020123
7971  047762  045522  041504     000
7972  047767     101  020103  047514  EM37:   .ASCIZ  /AC LOW NOT SET IN RKMR3/
7973  047774  020127  047516  020124
7974  050002  042523  020124  047111
7975  050010  051040  046513  031522
7976  050016     000
7977  050017     101  020103  047514  EM38:   .ASCIZ  /AC LOW DID NOT SET FAULT IN RKMR3/
7978  050024  020127  044504  020104
7979  050032  047516  020124  042523
7980  050040  020124  040506  046125
7981  050046  020124  047111  051040
7982  050054  046513  031522     000
7983  050061     103  046131  042040  EM39:   .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
7984  050066  043111  020106  020046
7985  050074  043117  051506  052105
7986  050102  044440  020116  045522
7987  050110  051115  020062  047516
7988  050116  020124  046103  040505
7989  050124  042522  000104
7990  050130  054503  020114  042101  EM40:   .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
7991  050136  051104  044440  020116
7992  050144  045522  051115  020063
7993  050152  047516  020124  046103
7994  050160  040505  042522  000104
7995  050166  054503  020114  042101  EM41:   .ASCIZ  /CYL ADDR IN RKMR3 DID NOT REMAIN CLEARED/
7996  050174  051104  044440  020116
7997  050202  045522  051115  020063
7998  050210  044504  020104  047516
7999  050216  020124  042522  040515
8000  050224  047111  041440  042514
8001  050232  051101  042105     000
8002  050237     116  042105  047040  EM42:   .ASCIZ  /NED NOT SET IN RKCS2/
8003  050244  052117  051440  052105
8004  050252  044440  020116  045522
8005  050260  051503  000062
8006  050264  041501  047514  047040  EM43:   .ASCIZ  /ACLO NOT CLEARED IN RKMR3/
8007  050272  052117  041440  042514
8008  050300  051101  042105  044440
8009  050306  020116  045522  051115
8010  050314  000063
8011  050316  047526  052514  042515  EM44:   .ASCIZ  /VOLUME VALID NOT CLEARED IN RKMR2/
8012  050324  053040  046101  042111
8013  050332  047040  052117  041440
```

```
8014  050340  042514  051101  042105
8015  050346  044440  020116  045522
8016  050354  051115  000062
8017  050360  047526  052514  042515  EM45:   .ASCIZ  /VOLUME VALID SET IN RKMR2 AFTER HEADS LOADED/
8018  050366  053040  046101  042111
8019  050374  051440  052105  044440
8020  050402  020116  045522  051115
8021  050410  020062  043101  042524
8022  050416  020122  042510  042101
8023  050424  020123  047514  042101
8024  050432  042105     000
8025  050435     116  047117  042455  EM46:   .ASCIZ  /NON-EXECUTABLE FUNCTION (NXF) NOT SET IN RKMR3/
8026  050442  042530  052503  040524
8027  050450  046102  020105  052506
8028  050456  041516  044524  047117
8029  050464  024040  054116  024506
8030  050472  047040  052117  051440
8031  050500  052105  044440  020116
8032  050506  045522  051115  000063
8033  050514  054503  044514  042116  EM47:   .ASCIZ  /CYLINDER ADDRESS CHANGED FROM 0/
8034  050522  051105  040440  042104
8035  050530  042522  051523  041440
8036  050536  040510  043516  042105
8037  050544  043040  047522  020115
8038  050552  000060
8039  050554  051127  052111  020105  EM48:   .ASCIZ  /WRITE LOCK IN RKMR2 NOT CLEARED/
8040  050562  047514  045503  044440
8041  050570  020116  045522  051115
8042  050576  020062  047516  020124
8043  050604  046103  040505  042522
8044  050612  000104
8045  050614  051127  052111  020105  EM49:   .ASCIZ  /WRITE LOCK IN RKMR2 NOT SET/
8046  050622  047514  045503  044440
8047  050630  020116  045522  051115
8048  050636  020062  047516  020124
8049  050644  042523  000124
8050  050650  051127  052111  020105  EM50:   .ASCIZ  /WRITE LOCK ERROR IN RKMR3 NOT SET/
8051  050656  047514  045503  042440
8052  050664  051122  051117  044440
8053  050672  020116  045522  051115
8054  050700  020063  047516  020124
8055  050706  042523  000124
8056  050712  051127  052111  020105  EM51:   .ASCIZ  /WRITE LOCK DID NOT OCCUR AT SECTOR BOUNDRY/
8057  050720  047514  045503  042040
8058  050726  042111  047040  052117
8059  050734  047440  041503  051125
8060  050742  040440  020124  042523
8061  050750  052103  051117  041040
8062  050756  052517  042116  054522
8063  050764     000
8064  050765     125  051516  047040  EM52:   .ASCIZ  /UNS NOT SET IN RKMR3/
8065  050772  052117  051440  052105
8066  051000  044440  020116  045522
8067  051006  051115  000063
8068
8069  051012  047125  047514  042101  EM53:   .ASCIZ  /UNLOAD NOT SET IN RKMR2/
```

```
8070  051020  047040  052117  051440
8071  051026  052105  044440  020116
8072  051034  045522  051115  000062
8073  051042  040503  047116  052117  EM54:   .ASCIZ  /CANNOT FIND MULT. DRIVE SELECT IN RKCS2/
8074  051050  043040  047111  020104
8075  051056  052515  052114  020056
8076  051064  051104  053111  020105
8077  051072  042523  042514  052103
8078  051100  044440  020116  045522
8079  051106  051503  000062
8080  051112  052101  047124  047040  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
8081  051120  052117  041440  042514
8082  051126  051101  042105  044440
8083  051134  020116  045522  051501
8084  051142  043117     000
8085  051145     125  042516  050130  EM56:   .ASCIZ  /UNEXPECTED MEMORY PARITY ERROR TRAP/
8086  051152  041505  042524  020104
8087  051160  042515  047515  054522
8088  051166  050040  051101  052111
8089  051174  020131  051105  047522
8090  051202  020122  051124  050101
8091  051210     000
8092  051211     103  051105  020122  EM57:   .ASCIZ  /CERR IN RKCS1 NOT SET/
8093  051216  047111  051040  041513
8094  051224  030523  047040  052117
8095  051232  051440  052105     000
8096  051237     120  051101  052111  EM58:   .ASCIZ  /PARITY NOT SET IN RKMR3/
8097  051244  020131  047516  020124
8098  051252  042523  020124  047111
8099  051260  051040  046513  031522
8100  051266     000
8101  051267     103  047524  051440  EM59:   .ASCIZ  /CTO SET IN RKCS1/
8102  051274  052105  044440  020116
8103  051302  045522  051503  000061
8104  051310  042510  042101  020123  EM60:   .ASCIZ  /HEADS HOME NOT FOUND IN RKMR2/
8105  051316  047510  042515  047040
8106  051324  052117  043040  052517
8107  051332  042116  044440  020116
8108  051340  045522  051115  000062
8109  051346  054116  020106  044504  EM61:   .ASCIZ  /NXF DID NOT SET FAULT/
8110  051354  020104  047516  020124
8111  051362  042523  020124  040506
8112  051370  046125  000124
8113  051374  052104  020105  042523  EM62:   .ASCIZ  /DTE SET IN RKER/
8114  051402  020124  047111  051040
8115  051410  042513  000122
8116  051414  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
8117  051422  020124  047111  051040
8118  051430  041513  031123     000
8119  051435     122  042113  020103  EM64:   .ASCII  /RKDC & RKDA INDICATE THAT WRITE CHECK ERROR/
8120  051442  020046  045522  040504
8121  051450  044440  042116  041511
8122  051456  052101  020105  044124
8123  051464  052101  053440  044522
8124  051472  042524  041440  042510
8125  051500  045503  042440  051122
```

```
8126  051506  051117
8127  051510  005015  041517  052503          .ASCIZ  <CR><LF>/OCCURRED AT CYL 411, TRACK 2, SECTOR 21/
8128  051516  051122  042105  040440
8129  051524  020124  054503  020114
8130  051532  030464  026061  052040
8131  051540  040522  045503  031040
8132  051546  020054  042523  052103
8133  051554  051117  031040  000061
8134  051562  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERROR/
8135  051570  040505  042504  020122
8136  051576  051105  047522  000122
8137  051604  054503  020114  042101  EM66:   .ASCIZ  /CYL ADDR IN RKMR3 INCORRECT/
8138  051612  051104  044440  020116
8139  051620  045522  051115  020063
8140  051626  047111  047503  051122
8141  051634  041505  000124
8142  051640  042516  020104  042523  EM67:   .ASCIZ  /NED SET IN RKCS2/
8143  051646  020124  047111  051040
8144  051654  041513  031123     000
8145  051661     103  047101  047516  EM68:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
8146  051666  020124  042522  042101
8147  051674  041040  042101  051440
8148  051702  041505  047524  020122
8149  051710  047111  047506  046522
8150  051716  052101  047511  000116
8151  051724  047516  042040  044522  EM69:   .ASCII  /NO DRIVES FOUND ON BUSS/
8152  051732  042526  020123  047506
8153  051740  047125  020104  047117
8154  051746  041040  051525     123
8155  051753     015  051412  052105          .ASCIZ  <CR><LF>/SETUP CORRECTLY & PRESS 'CONTINUE'/<CR><LF>
8156  051760  050125  041440  051117
8157  051766  042522  052103  054514
8158  051774  023040  050040  042522
8159  052002  051523  023440  047503
8160  052010  052116  047111  042525
8161  052016  006447  000012
8162  052022  044127  046111  020105  EM70:   .ASCIZ  /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
8163  052030  040527  052111  047111
8164  052036  020107  047506  020122
8165  052044  047503  052116  020122
8166  052052  042522  042101  020131
8167  052060  051117  040440  052106
8168  052066  051105  041440  047117
8169  052074  051124  051040  040505
8170  052102  054504  051040  041505
8171  052110  042047     000
8172  052113     104  052105  041505  EM71:   .ASCIZ  /DETECTED 10 BAD SECTORS...ABORTING TEST/
8173  052120  042524  020104  030061
8174  052126  041040  042101  051440
8175  052134  041505  047524  051522
8176  052142  027056  040456  047502
8177  052150  052122  047111  020107
8178  052156  042524  052123     000
8179  052163     104  052105  041505  EM72:   .ASCIZ  /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
8180  052170  042524  020104  051502
8181  052176  020105  052502  020124
```

```
 8182  052204  047516  020124  044514
 8183  052212  052123  042105  044440
 8184  052220  020116  040502  020104
 8185  052226  042523  052103  051117
 8186  052234  043040  046111  000105
 8187  052242  042504  042524  052103  EM73:   .ASCII  /DETECTED BSE IN READ COMMAND/
 8188  052250  042105  041040  042523
 8189  052256  044440  020116  042522
 8190  052264  042101  041440  046517
 8191  052272  040515  042116
 8192  052276  005015  052502  020124          .ASCIZ  <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
 8193  052304  047516  020124  047111
 8194  052312  050040  042522  044526
 8195  052320  052517  020123  051127
 8196  052326  052111  020105  047503
 8197  052334  046515  047101  020104
 8198  052342  047524  051440  046501
 8199  052350  020105  042523  052103
 8200  052356  051117     000
 8201  052361     122  055124  047040  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
 8202  052366  052117  051440  052105
 8203  052374  044440  020116  045522
 8204  052402  051115  000062
 8205  052406  005015  042504  042524  EM75:   .ASCIZ  <CR><LF>/DETECTED 10 BAD CYLINDERS...ABORTING TEST/
 8206  052414  052103  042105  030440
 8207  052422  020060  040502  020104
 8208  052430  054503  044514  042116
 8209  052436  051105  027123  027056
 8210  052444  041101  051117  044524
 8211  052452  043516  052040  051505
 8212  052460  000124
 8213  052462  047516  042040  044522  EM76:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/
 8214  052470  042526  020123  047506
 8215  052476  047125  020104  047111
 8216  052504  042040  053105  041511
 8217  052512  020105  040515  020120
 8218  052520  022050  042504  046526
 8219  052526     051
 8220  052527     015  051412  052105          .ASCIZ  <CR><LF>/SETUP CORRECTLY & RESTART/<CR><LF>
 8221  052534  050125  041440  051117
 8222  052542  042522  052103  054514
 8223  052550  023040  051040  051505
 8224  052556  040524  052122  005015
 8225  052564     000
 8226  052565     127  044522  042524  EM80:   .ASCIZ  /WRITE CHECK ERROR SET IN RKCS2/
 8227  052572  041440  042510  045503
 8228  052600  042440  051122  051117
 8229  052606  051440  052105  044440
 8230  052614  020116  045522  051503
 8231  052622  000062
 8232  052624  051127  052111  020105  EM81:   .ASCIZ  /WRITE CHECK COMMAND NOT FUNCTIONING/
 8233  052632  044103  041505  020113
 8234  052640  047503  046515  047101
 8235  052646  020104  047516  020124
 8236  052654  052506  041516  044524
 8237  052662  047117  047111  000107
```

```
8238  052670  042522  042101  042040  EM82:   .ASCIZ  /READ DATA DID NOT COMPARE WITH WRITE DATA/
8239  052676  052101  020101  044504
8240  052704  020104  047516  020124
8241  052712  047503  050115  051101
8242  052720  020105  044527  044124
8243  052726  053440  044522  042524
8244  052734  042040  052101  000101
8245  052742  040504  040524  041440  EM83:   .ASCIZ  /DATA CHECK ERROR SET IN RKER/
8246  052750  042510  045503  042440
8247  052756  051122  051117  051440
8248  052764  052105  044440  020116
8249  052772  045522  051105     000
8250  052777     117  043106  042523  EM84:   .ASCIZ  /OFFSET REG IN RKMR2  NOT  177/
8251  053004  020124  042522  020107
8252  053012  047111  051040  046513
8253  053020  031122  020040  047516
8254  053026  020124  030440  033467
8255  053034     000
8256  053035     117  043106  042523  EM85:   .ASCIZ  /OFFSET STATUS BIT IN RKMR2 CLEARED/
8257  053042  020124  052123  052101
8258  053050  051525  041040  052111
8259  053056  044440  020116  045522
8260  053064  051115  020062  046103
8261  053072  040505  042522  000104
8262  053100  043117  051506  052105  EM86:   .ASCIZ  /OFFSET REG IN RKMR2 NOT 77/
8263  053106  051040  043505  044440
8264  053114  020116  045522  051115
8265  053122  020062  047516  020124
8266  053130  033467     000
8267  053133     127  044522  042524  EM87:   .ASCIZ  /WRITE CHECK FAILURE AT OFFSET IN RKASOF/
8268  053140  041440  042510  045503
8269  053146  043040  044501  052514
8270  053154  042522  040440  020124
8271  053162  043117  051506  052105
8272  053170  044440  020116  045522
8273  053176  051501  043117     000
8274  053203     116  020117  051127  EM88:   .ASCIZ  /NO WRITE CHECK ERROR/
8275  053210  052111  020105  044103
8276  053216  041505  020113  051105
8277  053224  047522  000122
8278  053230  042510  042101  020123  EM89:   .ASCIZ  /HEADS HOME NOT CLEARED IN RKMR2/
8279  053236  047510  042515  047040
8280  053244  052117  041440  042514
8281  053252  051101  042105  044440
8282  053260  020116  045522  051115
8283  053266  000062
8284  053270  051124  041501  020113  EM90:   .ASCIZ  /TRACK FOLL OK NOT SET IN RKMR2/
8285  053276  047506  046114  047440
8286  053304  020113  047516  020124
8287  053312  042523  020124  047111
8288  053320  051040  046513  031122
8289  053326     000
8290  053327     122  053105  047040  EM91:   .ASCIZ  /REV NOT SET IN RKMR2/
8291  053334  052117  051440  052105
8292  053342  044440  020116  045522
8293  053350  051115  000062
```

```
8294  053354  042522  020126  047516  EM92:   .ASCIZ  /REV NOT CLEARED IN RKMR2/
8295  053362  020124  046103  040505
8296  053370  042522  020104  047111
8297  053376  051040  046513  031122
8298  053404     000
8299  053405     122  040505  044504  EM93:   .ASCIZ  /READING WRONG CYLINDER # IN HEADER/
8300  053412  043516  053440  047522
8301  053420  043516  041440  046131
8302  053426  047111  042504  020122
8303  053434  020043  047111  044040
8304  053442  040505  042504  000122
8305  053450  043117  051506  052105  EM94:   .ASCIZ  /OFFSET STATUS BIT IN RKMR2 NOT CLEARED/
8306  053456  051440  040524  052524
8307  053464  020123  044502  020124
8308  053472  047111  051040  046513
8309  053500  031122  047040  052117
8310  053506  041440  042514  051101
8311  053514  042105     000
8312  053517     106  051117  040515  EM95:   .ASCIZ  /FORMAT BIT NOT SET IN RKMR2/
8313  053524  020124  044502  020124
8314  053532  047516  020124  042523
8315  053540  020124  047111  051040
8316  053546  046513  031122     000
8317  053553     103  047101  047516  EM96:   .ASCIZ  /CANNOT FIND SECTOR 17/
8318  053560  020124  044506  042116
8319  053566  051440  041505  047524
8320  053574  020122  033461     000
8321  053601     110  040505  020104  EM97:   .ASCIZ  /HEAD SWITCHING LONGER THAN 16 MS/
8322  053606  053523  052111  044103
8323  053614  047111  020107  047514
8324  053622  043516  051105  052040
8325  053630  040510  020116  033061
8326  053636  046440  000123
8327  053642  040503  047116  052117  EM98:   .ASCIZ  /CANNOT FIND CYLINDER 128/
8328  053650  043040  047111  020104
8329  053656  054503  044514  042116
8330  053664  051105  030440  034062
8331  053672     000
8332  053673     103  047101  047516  EM99:   .ASCIZ  /CANNOT FIND CYLINDER 256/
8333  053700  020124  044506  042116
8334  053706  041440  046131  047111
8335  053714  042504  020122  032462
8336  053722  000066
8337  053724  051104  053111  020105  EM100:  .ASCIZ  /DRIVE OFF TRACK SET IN RKMR3/
8338  053732  043117  020106  051124
8339  053740  041501  020113  042523
8340  053746  020124  047111  051040
8341  053754  046513  031522     000
8342  053761     104  042111  047040  EM101:  .ASCIZ  /DID NOT GO TO CYLINDER 0/
8343  053766  052117  043440  020117
8344  053774  047524  041440  046131
8345  054002  047111  042504  020122
8346  054010  000060
8347
8348                              .SBTTL  DATA HEADERS
8349
```

```
8350  054012  042524  052123  047040  DH1:    .ASCIZ  /TEST NO.  PC/
8351  054020  027117  020040  041520
8352  054026     000
8353  054027     122  046513  031122  DH2:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKCS1   RKCS2   RKASOF/
8354  054034  051011  046513  031522
8355  054042  051011  042513  004522
8356  054050  045522  051504  051011
8357  054056  041513  030523  051011
8358  054064  041513  031123  051011
8359  054072  040513  047523  000106
8360  054100  043101  042524  020122  DH3:    .ASCIZ  /AFTER AC SWITCHED OFF/
8361  054106  041501  051440  044527
8362  054114  041524  042510  020104
8363  054122  043117  000106
8364  054126  043101  042524  020122  DH4:    .ASCIZ  /AFTER PACK RE-INSERTED & HEADS LOADED/
8365  054134  040520  045503  051040
8366  054142  026505  047111  042523
8367  054150  052122  042105  023040
8368  054156  044040  040505  051504
8369  054164  046040  040517  042504
8370  054172  000104
8371  054174  045522  051115  004462  DH5:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDC    RKCS1   RKCS2/
8372  054202  045522  051115  004463
8373  054210  045522  051105  051011
8374  054216  042113  004523  045522
8375  054224  041504  051011  041513
8376  054232  030523  051011  041513
8377  054240  031123     000
8378  054243     122  046513  031122  DH6:    .ASCIZ  /RKMR2   RKMR3   RKDC   FROM CYL  TO CYL  CYL DIFF/
8379  054250  020040  051040  046513
8380  054256  031522  020040  051040
8381  054264  042113  020103  020040
8382  054272  051106  046517  041440
8383  054300  046131  020040  047524
8384  054306  041440  046131  020040
8385  054314  054503  020114  044504
8386  054322  043106     000
8387  054325     122  046513  031122  DH7:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDA    RKCS1   RKCS2/
8388  054332  051011  046513  031522
8389  054340  051011  042513  004522
8390  054346  045522  051504  051011
8391  054354  042113  004501  045522
8392  054362  051503  004461  045522
8393  054370  051503  000062
8394  054374  043101  042524  020122  DH8:    .ASCIZ  /AFTER DRIVE UNLOADED & DOOR OPENED/
8395  054402  051104  053111  020105
8396  054410  047125  047514  042101
8397  054416  042105  023040  042040
8398  054424  047517  020122  050117
8399  054432  047105  042105     000
8400  054437     101  052106  051105  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
8401  054444  051440  040524  052122
8402  054452  051440  044520  042116
8403  054460  042514  041440  046517
8404  054466  040515  042116  051040
8405  054474  041505  042047  041040
```

```
8406  054502  020131  051104  053111
8407  054510  000105
8408  054512  052101  042440  042116  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
8409  054520  047440  020106  042510
8410  054526  042101  046040  040517
8411  054534  044504  043516     000
8412  054541     101  052106  051105  DH11:   .ASCIZ  /AFTER MANUALLY LOADING HEADS WITH DOOR OPEN/
8413  054546  046440  047101  040525
8414  054554  046114  020131  047514
8415  054562  042101  047111  020107
8416  054570  042510  042101  020123
8417  054576  044527  044124  042040
8418  054604  047517  020122  050117
8419  054612  047105     000
8420  054615     101  052106  051105  DH12:   .ASCIZ  /AFTER DISK PACK REMOVED/
8421  054622  042040  051511  020113
8422  054630  040520  045503  051040
8423  054636  046505  053117  042105
8424  054644     000
8425  054645     101  052106  051105  DH13:   .ASCIZ  /AFTER MANUALLY LOADING HEADS WITH DISK PACK REMOVED/
8426  054652  046440  047101  040525
8427  054660  046114  020131  047514
8428  054666  042101  047111  020107
8429  054674  042510  042101  020123
8430  054702  044527  044124  042040
8431  054710  051511  020113  040520
8432  054716  045503  051040  046505
8433  054724  053117  042105     000
8434  054731     101  052106  051105  DH14:   .ASCIZ  /AFTER VOLUME VALID RESET/
8435  054736  053040  046117  046525
8436  054744  020105  040526  044514
8437  054752  020104  042522  042523
8438  054760  000124
8439  054762  044527  044124  052517  DH15:   .ASCIZ  /WITHOUT PACK COMMAND/
8440  054770  020124  040520  045503
8441  054776  041440  046517  040515
8442  055004  042116     000
8443  055007     101  052106  051105  DH16:   .ASCIZ  /AFTER UNIT SELECT PLUG REMOVED/
8444  055014  052440  044516  020124
8445  055022  042523  042514  052103
8446  055030  050040  052514  020107
8447  055036  042522  047515  042526
8448  055044  000104
8449  055046  043101  042524  020122  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
8450  055054  042522  040503  020114
8451  055062  047503  046515  047101
8452  055070  000104
8453  055072  043101  042524  020122  DH18:   .ASCIZ  /AFTER UNLOAD COMMAND/
8454  055100  047125  047514  042101
8455  055106  041440  046517  040515
8456  055114  042116     000
8457  055117     101  052106  051105  DH19:   .ASCIZ  /AFTER PACK COMMAND/
8458  055124  050040  041501  020113
8459  055132  047503  046515  047101
8460  055140  000104
8461  055142  043101  042524  020122  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
```

```
8462  055150  042523  042514  052103
8463  055156  042040  044522  042526
8464  055164  041440  046517  040515
8465  055172  042116     000
8466  055175     101  052106  051105  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
8467  055202  051440  041125  054523
8468  055210  052123  046505  041440
8469  055216  042514  051101     000
8470  055223     101  052106  051105  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
8471  055230  042040  044522  042526
8472  055236  041440  042514  051101
8473  055244  041440  046517  040515
8474  055252  042116     000
8475  055255     101  052106  051105  DH23:   .ASCIZ  /AFTER WRONG PORT SELECTED/
8476  055262  053440  047522  043516
8477  055270  050040  051117  020124
8478  055276  042523  042514  052103
8479  055304  042105     000
8480  055307     101  052106  051105  DH24:   .ASCIZ  /AFTER OFFSET COMMAND/
8481  055314  047440  043106  042523
8482  055322  020124  047503  046515
8483  055330  047101  000104
8484  055334  043101  042524  020122  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
8485  055342  042523  045505  041440
8486  055350  046517  040515  042116
8487  055356     000
8488  055357     101  052106  051105  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
8489  055364  051040  040505  020104
8490  055372  040504  040524  041440
8491  055400  046517  040515  042116
8492  055406     000
8493  055407     101  052106  051105  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
8494  055414  053440  044522  042524
8495  055422  042040  052101  020101
8496  055430  047503  046515  047101
8497  055436  000104
8498  055440  043101  042524  020122  DH28:   .ASCIZ  /AFTER BOTH PORTS DESELECTED/
8499  055446  047502  044124  050040
8500  055454  051117  051524  042040
8501  055462  051505  046105  041505
8502  055470  042524  000104
8503  055474  043101  042524  020122  DH29:   .ASCIZ  /AFTER CORRECT PORT SELECTED/
8504  055502  047503  051122  041505
8505  055510  020124  047520  052122
8506  055516  051440  046105  041505
8507  055524  042524  000104
8508  055530  043101  042524  020122  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
8509  055536  042522  042101  044040
8510  055544  040505  042504  020122
8511  055552  047503  046515  047101
8512  055560  000104
8513  055562  043101  042524  020122  DH31:   .ASCIZ  /AFTER DRIVE MANUALLY LOADED/
8514  055570  051104  053111  020105
8515  055576  040515  052516  046101
8516  055604  054514  046040  040517
8517  055612  042504  000104
```

```
8518  055616  043101  042524  020122  DH32:   .ASCIZ  /AFTER WRITE CHECK COMMAND/
8519  055624  051127  052111  020105
8520  055632  044103  041505  020113
8521  055640  047503  046515  047101
8522  055646  000104
8523  055650  052504  044522  043516  DH33:   .ASCIZ  /DURING SEEK COMMAND/
8524  055656  051440  042505  020113
8525  055664  047503  046515  047101
8526  055672  000104
8527  055674  043101  042524  020122  DH34:   .ASCIZ  /AFTER START SPINDLE COMMAND/
8528  055702  052123  051101  020124
8529  055710  050123  047111  046104
8530  055716  020105  047503  046515
8531  055724  047101  000104
8532  055730  043101  042524  020122  DH35:   .ASCIZ  /AFTER MANUALLY UNLOADING/
8533  055736  040515  052516  046101
8534  055744  054514  052440  046116
8535  055752  040517  044504  043516
8536  055760      000
8537  055761      122  046513  031122  DH36:   .ASCIZ  /RKMR2   RKMR3   RKCS1   RKCS2  FROM SECT  TO SECT/
8538  055766  020040  051040  046513
8539  055774  031522  020040  051040
8540  056002  041513  030523  020040
8541  056010  051040  041513  031123
8542  056016  020040  051106  046517
8543  056024  051440  041505  020124
8544  056032  052040  020117  042523
8545  056040  052103      000
8546  056043      101  052106  051105  DH37:   .ASCIZ  /AFTER TIMEOUT TO POWER DOWN/
8547  056050  052040  046511  047505
8548  056056  052125  052040  020117
8549  056064  047520  042527  020122
8550  056072  047504  047127      000
8551  056077      101  052106  051105  DH38:   .ASCIZ  /AFTER AC POWERED UP/
8552  056104  040440  020103  047520
8553  056112  042527  042522  020104
8554  056120  050125      000
8555  056123      101  052106  051105  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
8556  056130  053440  044522  042524
8557  056136  044040  040505  042504
8558  056144  020122  047503  046515
8559  056152  047101  000104
8560  056156  045522  051115  004462  DH40:   .ASCIZ  /RKMR2  RKMR3   RKDA    WORD#   HEADER WAS  SHOULD BE/
8561  056164  045522  051115  004463
8562  056172  045522  040504  053411
8563  056200  051117  021504  044011
8564  056206  040505  042504  020122
8565  056214  040527  020123  051440
8566  056222  047510  046125  020104
8567  056230  042502      000
8568  056233      104  051125  047111  DH41:   .ASCIZ  /DURING RECAL COMMAND/
8569  056240  020107  042522  040503
8570  056246  020114  047503  046515
8571  056254  047101  000104
8572  056260  047117  051440  041505  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
8573  056266  047524  051522  030040
```

```
8574   056274   031054   032054   033054
8575   056302   047440   020122   020070
8576   056310   041440   046131   032040
8577   056316   030061   052040   040522
8578   056324   045503   031040      000
8579   056331      106   051117   040515  DH44:   .ASCIZ  /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8580   056336   020124   020046   046101
8581   056344   020114   042522   042101
8582   056352   053455   044522   042524
8583   056360   052040   051505   051524
8584   056366   053440   046111   020114
8585   056374   042502   041040   050131
8586   056402   051501   042523   000104
8587   056410   043101   042524   020122  DH45:   .ASCIZ  /AFTER SEEK WITH VOLUME VALID=0/
8588   056416   042523   045505   053440
8589   056424   052111   020110   047526
8590   056432   052514   042515   053040
8591   056440   046101   042111   030075
8592   056446      000
8593   056447      101   052106   051105  DH46:   .ASCIZ  /AFTER WRITE DATA WITH VOLUME VALID=0/
8594   056454   053440   044522   042524
8595   056462   042040   052101   020101
8596   056470   044527   044124   053040
8597   056476   046117   046525   020105
8598   056504   040526   044514   036504
8599   056512   000060
8600   056514   043101   042524   020122  DH47:   .ASCIZ  /AFTER SEEK COMMAND WITH MOVEMENT/
8601   056522   042523   045505   041440
8602   056530   046517   040515   042116
8603   056536   053440   052111   020110
8604   056544   047515   042526   042515
8605   056552   052116      000
8606   056555      101   052106   051105  DH48:   .ASCIZ  /AFTER WRITE LOCK SWITCH DISABLED/
8607   056562   053440   044522   042524
8608   056570   046040   041517   020113
8609   056576   053523   052111   044103
8610   056604   042040   051511   041101
8611   056612   042514   000104
8612   056616   043101   042524   020122  DH49:   .ASCIZ  /AFTER WRITE LOCK SWITCH ENABLED/
8613   056624   051127   052111   020105
8614   056632   047514   045503   051440
8615   056640   044527   041524   020110
8616   056646   047105   041101   042514
8617   056654   000104
8618   056656   043101   042524   020122  DH50:   .ASCIZ  /AFTER WRITING WITH WRITE LOCK ENABLED/
8619   056664   051127   052111   047111
8620   056672   020107   044527   044124
8621   056700   053440   044522   042524
8622   056706   046040   041517   020113
8623   056714   047105   041101   042514
8624   056722   000104
8625   056724   043101   042524   020122  DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
8626   056732   042523   045505   052040
8627   056740   020117   042523   043114
8628   056746   041440   046517   040515
8629   056754   042116      000
```

```
8630  056757     127  052111  020110  DH52:   .ASCIZ  /WITH INTENTIONAL MISCOMPARE/
8631  056764  047111  042524  052116
8632  056772  047511  040516  020114
8633  057000  044515  041523  046517
8634  057006  040520  042522     000
8635  057013     104  051125  047111  DH53:   .ASCIZ  /DURING OFFSET COMMAND/
8636  057020  020107  043117  051506
8637  057026  052105  041440  046517
8638  057034  040515  042116     000
8639  057041     101  020124  040515  DH54:   .ASCIZ  /AT MAX POSITIVE OFFSET/
8640  057046  020130  047520  044523
8641  057054  044524  042526  047440
8642  057062  043106  042523  000124
8643  057070  052101  046440  054101  DH55:   .ASCIZ  /AT MAX NEGATIVE OFFSET/
8644  057076  047040  043505  052101
8645  057104  053111  020105  043117
8646  057112  051506  052105     000
8647  057117     122  046513  031122  DH56:   .ASCIZ  /RKMR2  RKMR3   RKDC    CYL #   HEADER WORD 0/
8648  057124  051011  046513  031522
8649  057132  051011  042113  004503
8650  057140  054503  020114  004443
8651  057146  042510  042101  051105
8652  057154  053440  051117  020104
8653  057162  000060
8654  057164  043101  042524  020122  DH57:   .ASCIZ  /AFTER WRITE COMMAND WITH OFFSET/
8655  057172  051127  052111  020105
8656  057200  047503  046515  047101
8657  057206  020104  044527  044124
8658  057214  047440  043106  042523
8659  057222  000124
8660  057224  040504  040524  053440  DH58:   .ASCIZ  /DATA WAS       SHOULD BE/
8661  057232  051501  004440  044123
8662  057240  052517  042114  041040
8663  057246  000105
8664  057250  043101  042524  020122  DH59:   .ASCIZ  /AFTER DRIVE SELECT COMMAND WITH EVEN PARITY/
8665  057256  051104  053111  020105
8666  057264  042523  042514  052103
8667  057272  041440  046517  040515
8668  057300  042116  053440  052111
8669  057306  020110  053105  047105
8670  057314  050040  051101  052111
8671  057322  000131
8672  057324  043101  042524  020122  DH60:   .ASCIZ  /AFTER WRITE LOCK ENABLED DURING CONTINUOUS WRITING/
8673  057332  051127  052111  020105
8674  057340  047514  045503  042440
8675  057346  040516  046102  042105
8676  057354  042040  051125  047111
8677  057362  020107  047503  052116
8678  057370  047111  047525  051525
8679  057376  053440  044522  044524
8680  057404  043516     000
8681  057407     122  046513  031122  DH61:   .ASCII  /RKMR2  RKMR3   RKCS1   RKCS2   RKDA    WORD    EXPECTED/
8682  057414  051011  046513  031522
8683  057422  051011  041513  030523
8684  057430  051011  041513  031123
8685  057436  051011  042113  004501
```

```
 8686  057444  047527  042122  042411
 8687  057452  050130  041505  042524
 8688  057460     104
 8689  057461     015  004412  004411          .ASCIZ  <CR><LF>/                                  @ WRL   WAS     WORD/
 8690  057466  040011  053440  046122
 8691  057474  053411  051501  053411
 8692  057502  051117  000104
 8693  057506  045522  051115  004462  DH62:   .ASCII  /RkMR2  RKMR3   RKCS1   RKCS2   RKDA    WORD    EXPECTED/
 8694  057514  045522  051115  004463
 8695  057522  045522  051503  004461
 8696  057530  045522  051503  004462
 8697  057536  045522  040504  053411
 8698  057544  051117  004504  054105
 8699  057552  042520  052103  042105
 8700  057560  005015  004411  004411          .ASCII  <CR><LF>/                                  BEFORE  WAS     WORD/
 8701  057566  042502  047506  042522
 8702  057574  053411  051501  053411
 8703  057602  051117     104
 8704  057605     015  004412  004411          .ASCIZ  <CR><LF>/                                  WRL/
 8705  057612  053411  046122     000
 8706  057617     101  052106  051105  DH63:   .ASCIZ  /AFTER WRITE LOCK ENABLED FROM AC FAILURE/
 8707  057624  053440  044522  042524
 8708  057632  046040  041517  020113
 8709  057640  047105  041101  042514
 8710  057646  020104  051106  046517
 8711  057654  040440  020103  040506
 8712  057662  046111  051125  000105
 8713  057670  043101  042524  020122  DH64:   .ASCIZ  /AFTER MDS DETECTED IN RKCS2/
 8714  057676  042115  020123  042504
 8715  057704  042524  052103  042105
 8716  057712  044440  020116  045522
 8717  057720  051503  000062
 8718  057724  043101  042524  020122  DH65:   .ASCIZ  /AFTER SEARCHING ALL DRIVES PRESENT/
 8719  057732  042523  051101  044103
 8720  057740  047111  020107  046101
 8721  057746  020114  051104  053111
 8722  057754  051505  050040  042522
 8723  057762  042523  052116     000
 8724  057767     124  051505  020124  DH66:   .ASCIZ  /TEST NO.       TRAP PC/
 8725  057774  047516  004456  051124
 8726  060002  050101  050040  000103
 8727  060010  043101  042524  020122  DH67:   .ASCIZ  /AFTER TIMEOUT TO ENABLE WRITE LOCK/
 8728  060016  044524  042515  052517
 8729  060024  020124  047524  042440
 8730  060032  040516  046102  020105
 8731  060040  051127  052111  020105
 8732  060046  047514  045503     000
 8733  060053     122  046513  031122  DH68:   .ASCIZ  /RKMR2  RKMR3   RKCS1   RKCS2   RKDC    RKDA/
 8734  060060  051011  046513  031522
 8735  060066  051011  041513  030523
 8736  060074  051011  041513  031123
 8737  060102  051011  042113  004503
 8738  060110  045522  040504     000
 8739  060115     105  050130  041505  DH69:   .ASCIZ  /EXPECT EXPECT  EXPECT  EXPECT  EXPECT  EXPECT  EXPECT/
 8740  060122  004524  054105  042520
 8741  060130  052103  042411  050130
```

```
8742  060136  041505  004524  054105
8743  060144  042520  052103  042411
8744  060152  050130  041505  004524
8745  060160  054105  042520  052103
8746  060166  042411  050130  041505
8747  060174  000124
8748  060176  041501  052524  046101  DH70:   .ASCIZ  /ACTUAL ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL  ACTUAL/
8749  060204  040411  052103  040525
8750  060212  004514  041501  052524
8751  060220  046101  040411  052103
8752  060226  040525  004514  041501
8753  060234  052524  046101  040411
8754  060242  052103  040525  004514
8755  060250  041501  052524  046101
8756  060256     000
8757  060257     101  004460  030102  DH71:   .ASCIZ  /A0      B0      A1      B1      A2      B2      B3/
8758  060264  040411  004461  030502
8759  060272  040411  004462  031102
8760  060300  041011  000063
8761  060304  045522  051503  004461  DH72:   .ASCIZ  /RKCS1  RKCS2   RKASOF  RKER    RKDS    RKDC/
8762  060312  045522  051503  004462
8763  060320  045522  051501  043117
8764  060326  051011  042513  004522
8765  060334  045522  051504  051011
8766  060342  042113  000103
8767  060346  051515  020107  020101  DH73:   .ASCIZ  /MSG A & B IN RKMR2 & RKMR3 RESP ARE INVALID/
8768  060354  020046  020102  047111
8769  060362  051040  046513  031122
8770  060370  023040  051040  046513
8771  060376  031522  051040  051505
8772  060404  020120  051101  020105
8773  060412  047111  040526  044514
8774  060420  000104
8775  060422  047117  051440  041505  DH74:   .ASCIZ  /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
8776  060430  047524  051522  030440
8777  060436  026060  030440  026062
8778  060444  030440  026064  030440
8779  060452  026066  030440  020070
8780  060460  051117  031040  020060
8781  060466  054503  020114  030464
8782  060474  020060  051124  041501
8783  060502  020113  000062
8784
8785                          .SBTTL  ERROR OUTPUT DATA
8786
8787                          .EVEN
8788  060506  001214  001116  005434  DT1:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HCS1,HCS2,HASOF
8789  060514  005436  005422  005420
8790  060522  005406  005410  005424
8791  060530  001214  001116  005434  DT3:    $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,$TMP3,WD1,WD2
8792  060536  005436  005406  005410
8793  060544  001166  001502  001504
8794  060552  001214  001116  005434  DT4:    $TESTN,$ERRPC,HMR2,HMR3,HDC,FRCYL,TOCYL,CALDIF
8795  060560  005436  005426  001364
8796  060566  001366  001374
8797  060572  001214  001116  005434  DT5:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HDA,HCS1,HCS2
```

```
 8798  060600  005436  005422  005420
 8799  060606  005416  005406  005410
 8800  060614  001214  001354              DT6:    $TESTN,TRAPPC
 8801  060620  001214  001116  005434      DT7:    $TESTN,$ERRPC,HMR2,HMR3,HDA,WDCNT,HDWD,TEMP1
 8802  060626  005436  005416  001522
 8803  060634  001540  005444
 8804  060640  001214  001116  005434      DT8:    $TESTN,$ERRPC,HMR2,HMR3,HDC,TOCYL,FRCYL,CALDIF
 8805  060646  005436  005426  001366
 8806  060654  001364  001374
 8807  060660  001214  001116  005434      DT9:    $TESTN,$ERRPC,HMR2,HMR3,HDC,TOCYL,RHTAB
 8808  060666  005436  005426  001366
 8809  060674  001760
 8810  060676  001214  001116  005434      DT10:   $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,HDC,HDA
 8811  060704  005436  005406  005410
 8812  060712  005426  005416
 8813  060716  001214  001116  005476      DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
 8814  060724  005500  005502  005504
 8815  060732  005456  005460  005462
 8816  060740  005464
 8817  060742  005406  005410  005424              HCS1,HCS2,HASOF,HER,HDS,HDC
 8818  060750  005422  005420  005426
 8819
 8820  060756  001214  001116  005476      DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
 8821  060764  005500  005502  005504
 8822  060772  005506  005510
 8823  060776  005456  005460  005462              H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
 8824  061004  005464  005466  005470
 8825  061012  005406  005410  005424              HCS1,HCS2,HASOF,HER,HDS,HDC
 8826  061020  005422  005420  005426
 8827
 8828  061026  001214  001116  005476      DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
 8829  061034  005500  005502  005504
 8830  061042  005506  005510  005514
 8831  061050  005456  005460  005462              H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
 8832  061056  005464  005466  005470
 8833  061064  005474
 8834  061066  005406  005410  005424              HCS1,HCS2,HASOF,HER,HDS,HDC
 8835  061074  005422  005420  005426
 8836
 8837                                  .SBTTL  ERROR DATA FORMATS
 8838
 8839  061102  000002                  DF1:    2
 8840  061104     002     000                  .BYTE 2,0
 8841  061106  054027                          DH2
 8842  061110     007     000                  .BYTE 7,0
 8843
 8844  061112  000003                  DF3:    3
 8845  061114     000     000                  .BYTE 0,0
 8846  061116  054012                          DH1
 8847  061120     002     000                  .BYTE 2,0
 8848  061122  057407                          DH61
 8849  061124     007     000                  .BYTE 7,0
 8850
 8851  061126  000003                  DF4:    3
 8852  061130     000     000                  .BYTE 0,0
 8853  061132  054012                          DH1
```

```
8854  061134    002     000          .BYTE 2,0
8855  061136  057506                 DH62
8856  061140    007     000          .BYTE 7,0
8857
8858  061142  000001                 DF5:  1
8859  061144    002     000          .BYTE   2,0
8860
8861  061146  000003                 DF6:  3
8862  061150    000     000          .BYTE 0,0
8863  061152  054012                 DH1
8864  061154    002     000          .BYTE 2,0
8865  061156  054243                 DH6
8866  061160    006     000          .BYTE 6,0
8867
8868  061162  000002                 DF7:  2
8869  061164    002     000          .BYTE   2,0
8870  061166  060053                 DH68
8871  061170    006     000          .BYTE   6,0
8872  061172  000003                 DF10:  3
8873  061174    000     000          .BYTE 0,0
8874  061176  054012                 DH1
8875  061200    002     000          .BYTE 2,0
8876  061202  054027                 DH2
8877  061204    007     000          .BYTE 7,0
8878
8879  061206  000004                 DF12:  4
8880  061210    000     000          .BYTE   0,0
8881  061212  060346                 DH73
8882  061214    000     000          .BYTE   0,0
8883  061216  054012                 DH1
8884  061220    002     000          .BYTE   2,0
8885  061222  054027                 DH2
8886  061224    007     000          .BYTE   7,0
8887
8888  061226  000002                 DF14:  2
8889  061230    002     000          .BYTE   2,0
8890  061232  056156                 DH40
8891  061234    006     000          .BYTE   6,0
8892
8893
8894  061236  000003                 DF15:  3
8895  061240    000     000          .BYTE   0,0
8896  061242  054012                 DH1
8897  061244    002     000          .BYTE   2,0
8898  061246  054325                 DH7
8899  061250    007     000          .BYTE   7,0
8900
8901
8902  061252  000004                 DF17:  4
8903  061254    000     000          .BYTE   0,0
8904  061256  056331                 DH44
8905  061260    000     000          .BYTE   0,0
8906  061262  054012                 DH1
8907  061264    002     000          .BYTE   2,0
8908  061266  054027                 DH2
8909  061270    007     000          .BYTE   7,0
```

ERROR DATA FORMATS

```
8910  061272  000003                DF20:   3
8911  061274     000      000               .BYTE   0,0
8912  061276  054012                        DH1
8913  061300     002      000               .BYTE   2,0
8914  061302  057117                        DH56
8915  061304     005      000               .BYTE   5,0
8916
8917  061306  000007                DF21:   7
8918  061310     000      000               .BYTE   0,0
8919  061312  054012                        DH1
8920  061314     002      000               .BYTE   2,0
8921  061316  060115                        DH69
8922  061320     000      000               .BYTE   0,0
8923  061322  060257                        DH71
8924  061324     004      000               .BYTE   4,0
8925  061326  060176                        DH70
8926  061330     000      000               .BYTE   0,0
8927  061332  060257                        DH71
8928  061334     004      000               .BYTE   4,0
8929  061336  060304                        DH72
8930  061340     006      000               .BYTE   6,0
8931
8932  061342  000007                DF22:   7
8933  061344     000      000               .BYTE   0,0
8934  061346  054012                        DH1
8935  061350     002      000               .BYTE   2,0
8936  061352  060115                        DH69
8937  061354     000      000               .BYTE   0,0
8938  061356  060257                        DH71
8939  061360     006      000               .BYTE   6,0
8940  061362  060176                        DH70
8941  061364     000      000               .BYTE   0,0
8942  061366  060257                        DH71
8943  061370     006      000               .BYTE   6,0
8944  061372  060304                        DH72
8945  061374     006      000               .BYTE   6,0
8946
8947  061376  000007                DF23:   7
8948  061400     000      000               .BYTE   0,0
8949  061402  054012                        DH1
8950  061404     002      000               .BYTE   2,0
8951  061406  060115                        DH69
8952  061410     000      000               .BYTE   0,0
8953  061412  060257                        DH71
8954  061414     007      000               .BYTE   7,0
8955  061416  060176                        DH70
8956  061420     000      000               .BYTE   0,0
8957  061422  060257                        DH71
8958  061424     007      000               .BYTE   7,0
8959  061426  060304                        DH72
8960  061430     006      000               .BYTE   6,0
8961                                ;;*****************************************************************
8962                                .SBTTL  TYPE ERROR ROUTINE
8963                                ;*ENTRY JSR PC,TYP ERR
8964                                ;*RETURN RTS PC
8965                                ;*
```

```
 8966                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 8967                                   ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
 8968                                   ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
 8969                                   ;*THE ERROR.
 8970                                   ;;*****************************************************************
 8971  061432  104413                  TYPERR: SAVREG
 8972  061434  032777  020000  117476          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUTS?
 8973  061442  001107                          BNE     20$             ;YES-BRANCH
 8974  061444  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
 8975  061450  042700  177400                  BIC     #177400,R0      ;CLEAR SIGN EXTENSION
 8976  061454  005300                          DEC     R0              ;FORM INDEX FOR ERROR TABLE
 8977  061456  006300                          ASL     R0
 8978  061460  006300                          ASL     R0
 8979  061462  006300                          ASL     R0
 8980  061464  062700  005560          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
```

```
 8981  061470  012037  061504            MOV     (R0)+,2$        ;GET EM POINTER
 8982  061474  001404                    BEQ     3$              ;BRANCH IF THERE ISN'T ONE
 8983  061476  104401  001205            TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
 8984  061502  104401                    TYPE                    ;TYPE ERROR MESSAGE (EM)
 8985  061504  000000            2$:     .WORD   0               ;EM POINTER GOES HERE
 8986  061506  012037  061522    3$:     MOV     (R0)+,4$        ;GET DH POINTER
 8987  061512  001404                    BEQ     5$              ;BRANCH IF THERE ISN'T ONE
 8988  061514  104401  001205            TYPE    ,$CRLF          ;TYPE CR-LF
 8989  061520  104401                    TYPE                    ;TYPE DATA HEADER
 8990  061522  000000            4$:     .WORD   0               ;DH POINTER GOES HERE
 8991  061524  012001            5$:     MOV     (R0)+,R1        ;GET DT POINTER
 8992  061526  001455                    BEQ     20$             ;BRANCH IF THERE ARE NONE
 8993  061530  005004                    CLR     R4              ;SET INDENT SWITCH
 8994  061532  012000                    MOV     (R0)+,R0        ;GET DF POINTER
 8995  061534  012002                    MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
 8996  061536  001446                    BEQ     17$             ;DH NUM IS 0-BRANCH
 8997  061540  005104                    COM     R4              ;NO INDENT
 8998  061542  104401  001205            TYPE    ,$CRLF
 8999  061546  112003            10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
 9000  061550  105720                    TSTB    (R0)+           ;BUMP PAST FORMAT WORD
 9001  061552  005703                    TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
 9002  061554  001407                    BEQ     14$             ;NO - SKIP DATA PRINT
 9003  061556  013146            11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
 9004  061560  104402                    TYPOC                   ;TYPE IT
 9006  061562  005303                    DEC     R3              ;MORE DATA WORDS
 9006  061564  001403                    BEQ     14$             ;NO-BRANCH
 9007  061566  104401  061716            TYPE    ,SPACE2         ;TYPE SEPARATORS
 9008  061572  000771                    BR      11$             ;LOOP
 9009  061574  005302            14$:    DEC     R2              ;MORE DH'S?
 9010  061576  003431                    BLE     20$             ;NO-BRANCH
 9011  061600  104401  001205            TYPE    ,$CRLF
 9012  061604  005760  000002            TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
 9013  061610  001404                    BEQ     15$             ;YES-BRANCH BYPASS INDENT
 9014  061612  005104                    COM     R4              ;INDENT?
 9015  061614  001002                    BNE     15$             ;NO-BRANCH
 9016  061616  104401  061716            TYPE    ,SPACE2         ;YES-TYPE SPACES
 9017  061622  012037  061630    15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
 9018  061626  104401                    TYPE                    ;TYPE DH
 9019  061630  000000            16$:    .WORD   0               ;DH POINTER GOES HERE
 9020  061632  105710                    TSTB    (R0)            ;TYPE A DT?
 9021  061634  001003                    BNE     21$             ;YES-BRANCH
 9022  061636  062700  000002            ADD     #2,R0           ;INCREMENT DF POINTER
 9023  061642  000754                    BR      14$             ;SEE IF END OF DF BLOCK
 9024  061644  104401  001205    21$:    TYPE    ,$CRLF
 9025  061650  005704                    TST     R4              ;INDENT?
 9026  061652  001335                    BNE     10$             ;NO-BRANCH
 9027  061654  104401  061716    17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES
 9028  061660  000732                    BR      10$             ;LOOP
 9029  061662  104414            20$:    RESREG
 9030  061664  032777  010000  117246    BIT     #SW12,@SWR      ;SEE IF EXIT AFTER 20 ERRORS
 9031  061672  001410                    BEQ     25$             ;BR IF NO
 9032  061674  023727  001103  000024    CMP     $ERFLG,#20.     ;ELSE SEE IF HAVE 20 ERRORS
 9033  061702  001004                    BNE     25$             ;BR IF NO
 9034  061704  012706  001100            MOV     #STACK,SP       ;ELSE RESTORE STACK
 9035  061710  000137  024006            JMP     ENDRV           ;AND BYPASS DRIVE
 9036  061714  000207            25$:    RTS     PC
```

```
 9037  061716  020040      000          SPACE2: .ASCIZ/  /                ;2 SPACES
 9038                                   ;  ODT-11 -- V005A
 9039
 9040                                   ; DEC-11-UODPA-A-LA
 9041
 9042                                   ; COPYRIGHT 1969,1970,1972
 9043                                   ; DIGITAL EQUIPMENT CORPORATION
 9044                                   ; MAYNARD, MASSACHUSETTS 01754
 9045                                           .ENABL  ABS,AMA
 9046          061722                           .EVEN
 9047          062002                           .=.+60
 9048          000000                   R0      =       %0      ; REGISTER
 9049          000001                   R1      =       %1      ;  NAMING
 9050          000002                   R2      =       %2      ;   CONVENTIONS
 9051          000003                   R3      =       %3
 9052          000004                   R4      =       %4
 9053          000005                   R5      =       %5
 9054          000006                   SP      =       %6
 9055          000007                   PC      =       %7
 9056          177776                   ST      =       177776          ;STATUS REGISTER
 9057                                   ;
 9058          000014                   O.TVEC  =       14              ;TRT VECTOR LOCATION
 9059          000340                   O.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
 9060          000020                   O.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
 9061          000003                   TRT     =       000003          ;TRT INSTRUCTION
 9062          000006                   RTT     =       000006          ;RTT INSTRUCTION
 9063                                   ;
 9064                                   ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
 9065                                   ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
 9066                                   ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
 9067                                   ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
 9068                                   ;
 9069          177562                   O.RDB   =       177562  ;R DATA BUFFER
 9070          177560                   O.RCSR  =       177560  ;R C/SR
 9071          177566                   O.TDB   =       177566  ;T DATA BUFFER
 9072          177564                   O.TCSR  =       177564  ;T C/SR
 9073                                   ;
 9074                                   ;
 9075                                   ; INITIALIZE ODT
 9076                                   ;  USE O.ODT FOR A NORMAL ENTRY
 9077                                   ;  USE O.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
 9078                                   ;  USE O.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
 9079
 9080
 9081
 9082                                   ;
 9083  062002  000413                   O.ODT:  BR      O.STRT          ;NORMAL ENTRY
 9084  062004  000417                           BR      O.RST           ;RESTART
 9085  062006  013737  177776  061762   O.ENTR: MOV     ST,O.UST        ;RE-ENTER  --  SAVE STATUS
 9086  062014  013737  000016  177776           MOV     O.TVEC+2,ST     ;SET UP LOCAL STATUS
Z9087  062022  010737  061760                   MOV     PC,O.UPC        ;FAKE THE PC
 9088  062026  000137  063160                   JMP     O.BK1
 9089                                   ;
 9090  062032  012706  061742           O.STRT: MOV     #O.URO,SP       ;SET UP STACK
 9091  062036  010637  061756                   MOV     SP,O.USP        ;FAKE THE SAVED STACK
 9092  062042  000414                           BR      O.RST1          ;CLEAR BREAKPOINT TABLES
```

```
 9093  062044  004037  063366          0.RST:  JSR     0,0.SVR         ;SAVE REGISTERS
 9094  062050  013777  062000  177716          MOV     @0.UIN,@0.ADR1  ;REMOVE THE BREAKPOINT
 9095  062056  113704  061764                  MOVB    0.PRI,R4        ;GET ODT PRIORITY
 9096  062062  106004                          RORB    R4              ;SHIFT
 9097  062064  106004                          RORB    R4              ;  INTO
 9098  062066  106004                          RORB    R4              ;    POSITION
 9099  062070  110437  177776                  MOVB    R4,ST           ;STORE IN STATUS
Z9100  062074  000127                  0.RST1: JMP     (PC)+
 9101  062076  000403                          BR      0.45
 9102  062100  012737  000002  063070          MOV     #RTI,0.RTIT     ;SET TO RTI IF 11/20 OR /05
 9103  062106  105037  064007          0.45:   CLRB    0.P             ;DISALLOW PROCEED
 9104  062112  012737  000340  000016          MOV     #0.STM,0.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
 9105  062120  012737  063150  000014          MOV     #0.BRK,0.TVEC   ;PC TO TRT VECTOR
 9106  062126  000447                          BR      0.RALL          ;CLEAR BREAKPOINT TABLES
 9107                                  ;
 9108                                  ; SPECIAL NAME HANDLER
 9109                                  ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URD
 9110                                  ;
 9111  062130  004537  063610          0.REGT: JSR     5,0.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
 9112  062134  012704  064033                  MOV     #0.TL,R4        ;TABLE START ADDRESS
 9113  062140  120024                  0.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
 9114  062142  001413                          BEQ     0.SP            ;JUMP IF YES
 9115  062144  022704  064041                  CMP     #0.TL+0.LG,R4   ;IS THE SEARCH DONE?
 9116  062150  101373                          BHI     0.RSP           ;BRANCH IF NOT
 9117  062152  042700  177770                  BIC     #177770,R0      ;MASK OFF OCTAL
 9118  062156  010004                          MOV     R0,R4
 9119  062160  006304                  0.SP1:  ASL     R4
 9120  062162  062704  061742                  ADD     #0.URD,R4       ;GENERATE ADDRESS
 9121  062166  005202                          INC     R2              ;SET FOUND FLAG
 9122  062170  000444                          BR      0.SCAN          ;GO FIND NEXT CHARACTER
 9123  062172  162704  064024          0.SP:   SUB     #0.TL-7,R4      ;CORRECT CONSTANT
 9124  062176  000770                          BR      0.SP1
 9125                                  ;
 9126                                  ; ← HANDLER - OPEN INDEXED ON THE PC
 9127                                  ;
 9128  062200  004737  063734          0.ORPC: JSR     PC,0.TCLS
 9129  062204  010502                          MOV     R5,R2           ;CURRENT ADDRESS IN R2
 9130  062206  061202                          ADD     @R2,R2          ;COMPUTE
 9131  062210  006202                          ASR     R2              ;MOVE ONE BIT TO CARRY
 9132  062212  103421                          BCS     0.ERR           ;ERROR IF ODD NUMBER
 9133  062214  006302                          ASL     R2              ;RESTORE WORD
 9134  062216  005722                          TST     (R2)+           ;  AND INCREMENT BY TWO
 9135  062220  010205                          MOV     R2,R5           ;UPDATE CAD
 9136  062222  000137  062474                  JMP     0.OP2           ;GO FINISH UP
 9137                                  ;
 9138                                  ; B HANDLER - SET AND REMOVE BREAKPOINTS
 9139                                  ;
 9140  062226  005702                  0.BKPT: TST     R2              ;IF NO NUMBER TYPED
 9141  062230  001406                          BEQ     0.RALL          ;  REMOVE BREAKPOINT
 9142  062232  006204                          ASR     R4              ;CHECK IF ODD
 9143  062234  103410                          BCS     0.ERR           ;JUMP IF ODD
 9144  062236  006304                          ASL     R4              ;RESTORE ONE BIT
 9145  062240  010437  061774                  MOV     R4,0.ADR1       ;SET A BREAKPOINT
 9146
 9147  062244  000412                          BR      0.DCD
 9148
```

```
 9149
 9150
 9151  062246  012737  064050  061774  0.RALL: MOV     #0.TRTC,0.ADR1  ;CLEAR BREAKPOINT
 9152  062254  000406                          BR      0.DCD
 9153                                  ;
 9154                                  ; COMMAND DECODER - ODT11
 9155                                  ;
 9156                                  ;   REGISTERS R0-R4 MAY BE USED,
 9157                                  ;    REGISTER R5 WILL BE CONSIDERED SAFE
 9158                                  ;
 9159  062256  052705  000001          0.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
 9160  062262  012700  000077                  MOV     #'?,R0          ;  ? TO BE TYPED
 9161  062266  004537  063666                  JSR     5,0.FTYP        ; OUTPUT ?
 9162  062272  004537  063766          0.DCD:  JSR     5,0.CRLS        ;TYPE  <CR><LF>*
 9163  062276  005004                  0.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
 9164  062300  005002                          CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
 9165  062302  004537  063610          0.SCAN: JSR     5,0.GET         ;GET A CHAR, RETURN IN R0
 9166  062306  022700  000060                  CMP     #'0,R0          ;COMPARE WITH ASCII 0
 9167  062312  101013                          BHI     0.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
 9168  062314  022700  000067                  CMP     #'7,R0          ;COMPARE WITH ASCII 7
 9169  062320  103410                          BLO     0.CLGL          ;CHECK LEGALITY IF NOT OCTAL
 9170  062322  042700  177770                  BIC     #177770,R0      ;CONVERT TO BCD
 9171  062326  006304                          ASL     R4              ; MAKE ROOM
 9172  062330  006304                          ASL     R4              ;  IN
 9173  062332  006304                          ASL     R4              ;    R4
 9174  062334  060004                          ADD     R0,R4           ;PACK THREE BITS IN R4
 9175  062336  005202                          INC     R2              ;R2 HAS NUMERIC FLAG
 9176  062340  000760                          BR      0.SCAN          ;   AND TRY AGAIN
 9177  062342  005001          0.CLGL: CLR     R1                      ;CLEAR INDEX
 9178  062344  120061  064017  0.LGL1: CMPB    R0,0.LGCH(R1)           ;DO THE CODES MATCH?
 9179  062350  001405                          BEQ     0.LGL2          ;JUMP IF YES
 9180  062352  005201                          INC     R1              ; SET INDEX FOR NEXT SEARCH
 9181  062354  020127  000014                  CMP     R1,#0.CLGT      ;IS THE SEARCH DONE?
 9182  062360  103336                          BHIS    0.ERR           ;    OOPS!
 9183  062362  000770                          BR      0.LGL1          ;RE-LOOP
 9184  062364  006301          0.LGL2: ASL     R1                      ;MULTIPLY BY TWO
 9185  062366  000171  062372                  JMP     @0.LGDR(R1)     ;GO TO PROPER ROUTINE
 9186
 9187                                  ;
 9188  062372  062422          0.LGDR: 0.WRD   ;  /     OPEN WORD
 9189  062374  062454                  0.CRET  ;  CARRIAGE RETURN    CLOSE
 9190  062376  062130                  0.REGT  ;  $  REGISTER OPS
 9191  062400  062764                  0.GO    ;  G  GO TO ADDRESS K
 9192  062402  062466                  0.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
 9193  062404  062200                  0.ORPC  ;  ←  OPEN RELATED, INDEX - PC
 9194  062406  062520                  0.BACK  ;  ↑  OPEN PREVIOUS
 9195  062410  062530                  0.OFST  ;  O  OFFSET
 9196  062412  062606                  0.WSCH  ;  W  SEARCH WORD
 9197  062414  062602                  0.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
 9198  062416  062226                  0.BKPT  ;  B  BREAKPOINTS
 9199  062420  063072                  0.PROC  ;  P  PROCEED
 9200          000030          0.LGL   =       .-0.LGDR                ;LGL MUST EQUAL 2X CHLGT ALWAYS
 9201                                  ;
 9202                                  ; PROCESS / - OPEN WORD
 9203                                  ;
 9204  062422  005702          0.WRD:  TST     R2                      ;GET VALUE IF R2 IS NON-ZERO
```

```
 9205  062424  001410                        BEQ     0.WRDA          ;SKIP OTHERWISE
 9206  062426  010405                        MOV     R4,R5           ;   PUT VALUE IN CAD
 9207  062430  006205                0.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
 9208  062432  103711                0.ERR2: BCS     0.ERR           ;JUMP IF ODD ADDRESS
 9209  062434  006305                        ASL     R5              ;RESTORE THE CARRY BIT
 9210  062436  011500                        MOV     @R5,R0          ;GET CONTENTS OF WORD
 9211  062440  004537  063524                JSR     5,0.CADV        ;GO GET AND TYPE OUT @CAD
 9212  062444  000714                        BR      0.DCD1          ;GO BACK TO DECODER
 9213  062446  042705  000001        0.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
 9214  062452  000766                        BR      0.WRD1          ;GO BACK TO MAIN-LINE
 9215                                ;
 9216                                ; PROCESS CARRIAGE RETURN
 9217                                ;
 9218  062454  004737  063734        0.CRET: JSR     PC,0.TCLS       ;CLOSE LOCATION
 9219  062460  052705  000001                BIS     #1,R5           ;CLOSE EVERYTHING
 9220  062464  000702                        BR      0.DCD           ;RETURN TO DECODER
 9221                                ;
 9222                                ; PROCESS <LF>, OPEN NEXT WORD
 9223                                ;
 9224  062466  004737  063734        0.OP1:  JSR     PC,0.TCLS       ;CLOSE PRESENT CELL
 9225  062472  005725                        TST     (R5)+           ;GENERATE NEW ADDRESS
 9226  062474  004537  063760        0.OP2:  JSR     5,0.CRLF        ;<CR><LF>
 9227  062500  010500                        MOV     R5,R0           ;NUMBER TO TYPE
 9228  062502  004537  063524                JSR     5,0.CADV        ;  TYPE OUT ADDRESS
 9229  062506  012700  000057                MOV     #'/,R0          ;TYPE A /
 9230  062512  004537  063666                JSR     5,0.FTYP
 9231  062516  000744                        BR      0.WRD1          ;GO PROCESS IT
 9232                                ;
 9233                                ; PROCESS ↑, OPEN PREVIOUS WORD
 9234                                ;
 9235  062520  004737  063734        0.BACK: JSR     PC,0.TCLS
 9236  062524  005745                        TST     -(R5)           ;GENERATE NEW ADDRESS
 9237  062526  000762                        BR      0.OP2           ;GO DO THE REST
 9238                                ;
 9239                                ; PROCESS 0, COMPUTE OFFSET
 9240                                ;
 9241  062530  006205                0.OFST: ASR     R5              ;GET LOW ORDER BIT
 9242  062532  103737                        BCS     0.ERR2          ;ERROR IF CLOSED
 9243  062534  006305                        ASL     R5              ;RESTORE WORD
 9244  062536  012700  000040                MOV     #' ,R0          ;TYPE ONE BLANK
 9245  062542  004537  063666                JSR     5,0.FTYP        ;  AS A SEPARATOR
 9246  062546  160504                        SUB     R5,R4           ;COMPUTE
 9247  062550  005304                        DEC     R4
 9248  062552  005304                        DEC     R4              ;     16 BIT OFFSET
 9249  062554  010400                        MOV     R4,R0           ;TYPE A
 9250  062556  010402                        MOV     R4,R2           ;SAVE R4
 9251  062560  004537  063524                JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
 9252  062564  010200                        MOV     R2,R0
 9253  062566  006200                        ASR     R0              ;DIVIDE BY TWO
 9254  062570  103402                        BCS     0.OF1           ;BRANCH IF ODD
 9255  062572  004537  063524                JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
 9256  062576  000137  062276        0.OF1:  JMP     0.DCD1          ;ALL DONE
 9257                                ;
 9258                                ; SEARCHES - $MSK   HAS THE MASK
 9259                                ;            $MSK+2 HAS THE FWA
 9260                                ;            $MSK+4 HAS THE LWA
```

```
9261
9262
9263
9264
9265
9266  062602  005201                  O.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
9267  062604  000401                          BR      O.WDS
9268  062606  005001                  O.WSCH: CLR     R1              ;SET WORD SEARCH
9269  062610  005702                  O.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
9270  062612  001621                  O.ERR1: BEQ     O.ERR           ;ERROR IF NO OBJECT
9271  062614  013702  061770                  MOV     O.MSK+2,R2      ;SET ORIGIN
9272  062620  013705  061766                  MOV     O.MSK,R5        ;SET MASK
9273  062624  005105                          COM     R5              ;AND COMPLEMENT IT
9274  062626  020237  061772          O.WDS2: CMP     R2,O.MSK+4      ; IS THE SEARCH ALL DONE?
9275  062632  101217                          BHI     O.DCD           ;  YES
9276  062634  011200                          MOV     @R2,R0          ; GET OBJECT
9277  062636  005701                          TST     R1              ;NO
9278  062640  001027                          BNE     O.EFF1          ;BRANCH IF EFFECTIVE SEARCH
9279  062642  010046                          MOV     R0,-(SP)
9280  062644  010403                          MOV     R4,R3           ;EXCLUSIVE OR
9281  062646  040400                          BIC     R4,R0           ; IS DONE
9282  062650  042603                          BIC     (SP)+,R3        ;  IN A VERY
9283  062652  050003                          BIS     R0,R3           ;     FANCY MANNER HERE
9284  062654  040503                          BIC     R5,R3           ;AND RESULT WITH MASK
9285  062656  001016          O.WDS3: BNE     O.WDS4          ;RE-LOOP IF NO MATCH
9286  062660  010446                          MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
9287  062662  004537  063760                  JSR     5,O.CRLF        ;TYPE (CR,LF)
9288  062666  010200                          MOV     R2,R0           ;GET READY TO TYPE
9289  062670  004537  063524                  JSR     5,O.CADV        ;  TYPE ADDRESS
9290  062674  012700  000057                  MOV     #'/,R0          ;SLASH TO R0
9291  062700  004537  063666                  JSR     5,O.FTYP        ;TYPE IT
9292  062704  011200                          MOV     @R2,R0          ;GET CONTENTS
9293  062706  004537  063524                  JSR     5,O.CADV        ;TYPE CONTENTS
9294  062712  012604                          MOV     (SP)+,R4        ; RESTORE R4
9295  062714  005722          O.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
9296  062716  000743                          BR      O.WDS2          ;     RETURN
9297  062720  020004          O.EFF1: CMP     R0,R4           ; IS (X)=K?
9298  062722  001755                          BEQ     O.WDS3          ;TYPE IF EQUAL
9299  062724  010003                          MOV     R0,R3           ;(X) TO R3
9300  062726  060203                          ADD     R2,R3           ;(X)+X
9301  062730  005203                          INC     R3
9302  062732  005203                          INC     R3              ;(X)+X+2
9303  062734  020304                          CMP     R3,R4           ;IS (X)+X+2=K?
9304  062736  001747                          BEQ     O.WDS3          ;BRANCH IF EQUAL
9305  062740  042700  177400                  BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
9306  062744  110000                          MOVB    R0,R0           ;EXTEND SIGN
9307  062746  000257                          CCC
9308  062750  006300                          ASL     R0              ;MULTIPLY BY TWO
9309  062752  005200                          INC     R0              ;ADD TWO
9310  062754  005200                          INC     R0
9311  062756  060200                          ADD     R2,R0           ;ADD PC
9312  062760  020004                          CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
9313  062762  000735                          BR      O.WDS3
9314
9315                                  ;
9316                                  ; PROCESS G - GO
                                      ;
```

```
 9317  062764  105037  064007          0.GO:   CLRB    0.P             ;DISALLOW PROCEED
 9318  062770  006204                          ASR     R4              ;CHECK LOW ORDER BIT
 9319  062772  103617                          BCS     0.ERR2          ;ERROR IF ODD NUMBER
 9320  062774  006304                          ASL     R4              ;RESTORE WORD
 9321  062776  010437  061760                  MOV     R4,0.UPC        ;SET UP NEW PC
 9322  063002  112737  000340  177776          MOVB    #0.STM,ST       ;SET HIGH PRIORITY
 9323  063010  004537  063456                  JSR     5,0.RSTT        ;RESTORE TELETYPE
 9324  063014  105037  064006          0.TBIT: CLRB    0.T             ;CLEAR BOTH
 9325  063020  042737  000020  061762          BIC     #0.TBT,0.UST    ;  T-BIT FLAGS
 9326  063026  017737  176742  062000          MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
 9327  063034  013777  064050  176732          MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
 9328  063042  012600                  0.GO2:  MOV     (SP)+,R0        ;RESTORE
 9329  063044  012601                          MOV     (SP)+,R1        ; R0
 9330  063046  012602                          MOV     (SP)+,R2        ;   THRU
 9331  063050  012603                          MOV     (SP)+,R3
 9332  063052  012604                          MOV     (SP)+,R4        ;
 9333  063054  012605                          MOV     (SP)+,R5        ;   R5
 9334  063056  012606                          MOV     (SP)+,SP        ; AND SP
 9335  063060  013746  061762                  MOV     0.UST,-(SP)     ;   AND STATUS
 9336  063064  013746  061760                  MOV     0.UPC,-(SP)     ;   AND PC
 9337  063070  000006                  0.RTIT: RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
 9338                                          ;
 9339                                          ; PROCESS P - PROCEED
 9340                                          ;   ONLY ALLOWED AFTER A BREAKPOINT
 9341                                          ;
 9342  063072  105737  064007          0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
 9343  063076  001645                          BEQ     0.ERR1          ;NOT LEGAL
 9344  063100  105037  064007                  CLRB    0.P             ;CLEAR PROCEED FLAG
 9345  063104  005702                          TST     R2              ;WAS COUNT SPECIFIED?
 9346  063106  001402                          BEQ     0.PR1           ;NO
 9347  063110  010437  061776                  MOV     R4,0.CT         ;YES, PUT AWAY COUNT
 9348  063114  112737  000340  177776  0.PR1:  MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
 9349  063122  004537  063456                  JSR     5,0.RSTT        ;RESTORE TTY
 9350  063126  112737  000340  177776  0.C1:   MOVB    #0.STM,ST       ;SET HIGH PRIORITY
 9351  063134  105237  064006                  INCB    0.T             ;SET T-BIT FLAG
 9352  063140  052737  000020  061762          BIS     #0.TBT,0.UST    ;SET T-BIT
 9353  063146  000735                          BR      0.GO2
 9354                                          ;
 9355                                          ; BREAKPOINT HANDLER
 9356                                          ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
 9357                                          ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
 9358                                          ;  AND GIVES CONTROL TO THE COMMAND DECODER
 9359                                          ;
 9360  063150  012637  061760          0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
 9361  063154  012637  061762                  MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
 9362  063160  004037  063366          0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
 9363  063164  105737  064006                  TSTB    0.T             ;CHECK FOR T-BIT SET
 9364  063170  001311                          BNE     0.TBIT          ;JUMP IF SET
 9365  063172  013777  062000  176574          MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
 9366  063200  105737  061764                  TSTB    0.PRI           ;CHECK IF PRIORITY
 9367  063204  100003                          BPL     0.BK2           ;  IS AS SAME AS USER PGM
 9368  063206  113705  061762                  MOVB    0.UST,R5        ;PICK UP USER UST IF SO
 9369  063212  000407                          BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
 9370  063214  113705  061764          0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
 9371  063220  000257                          CCC                     ;CLEAR CARRY
 9372  063222  106005                          RORB    R5              ;SHIFT LOW ORDER BITS
```

```
 9373  063224  106005                          RORB    R5              ;  INTO
 9374  063226  106005                          RORB    R5              ;    HIGH ORDER
 9375  063230  106005                          RORB    R5              ;      POSITION
 9376  063232  110537  177776          0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
 9377  063236  013705  061760                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
 9378  063242  005745                          TST     -(R5)           ;SUBTRACT TWO
 9379  063244  010537  061760                  MOV     R5,0.UPC        ;FROM THE USER'S PC
 9380  063250  020537  061774                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
 9381  063254  001417                          BEQ     0.B2            ;JUMP IF FOUND
 9382  063256  004537  063424                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
 9383  063262  004537  063760                  JSR     5,0.CRLF
 9384  063266  012704  064012                  MOV     #0.BD,R4        ;ERROR, NOTHING FOUND
 9385  063272  012703  064013                  MOV     #0.BD+1,R3
 9386  063276  004537  063652                  JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
 9387  063302  010500                          MOV     R5,R0
 9388  063304  042737  000020  061762          BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
 9389  063312  000420                          BR      0.B3            ; AND CONTINUE
 9390  063314  005337  061776          0.B2:   DEC     0.CT
 9391  063320  003302                          BGT     0.C1            ;JUMP IF REPEAT
 9392  063322  012737  000001  061776          MOV     #1,0.CT         ;RESET COUNT TO 1
 9393  063330  105237  064007                  INCB    0.P             ;ALLOW PROCEED
 9394  063334  004537  063424                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
 9395  063340  012700  000102                  MOV     #'B,R0
 9396  063344  004537  063666                  JSR     5,0.FTYP        ;TYPE "B"
 9397  063350  013700  061774                  MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
 9398  063354  004537  063524          0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
 9399  063360  005005                          CLR     R5              ;CLEAR CAD
 9400  063362  000137  062272                  JMP     0.DCD           ;GO TO DECODER
 9401                                  ;
 9402                                  ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
 9403                                  ;
 9404  063366  012637  064004          0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
 9405  063372  010637  061756                  MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
 9406  063376  012706  061756                  MOV     #0.USP,SP       ;SET TO INTERNAL STACK
 9407  063402  010546                          MOV     R5,-(SP)        ;SAVE
 9408  063404  010446                          MOV     R4,-(SP)        ; REGISTERS
 9409  063406  010346                          MOV     R3,-(SP)        ;1
 9410  063410  010246                          MOV     R2,-(SP)        ; THRU
 9411  063412  010146                          MOV     R1,-(SP)        ;    5
 9412  063414  013746  064004                  MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
 9413  063420  005746                          TST     -(SP)
 9414  063422  000200                          RTS     R0
 9415                                  ;
 9416                                  ; SAVE TELETYPE STATUS
 9417                                  ;
 9418  063424  113737  177560  064010  0.SVTT: MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
 9419  063432  113737  177564  064011          MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
 9420  063440  105037  177560                  CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
 9421  063444  105037  177564                  CLRB    0.TCSR          ;  BITS IN BOTH C/SR
 9422  063450  004537  063760                  JSR     5,0.CRLF        ;TYPE <CR,LF>
 9423  063454  000205                          RTS     R5
 9424                                  ;
 9425                                  ; RESTORE TELETYPE STATUS
 9426                                  ;
 9427  063456  004537  063760          0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
 9428  063462  105737  177564                  TSTB    0.TCSR          ;WAIT READY ON PRINTER
```

```
 9429  063466  100375                    BPL     .-4
 9430  063470  032737  004000  177560    BIT     #4000,@.RCSR    ;CHECK BUSY FLAG ON READER
 9431  063476  001403                    BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
 9432  063500  105737  177560            TSTB    @.RCSR          ;WAIT READY
 9433  063504  100375                    BPL     .-4             ;    ON READER
 9434  063506  113737  064010  177560  0.RSE1: MOVB    @0.CSR1,@.RCSR  ;RESTORE
 9435  063514  113737  064011  177564          MOVB    @0.CSR2,@.TCSR  ;  THE STATUS REGISTERS
 9436  063522  000205                    RTS     R5
 9437                                    ;
 9438                                    ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
 9439                                    ; WORD IS IN R0
 9440                                    ;
 9441  063524  010246                  0.CADV: MOV     R2,-(SP)        ;SAVE R2
 9442  063526  012704  064047                  MOV     #@0.BUF+6,R4    ;BUFFER START ADDRESS
 9443  063532  012746  000060                  MOV     #'0,-(SP)               ;CONSTANT ASCII 0
 9444  063536  010002                  0.SPC:  MOV     R0,R2           ; GET
 9445  063540  042702  177770                  BIC     #177770,R2      ;    OCTAL CHARACTER
 9446  063544  061602                          ADD     @SP,R2          ;CONVERT TO ASCII
 9447  063546  110244                          MOVB    R2,-(R4)        ;STORE IN BUFFER
 9448  063550  006200                          ASR     R0              ;SHIFT THIS MESS
 9449  063552  006200                          ASR     R0              ; RIGHT
 9450  063554  006200                          ASR     R0              ;   THREE WHOLE PLACES
 9451  063556  020427  064042                  CMP     R4,#@0.BUF+1    ;DONE?
 9452  063562  101365                          BHI     0.SPC           ;    NO
 9453  063564  042700  177776                  BIC     #177776,R0      ;GET LAST BIT
 9454  063570  062600                          ADD     (SP)+,R0                ;CONVERT TO ASCII
 9455  063572  110044                          MOVB    R0,-(R4)        ;AND PUT IT AWAY
 9456  063574  012703  064047                  MOV     #@0.BUF+6,R3    ;LWA
 9457  063600  004537  063652                  JSR     5,@0.TYPE       ;TYPE WHOLE STRING OF CHARACTERS
 9458  063604  012602                          MOV     (SP)+,R2        ;RESTORE R2
 9459  063606  000205                          RTS     R5
 9460                                    ;
 9461                                    ; GENERAL CHARACTER INPUT ROUTINE
 9462                                    ; CHARACTER INPUT GOES TO R0
 9463                                    ;
 9464  063610  105737  177560          0.GET:  TSTB    @0.RCSR         ;WAIT FOR
 9465  063614  100375                          BPL     .-4             ;    INPUT FROM KEYBOARD
 9466  063616  113700  177562                  MOVB    @0.RDB,R0       ;GET A CHARACTER
 9467  063622  004537  063666                  JSR     5,@0.FTYP       ;ECHO CHARACTER
 9468  063626  042700  177600                  BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
 9469  063632  001766                          BEQ     0.GET           ;IGNORE NULLS
 9470  063634  122700  000040                  CMPB    #40,R0          ;CHECK FOR SPACES
 9471  063640  001763                          BEQ     0.GET           ;IGNORE NULLS
 9472  063642  122700  000073                  CMPB    #';,R0          ;CHECK FOR SEMI-COLON
 9473  063646  001760                          BEQ     0.GET           ;IGNORE THEM IF FOUND
 9474  063650  000205                          RTS     R5
 9475                                    ;
 9476                                    ; GENERAL CHARACTER OUTPUT ROUTINE
 9477                                    ;  ADDRESS OF FIRST BYTE IN R4,
 9478                                    ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
 9479                                    ;
 9480  063652  020304                  0.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
 9481  063654  103426                          BLO     0.TYP1          ;  EXIT WHEN DONE
 9482  063656  112400                          MOVB    (R4)+,R0        ;GET A CHARACTER
 9483  063660  004537  063666                  JSR     5,@0.FTYP       ;TYPE ONE CHARACTER
 9484  063664  000772                          BR      0.TYPE          ;LOOP UNTIL DONE
```

```
 9485                                   ;
 9486                                   ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
 9487                                   ;
 9488  063666  105737  177564  O.FTYP: TSTB    O.TCSR          ;CHECK STATUS
 9489  063672  100375                  BPL     .-4             ;WAIT UNTIL READY
 9490  063674  110037  177566          MOVB    R0,O.TDB        ;TYPE ONE CHARACTER
 9491  063700  120037  000045          CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
 9492  063704  001012                  BNE     O.TYP1          ;NO
 9493  063706  113746  000044          MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
 9494  063712  105737  177564  O.TYP2: TSTB    O.TCSR
 9495  063716  100375                  BPL     O.TYP2
 9496  063720  105037  177566          CLRB    O.TDB           ;GENERATE NULL FILLER
 9497  063724  105316                  DECB    @SP
 9498  063726  003371                  BGT     O.TYP2
 9499  063730  005726                  TST     (SP)+           ;POP STACK
 9500  063732  000205          O.TYP1: RTS     R5
 9501                                   ;
 9502                                   ; CLOSE WORD OR BYTE AND EXIT,
 9503                                   ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
 9504                                   ;
 9505  063734  006205          O.TCLS: ASR     R5              ;GET LOW ORDER BIT
 9506  063736  103405                  BCS     O.TC            ;JUMP IF ALREADY CLOSED
 9507  063740  006305                  ASL     R5
 9508  063742  005702                  TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
 9509  063744  001401                  BEQ     O.CLS1          ;NO CHANGE TO THE OPEN CELL
 9510  063746  010415                  MOV     R4,@R5          ;STORE WORD
 9511  063750  000207          O.CLS1: RTS     PC
 9512  063752  005746          O.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
 9513  063754  000137  062256          JMP     O.ERR           ;AND SCREAM BLOODY MURDER
 9514                                   ;
 9515                                   ; O.CRLF - TYPE <CR,LF>
 9516                                   ; O.CRLS - TYPE <CR,LF>*
 9517                                   ;
 9518  063760  012703  064015  O.CRLF: MOV     #O.CR+1,R3      ;LWA <CR,LF>
 9519  063764  000402                  BR      O.CRS
 9520  063766  012703  064016  O.CRLS: MOV     #O.CR+2,R3      ;LWA <CR,LF>*
 9521  063772  012704  064014  O.CRS:  MOV     #O.CR,R4        ;FWA
 9522  063776  004537  063652          JSR     5,O.TYPE        ;TYPE SOMETHING
 9523  064002  000205                  RTS     R5
 9524                                   ;
 9525  064004  000000  O.XXX:  .WORD   0               ;TEMPORARY STORAGE
 9526  064006     000  O.T:    .BYTE   0               ;  T-BIT FLAG
 9527  064007     000  O.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
 9528                                                  ;               = 1 IF PROCEED ALLOWED
 9529  064010     000  O.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
 9530  064011     000  O.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
 9531                                   ;
 9532                                   .EVEN
 9533  064012  042502  O.BD:   .WORD   "BE
 9534                                   ;
 9535  064014     015  O.CR:   .BYTE   015     ;  <CR>
 9536  064015     012          .BYTE   012     ;  <LF>
 9537  064016     052          .BYTE   '*      ;  *
 9538                                   ;
 9539  064017     057  O.LGCH: .BYTE   '/      ;  /
 9540  064020     015          .BYTE   015     ;  CARRIAGE RETURN
```

```
 9541  064021  044                    .BYTE   '$      ;  $
 9542  064022  107                    .BYTE   'G      ;  G
 9543  064023  012                    .BYTE   012     ;  <LF>
 9544  064024  137                    .BYTE   '←      ;  ←
 9545  064025  136                    .BYTE   '↑      ;  ↑
 9546  064026  117                    .BYTE   'O      ;  O
 9547  064027  127                    .BYTE   'W      ;  W
 9548  064030  105                    .BYTE   'E      ;  E
 9549  064031  102                    .BYTE   'B      ;  B
 9550  064032  120                    .BYTE   'P      ;  P
 9551          000014          O.CLGT  =       .-O.LGCH'               ;TABLE LENGTH
 9552                                  ;
 9553  064033  123           Ò.TL:    .BYTE   'S      ;DO       1
 9554  064034  120                    .BYTE   'P      ;NOT      2
 9555  064035  115                    .BYTE   'M      ;CHANGE   3
 9556  064036  000                    .BYTE   0       ;THE      4
 9557  064037  000                    .BYTE   0       ;ORDER    5
 9558  064040  102                    .BYTE   'B      ;HERE     6
 9559          000006          O.LG    =       .-O.TL
 9560                                  ;
 9561  064041                 Ò.BUF:                          ;6 CHAR. BUFFER WITH
 9562          064047          .       =       ;⁴6
 9563  064047  040                    .BYTE                   ;TRAILING BLANK
 9564                                 .EVEN
 9565                                 ;
 9566  064050  000003         Ò.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
 9567                                 ;
 9568                                 ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
 9569                                 ;
 9570          061742                 .       =       O.ODT-40
 9571  061742  000000         Ò.URO:   0       ;USER R0
 9572  061744  000000                  0       ;      R1
 9573  061746  000000                  0       ;      R2
 9574  061750  000000                  0       ;      R3
 9575  061752  000000                  0       ;      R4
 9576  061754  000000                  0       ;      R5
 9577  061756  000000         O.USP:   0       ;USER SP
 9578  061760  000000         O.UPC:   0       ;USER PC
 9579  061762  000000         O.UST:   0       ;USER ST
 9580  061764  000007         O.PRI:   7       ;ODT PRIORITY
 9581  061766  000000         O.MSK:   0       ;MASK
 9582  061770  000000                  0       ;LOW LIMIT
 9583  061772  000000                  0       ;HIGH LIMIT
 9584                                 ;
 9585                                 ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
 9586                                 ;   UIN = CONTENTS
 9587                                 ;
 9588  061774  000000         Ò.ADR1: 0
 9589  061776  000000         O.CT:   0
 9590  062000  000000         O.UIN:  0
 9591          000001                 .END
```

```
ABASE = 177440           1489    1530    1554#
ACDW1 = 000000           1489    1532
ACDW2 = 000000           1489    1533
ACLO  = 000010           1185#
ACPUOP= 000000           1489    1504
ACT11   005522           1738#   2684*
ADDW0 = 000000           1489    1534
ADDW1 = 000000           1489    1535
ADDW10= 000000           1489    1544
ADDW11= 000000           1489    1545
ADDW12= 000000           1489    1546
ADDW13= 000000           1489    1547
ADDW14= 000000           1489    1548
ADDW15= 000000           1489    1549
ADDW2 = 000000           1489    1536
ADDW3 = 000000           1489    1537
ADDW4 = 000000           1489    1538
ADDW5 = 000000           1489    1539
ADDW6 = 000000           1489    1540
ADDW7 = 000000           1489    1541
ADDW8 = 000000           1489    1542
ADDW9 = 000000           1489    1543
ADEVCT= 000000           1489    1495
ADEVM = 000000           1489    1531
AENV  = 000000           1489    1500
AENVM = 000000           1489    1501
AFATAL= 000000           1489    1492
AMADR1= 000000           1489    1517
AMADR2= 000000           1489    1521
AMADR3= 000000           1489    1524
AMADR4= 000000           1489    1527
AMAMS1= 000000           1489    1511
AMAMS2= 000000           1489    1519
AMAMS3= 000000           1489    1522
AMAMS4= 000000           1489    1525
AMSGAD= 000000           1489    1497
AMSGLG= 000000           1489    1498
AMSGTY= 000000           1489    1491
AMTYP1= 000000           1489    1512
AMTYP2= 000000           1489    1520
AMTYP3= 000000           1489    1523
AMTYP4= 000000           1489    1526
APASS = 000000           1489    1494
APRIOR= 000000           1489
APTCSU= 000040           6222    6394#
APTENV= 000001           6169    6215    6350    6392#
APTSIZ= 000200           2612    6391#
APTSPO= 000100           6217    6352    6393#
ASWREG= 000000           1489    1502
ATESTN= 000000           1489    1493
ATTN    005376           1673#   5087    5107    5134
AUNIT = 000000           1489    1496
AUSWR = 000000           1489    1503
AVECT1= 000000           1489    1528
AVECT2= 000000           1489    1529
BADHDR  005372           1663#   2692*   5877
```

```
BADTMO  031370           2764    5932#
BAI     = 000020         1148#   4030    4080    4134    4166    4240    4332    4392    4505    4566    4653    4713
BA16    = 000400         1134#
BA17    = 001000         1135#
BIT0    = 000001         1077#   1131    1163    1182    1728    2802    4234    4324    4386    4560    4645    4707    5785
BIT00   = 000001         1067#   1077
BIT01   = 000002         1066#   1076
BIT02   = 000004         1065#   1075
BIT03   = 000010         1064#   1074
BIT04   = 000020         1063#   1073
BIT05   = 000040         1062#   1072
BIT06   = 000100         1061#   1071
BIT07   = 000200         1060#   1070
BIT08   = 000400         1059#   1069    6104
BIT09   = 001000         1058#   1068    6112    6180
BIT1    = 000002         1076#   1164    1729
BIT10   = 002000         1057#   1136    1154    1173    1205    1219    1233    1246    1260    6157
BIT11   = 004000         1056#   1137    1155    1174    1191    1206    1220    1234    1247    1261    6119
BIT12   = 010000         1055#   1138    1156    1175    1207    1221    1235    1248    1262
BIT13   = 020000         1054#   1139    1157    1176    1192    1208    1222    1236    1249    1263    6164
BIT14   = 040000         1053#   1140    1158    1177    1193    1209    1223    1237    1250    1264    1277    5595    6090
BIT15   = 100000         1052#   1141    1142    1159    1178    1194    1210    1280    5591
BIT2    = 000004         1075#   1165    1184    1730
BIT3    = 000010         1074#   1147    1166    1185
BIT4    = 000020         1073#   1148    1167    1186    1199    1227    1254
BIT5    = 000040         1072#   1149    1168    1187    1200    1214    1228    1241    1255
BIT6    = 000100         1071#   1132    1150    1169    1188    1201    1215    1229    1242    1256
BIT7    = 000200         1070#   1133    1151    1170    1189    1202    1216    1230    1243    1257    2792
BIT8    = 000400         1069#   1134    1152    1171    1190    1203    1217    1231    1244    1258
BIT9    = 001000         1068#   1135    1153    1172    1204    1218    1232    1245    1259
BPTVEC= 000014           1084#
BSE     = 000200         1170#   4041    4261    4515
BSERR   001542           1647#   3908*   3913*   3969    4455    4548    4632
BSE22H  002370           1657#   3861    3910    3931    5589    5689
BSE22S  003370           1658#   3920    5593    5693
BYP     025570           2879    2902    2906    2921    2973    2981    2999    3003    3007    3011    5161#
BYPCER  001550           1650#   2790*   3025*   3189*   5275
CALADD  001402           1584#   3995*   4474*   5562    5568    5570    5616
CALDIF  001374           1581#   8794    8804
CCLR    = 100000         1142#   3282    3301    3359    3368    3374    3413    3429    3444    3539    3586    3598    3682
                         3712    3818    4052    4126    4621    4827    4837    5317    5327    5337    5347    5880    5898
CCSP    030746           3193    3346    3495    3565    3642    3759    3978    4452    4811    4862    5814#   5823
CCYL    001370           1579#   5713
CDT     = 002000         1136#
CERR    = 100000         1141#   2859    2959    3649    3873    3950    4013    4038    4090    4143    4176    4249    4343
                         4402    4491    4513    4576    4664    4723    5277    5373
CFMT    = 010000         1138#   5684
CHKFLG  001552           1651#   5171*   5253    5256    5260
CHKMSG  025604           3215    3240    3294    3674    3805    3885    4065    4108    4155    4194    4361    4420    4526
                         4599    4682    4741    5171#
CKCERR  026336           5037    5041    5052    5057    5275#
CKSWR   = 104407         6089    6153    6179    7054#
CLEAR   = 000005         1119#   5900
CLKOF   030620           5756#
CLKON   030524           5734#
CLOCK   030564           2713    5745#
```

```
CLRFLG  024524          2634    4913#
CQE     = 001000        1172#
COUNT   001406          1588#   5746*   5748*
CR      = 000015         992#   6261    6271    7066    7071    7073    7078    7082    7092    7101    7108    7116    7121
                        7126    7131    7137    7143    7149    7156    7161    7168    7177    7182    7187    7191    7196
                        7200    7206    7208    7213    7218    7222    7224    7228    7232    7237    7242    7247    7253
                        7262    7271    7276    7282    7286    7292    7298    7307    7308    7314    7323    7333    7344
                        7355    7362    7369    7378    7385    7389    7398    7406    7414    7418    7422    7429    7434
                        7445    7452    7460    7467    7476    7482    7490    7495    7505    7510    7515    7522    7532
                        7542    7550    7555    7562    7573    7580    7589    7601    7609    7614    7621    7627    7632
                        7640    7645    7654    7664    7670    7677    7684    7692    7700    7706    7714    7718    7727
                        7733    7740    7745    7748    7751    7759    7766    7773    7791    7799    8127    8155    8192
                        8205    8220    8689    8700    8704
CRLF    = 000200         993#   6232    6271
CTO     = 004000        1137#   5281
CYL     001442          1605#
CYLADD  001400          1583#   3811    5412*   5413*   5414*   5415*   5416*   5417*   5435*   5436*   5437*   5438*   5439*
                        5440*
CYLDIF  001376          1582#   5396*   5398*   5399*   5400*   5401*   5402*   5403    5405*   5422*   5423*   5424*   5425*
                        5426*   5427*   5428    5430*   5491    5495    5498
DATAO   001524          1638#   4029    4079    4082    4165    4168    4236    4331    4388    4562    4652    4709
DATAO1  001526          1639#
DATAI   001530          1640#   4133    4239    4326    4391    4504    4565    4647    4712
DCK     = 100000        1178#
DCLO    = 000020        1186#
DCPAR   = 020000        1139#
DDISP   = 177570         999#   1464    2600
DDPCH   005520          1737#   2680*   2867    3054
DDT     = 000400        1190#
DDUMP   005516          1736#   2647*   4913
DF1     061102          1778    1784    1790    1796    1801    1807    1813    1819    1925    1988    2033    8839#
DF10    061172          1824    1829    1834    1839    1844    1849    1854    1859    1864    1869    1879    1900    1920
                        1945    1951    1957    1962    1967    1982    2013    2038    2053    2078    2108    2133    2138
                        2143    2148    2153    2158    2163    2168    2173    2178    2183    2188    2193    2198    2208
                        2228    2233    2238    2243    2268    2273    2278    2283    2288    2293    2298    2303    2308
                        2313    2318    2323    2328    2333    2338    2368    2393    2398    2413    2418    2423    2448
                        2453    2463    2468    2488    2538    8872#
DF12    061206          2473    2478    2483    8879#
DF14    061226          8888#
DF15    061236          2403    2408    2428    2433    2543    8894#
DF17    061252          2373    2383    8902#
DF20    061272          1993    8910#
DF21    061306          1874    1889    1894    1905    1910    1915    1930    1935    1940    1972    1977    1998    2003
                        2008    2018    2023    2028    2043    2048    2058    2063    2068    2073    2088    2093    2098
                        2103    2203    2213    2218    2223    2348    2353    2358    2363    2443    2493    2498    2503
                        2508    8917#
DF22    061342          2343    8932#
DF23    061376          8947#
DF3     061112          2248    2258    8844#
DF4     061126          2253    2263    8851#
DF5     061142          2438    8858#
DF6     061146          8861#
DF7     061162          2458    8868#
DH1     054012          1776    1782    1788    1794    1799    1805    1811    1817    1923    1986    2031    2456    8350#
                        8846    8853    8863    8874    8883    8896    8906    8912    8919    8934    8949
DH10    054512          8408#
```

```
DH11    054541          2051    2056    2061    2066    2071    8412#
DH12    054615          2076    8420#
DH13    054645          2106    8425#
DH14    054731          8434#
DH15    054762          2136    2331    8439#
DH16    055007          2151    2156    2421    8443#
DH17    055046          2011    2206    8449#
DH18    055072          1852    1857    1948    1954    1960    8453#
DH19    055117          1898    2176    8457#
DH2     054027          8353#   8841    8876    8885    8908
DH20    055142          2181    8461#
DH21    055176          1877    2186    8466#
DH22    055223          2311    2326    8470#
DH23    055255          2196    8475#
DH24    055307          8480#
DH25    055334          1991    2231    2236    2466    8484#
DH26    055357          1832    1837    1862    1892    1903    2006    2016    2536    2541    8488#
DH27    055407          1822    1827    1872    1887    1943    1980    1996    2001    2036    8493#
DH28    055440          2241    8498#
DH29    055474          2226    8503#
DH3     054100          1928    1933    1938    2441    8360#
DH30    055530          2391    2401    2406    8508#
DH31    055562          2266    2276    8513#
DH32    055616          1842    1847    1867    1908    1913    2021    2026    8519#
DH33    055650          8523#
DH34    055674          2281    2286    8527#
DH35    055730          2291    8532#
DH36    055761          8537#
DH37    056043          2296    2301    2306    8546#
DH38    056077          2271    2316    2321    8551#
DH39    056123          2426    2431    2491    2496    2501    2506    8555#
DH4     054126          1918    2086    2091    2096    2101    8364#
DH40    056156          8560#   8890
DH41    056233          2486    8568#
DH42    056260          2371    8572#
DH44    056331          8579#   8904
DH45    056410          2336    2341    2346    2351    2356    2361    2396    8587#
DH46    056447          2366    8593#
DH47    056514          8600#
DH48    056555          2146    8606#
DH49    056616          2171    8612#
DH5     054174          8371#
DH50    056656          2191    2201    2211    2216    2221    2246    2461    8618#
DH51    056724          8625#
DH52    056757          8630#
DH53    057013          8635#
DH54    057041          8639#
DH55    057070          8643#
DH56    057117          8647#   8914
DH57    057164          8654#
DH58    057224          8660#
DH59    057250          2141    2161    2166    8664#
DH6     054243          8378#   8865
DH60    057324          2251    8672#
DH61    057407          8681#   8848
DH62    057506          8693#   8855
```

```
DH63    057617          2256    2261    8706#
DH64    057670          2131    2411    8713#
DH65    057724          2416    8718#
DH66    057767          2436    8724#
DH67    060010          2446    2451    8727#
DH68    060053          8733#   8870
DH69    060115          8739#   8921    8936    8951
DH7     054326          8387#   8898
DH70    060176          8748#   8925    8940    8955
DH71    060257          8757#   8923    8927    8938    8942    8953    8957
DH72    060304          8761#   8929    8944    8959
DH73    060346          8767#   8881
DH74    060422          2381    8775#
DH8     054374          1965    1970    1975    2041    2046    8394#
DH9     054437          8400#
DI      = 040000        1140#
DISPLA  001142          1464#   2600*   2608*   6134*   6156*
DISPRE  000174          1289#   2608
DLT     = 100000        1159#
DLY     025552          4835    5152#   5155
DMD     = 000040        1200#
DOTIM   005554          1756#   2712*   2721*
DPAT1   001532          1641#
DPAT2   001534          1642#
DRA     = 000001        1182#   2896    2989
DRDY    = 000200        1189#
DRIVS   005526          1740#   2797*   2804*   2821    2843*   2869*   2886    2977*   3043    4771    4804    4968*
DRIVO   005530          1745#   2694    2799    2845    2945    4951
DRIV1   005532          1746#
DRIV2   005534          1747#
DRIV3   005536          1748#
DRIV4   005540          1749#
DRIV5   005542          1750#
DRIV6   005544          1751#
DRIV7   005546          1752#
DROT    = 000040        1187#
DRPAR   = 000010        1166#
DRVMSK= 000007          1146#   2862    2962
DRVPTR  001362          1573#   2694*   3048    3058*
DSC     = 040000        1193#
DSWR    = 177570         998#   1463    2599
DTE     = 010000        1175#
DTYE    = 000040        1168#
DT1     060506          1777    1783    1789    1795    1800    1806    1812    1818    1823    1828    1833    1838    1843
                        1848    1853    1858    1863    1868    1878    1899    1919    1924    1944    1950    1956    1961
                        1966    1981    1987    2012    2032    2037    2052    2077    2107    2132    2137    2142    2147
                        2152    2157    2162    2167    2172    2177    2182    2187    2192    2197    2207    2227    2232
                        2237    2242    2267    2272    2277    2282    2287    2292    2297    2302    2307    2312    2317
                        2322    2327    2332    2337    2367    2372    2382    2392    2397    2412    2417    2422    2447
                        2452    2462    2467    2472    2477    2482    2487    2537    8788#
DT10    060676          2457    8810#
DT13    060716          1873    1888    1893    1904    1909    1914    1929    1934    1939    1971    1976    1997    2002
                        2007    2017    2022    2027    2042    2047    2057    2062    2067    2072    2087    2092    2097
                        2102    2202    2212    2217    2222    2347    2352    2357    2362    2442    2492    2497    2502
                        2507    8813#
DT14    060756          2342    8820#
```

```
DT15    061026  8828#
DT3     060530  2247    2252    2257    2262    8791#
DT4     060552  8794#
DT5     060572  2402    2407    2427    2432    2542    8797#
DT6     060614  2437    8800#
DT7     060620  8801#
DT8     060640  8804#
DT9     060660  1992    8807#
D.ACLO= 000100  1242#   3665    3670    3703    4589    4595    4616
D.BRHM= 000100  1229#   3212    3237    3291    3671    3802    3879    4059    4102    4152    4188    4355    4414
                4520    4596    4676    4735
D.CART= 000400  1231#   3212    3237    3257    3291    3671    3802    3879    4059    4102    4152    4188    4355
                4414    4520    4596    4676    4735
D.DDT = 000400  1217#
D.DOOR= 000200  1230#   3206    3291    3589    3671    3802    3879    4059    4102    4152    4188    4355    4414
                4520    4596    4676    4735
D.DRA = 000040  1214#   3210    3235    3289    3800    3877    4057    4100    4150    4186    4353    4412    4518
                4594    4674    4733
D.DRDY= 000200  1216#   3289    3577    3698    3800    3877    4057    4100    4150    4186    4353    4412    4518
                4674    4733
D.DROT= 020000  1249#
D.DSC = 040000  1223#   3289    3355    3669    3800    4150    4594
D.FLT = 000200  1243#   3670    3795    3801    4151    4595
D.FORM= 001000  1218#
D.FWD = 002000  1233#
D.HDFL= 000200  1257#
D.HDHM= 000040  1228#   3212    3237    3601    3671    5517
D.IDAE= 000040  1241#
D.LIMD= 020000  1263#
D.LOAD= 010000  1235#   5535
D.MHD = 000400  1258#
D.NMOV= 010000  1262#
D.NXF = 000400  1244#   3790    3801
D.OFF = 002000  1219#
D.PAR = 001000  1245#   3351
D.PIP = 020000  1222#   4594
D.PLO = 004000  1261#
D.REV = 004000  1234#   4596
D.RTZ = 020000  1236#   5891
D.SECT= 000020  1254#
D.SKI = 002000  1246#
D.SPIN= 010000  1221#   3231    3271    3289    3573    3739    3800    3877    4057    4100    4150    4186    4353
                4412    4518    4594    4674    4733    5863
D.SPLS= 010000  1248#   4595
D.SPOK= 001000  1232#   3291    3802    3879    4059    4102    4152    4188    4355    4414    4520    4676    4735
D.SSP = 000020  1227#   3291    3802    3879    4059    4102    4152    4188    4355    4414    4520    4596    4676
                4735
D.SUNS= 040000  1264#
D.TIB = 002000  1260#
D.UNLD= 040000  1237#   4596
D.UNS = 040000  1250#
D.VV  = 000100  1215#   3124    3138    3210    3235    3285    3308    3427    3466    3669    3707    3719    3767
                3772    3825    3877    4057    4100    4150    4186    4353    4412    4518    4594    4628    4674
                4733
D.WCUR= 000040  1255#
D.WGAT= 000100  1256#
```

```
D.WLE = 004000           1247#   4146    4151    4269
D.WRL = 004000           1220#   3982    3989    4128    4150    4222    4460    4468
D.XERR= 001000           1259#
ECCW  = 020000           1208#
ECH   = 000100           1169#
EMTVEC= 000030           1087#   2583*   2584*
EM1     045700           4977    7773#
EM10    046454           1810    7839#
EM100   053724           8337#
EM101   053761           8342#
EM11    046537           1816    7848#
EM12    046603           1821    1831    1841    1851    2175    2180    2185    2205    2230    2280    2310    2390    2425
                         2535    7855#
EM13    046641           1856    2010    2160    2235    2285    7861#
EM14    046663           1947    7865#
EM15    046710           1953    7869#
EM16    046742           1959    7874#
EM17    046767           1927    1969    1995    2005    2020    2055    2085    2200    2345    2490    7878#
EM18    047010           1871    1891    1907    1974    2060    2090    2210    2350    2440    2495    7881#
EM19    047031           1932    2000    2015    2025    2040    2065    2095    2215    2355    2500    7884#
EM2     045754           1775    7781#
EM20    047052           1881    1902    1912    1937    2045    2070    2100    2220    2360    2505    7887#
EM21    047073           1826    1836    1866    1876    2405    2430    2465    2540    7890#
EM22    047115           1964    7894#
EM23    047156           2050    2105    7900#
EM24    047206           1897    7904#
EM25    047244           2075    7909#
EM26    047312           2420    7916#
EM27    047354           2155    2195    2240    7922#
EM28    047401           2135    7926#
EM29    047433           2140    7931#
EM3     046044           5936    7791#
EM30    047460           2150    7935#
EM31    047511           2225    7940#
EM32    047542           2265    7945#
EM33    047576           2270    7950#
EM34    047627           2275    7955#
EM35    047670           2290    7961#
EM36    047724           7966#
EM37    047767           2300    7972#
EM38    050017           7977#
EM39    050061           7983#
EM4     046117           7799#
EM40    050130           7990#
EM41    050166           7995#
EM42    050237           2305    8002#
EM43    050264           2315    8006#
EM44    050316           1917    2320    8011#
EM45    050360           2330    8017#
EM46    050435           2335    2365    8025#
EM47    050514           2340    8033#
EM48    050554           2145    8039#
EM49    050614           2170    8045#
EM5     046173           1781    2480    7807#
EM50    050650           2190    2450    8050#
EM51    050712           2245    2250    2255    2260    8056#
```

```
EM52    050765          2130    8064#
EM53    051012          2410    8069#
EM54    051042          2415    8073#
EM55    051112          2325    8080#
EM56    051145          2435    8085#
EM57    051211          2295    2445    2460    8092#
EM58    051237          2165    8096#
EM59    051267          2470    8101#
EM6     046214          1787    7810#
EM60    051310          8104#
EM61    051346          2395    8109#
EM62    051374          8113#
EM63    051414          2400    8116#
EM64    051435          2455    8119#
EM65    051562          8134#
EM66    051604          8137#
EM67    051640          2475    8142#
EM68    051661          2370    2380    8145#
EM69    051724          2030    8151#
EM7     046235          1793    7813#
EM70    052022          2471    2476    2481    8162#
EM71    052113          1942    8172#
EM72    052163          1979    8179#
EM73    052242          1985    8187#
EM74    052361          2485    8201#
EM75    052406          2035    8205#
EM76    052462          1922    8213#
EM8     046324          1798    7823#
EM80    052565          1846    8226#
EM81    052624          8232#
EM82    052670          8238#
EM83    052742          1861    8245#
EM84    052777          8250#
EM85    053035          8256#
EM86    053100          8262#
EM87    053133          8267#
EM88    053203          8274#
EM89    053230          8278#
EM9     046400          1804    7831#
EM90    053270          8284#
EM91    053327          8290#
EM92    053354          8294#
EM93    053405          1990    8299#
EM94    053450          8305#
EM95    053517          8312#
EM96    053553          8317#
EM97    053601          8321#
EM98    053642          8327#
EM99    053673          8332#
ENDRV   024006          4115    4201    4368    4427    4689    4748    4754#   9035
ERRVEC= 000004          1080#   2597    2598*   2609*   2617*   2618*   2631*   2632*   2697*   2701*   2709*   2746*   2764*
                        6095    6096*   6098*   6101*
ESEC    001420          1594#
E.A0    005476          1717#   3210*   3235*   3289*   3669*   3671*   3800*   3877*   4057*   4100*   4150*   4186*   4353*
                        4412*   4518*   4594*   4674*   4733*   5175*   5182    5184*   5213    8813    8820    8829
E.A1    005502          1719#   3212*   3237*   3291*   3802*   3879*   4059*   4102*   4152*   4188*   4355*   4414*   4520*
```

```
                    4596*  4676*  4735*  5176*  5186   5188*  5233   8813   8820   8828
E.A2     005506     1721*  3881*  4061*  4104*  4190*  4357*  4416*  4522*  4678*  4737*  5177*  5190   5192*
                    8820   8828
E.A3     005512     1723*  5178*
E.B0     005500     1718*  3211*  3236*  3290*  3670*  3801*  3878*  4058*  4101*  4151*  4187*  4354*  4413*
                    4519*  4595*  4675*  4734*  5194   5196*  5223   8813   8820   8828
E.B1     005504     1720*  3213*  3238*  3292*  3672*  3803*  3880*  4060*  4103*  4153*  4189*  4356*  4415*
                    4521*  4597*  4677*  4736*  5198   5200*  5243   8813   8820   8828
E.B2     005510     1722*  3882*  4062*  4105*  4191*  4358*  4417*  4523*  4679*  4738*  5202   5204*  8820
                    8828
E.B3     005514     1724*  3883*  4063*  4106*  4192*  4359*  4418*  4524*  4680*  4739*  5206   5208*  8828
FATT1    025376     3749   5104*  5873   5895
FATT2    025472     3787   3947   5132*
FCYL     027542     5489*
FHDHM    027632     5514*  5523
FHDTAB   027754     3999   4478   5548*
FLGTST   030204     5590   5594   5609*
FLOAD    027706     5532*
FMTE   = 000020     1167*
FMT1     001520     1635*  5558*  5559*  5560*  5565
FORM     031370     5910   5922*
FORMAT   001516     1634*  3997*  4476*  5558   5587
FRCYL    001364     1577*  8794   8804
FRDY     025062     2836   2852   2857   2952   2957   3135   3160   3305   3319   3417   3433   3448   3614
                    3716   3745   3783   3822   3870   3943   4010   4035   4087   4139   4173   4246   4340
                    4399   4488   4510   4572   4625   4661   4720   4831   4841   5032*  5035   5307   5368
                    5869   5886   5902
FRDY1    025130     5046*  5049   5322   5332   5342   5351
FSEC17   027452     5465*
FTITLE   001360     1571*  4924   4926*
GBA      024764     2666   4991*
GDRVS    024624     2664   4946*
GETSP    031006     3198   3203   3228   3253   3268   3279   3366   3383   3474   3500   3510   3521   3530
                    3545   3570   3584   3595   3607   3693   3764   3986   4124   4220   4256   4275   4465
                    4551   4583   4611   4816   4823   5831*  5837
GINT     025012     2668   5004*
GNS    = ****** U   1288   7046   7047   7048   7049   7050   7052   7054   7055   7056   7057   7058   7059
                    7060
GO     = 000001     1131*
GSTAT    026420     3088   3205   3230   3256   3270   3284   3350   3360   3369   3375   3465   3502   3522
                    3540   3572   3588   3600   3648   3683   3766   3872   3981   3988   4012   4037   4089
                    4127   4142   4175   4221   4342   4401   4459   4467   4490   4512   4575   4663   4722
                    5114   5119   5139   5143   5303*  5372   5383   5395   5411   5516   5534   5890
GSTAT1   026462     5173   5315*
GTSWR  = 104406     4935   7052*
HASOF    005424     1689*  5071*  5087   8788   8817   8825   8834
HBA      005414     1685*  5067*
HCS1     005406     1682*  2859   2959   3649   3873   3950   4013   4038   4090   4143   4176   4249   4343
                    4402   4491   4513   4576   4664   4723   5064*  5277   5281   5373   5684   8788   8791
                    8797   8810   8817   8825   8834
HCS2     005410     1683*  2892   2894   2898   2910   2985   2987   2991   2994   4092   4178   4345   4404
                    4666   4725   4844   5065*  5286   5290   8788   8791   8797   8810   8817   8825   8834
HDA      005416     1686*  4277   4278   4606   4607   5068*  5721   8797   8801   8810
HDB      005430     1691*
HDC      005426     1690*  5072*  8794   8804   8807   8810   8817   8825   8834
HDS      005420     1687*  2896   2989   5069*  8788   8797   8817   8825   8834
```

```
HDTAB   001554          1653#   4001    4480    5550
HDWD    001540          1645#   8801
HD1     001514          1633#   5552*   5553*   5554*   5555*   5556*   5557*   5564
HEAD    001510          1631#   3996*   4003*   4004    4005*   4019*   4020    4475*   4482*   4483    4484*   4496*   4497
                        5552    5621
HEADA   001512          1632#   5455*   5456*   5457*   5458*
HER     005422          1688#   4041    4261    4515    5070*   8788    8797    8817    8825    8834
HMR1    005432          1692#   5073*
HMR2    005434          1693#   2861    2961    3089    3124    3138    3206    3231    3257    3271    3285    3308    3355
                        3427    3466    3573    3577    3589    3601    3698    3707    3719    3739    3767    3772    3825
                        3982    3989    4128    4222    4460    4468    4628    5050*   5055*   5074*   5324    5334    5344
                        5353    5396    5517    5535    5863    5891    8788    8791    8794    8797    8801    8804    8807
                        8810
HMR3    005436          1694#   3351    3665    3703    3790    3795    4146    4269    4583    4616    5051*   5056*   5075*
                        5325    5335    5345    5354    5384    5412    8788    8791    8794    8797    8801    8804    8807
                        8810
HOLD    025216          5036    5040    5064#   5316
HPAT    005442          1696#   5077*
HPEND   005374          1665#   2693*   5879*
HPOS    005440          1695#   5076*
HT    = 000011           990#   6230    6271
HVRC  = 000400          1171#
HWC     005412          1684#   5066*
HZ      001404          1586#   5748
H.A0    005456          1706#   5213    5353*   8813    8823    8831
H.A1    005462          1708#   5233    5344*   8813    8823    8831
H.A2    005466          1710#   5334*   5422    8823    8831
H.A3    005472          1712#   5324*
H.B0    005460          1707#   5223    5354*   8813    8823    8831
H.B1    005464          1709#   5243    5345*   8813    8823    8831
H.B2    005470          1711#   5335*   5435    8823    8831
H.B3    005474          1713#   5325*   5445    5455    8831
IDAE  = 002000          1173#
IE    = 000100          1132#
ILF   = 000001          1163#
INTER   031520          5019    5976#
IOTVEC= 000020          1085#   2581*   2582*
IR    = 000100          1150#
LCLKF   005550          1754#   2699*
LCVEC   001350          1562#   2700
LF    = 000012           991#   6265    6271    7066    7071    7073    7078    7082    7092    7101    7108    7116    7121
                        7126    7131    7137    7143    7149    7156    7161    7168    7177    7182    7187    7191    7196
                        7200    7206    7208    7213    7218    7222    7224    7228    7232    7237    7242    7247    7253
                        7262    7271    7276    7282    7286    7292    7298    7307    7308    7314    7323    7333    7344
                        7355    7362    7369    7379    7385    7389    7398    7406    7414    7418    7422    7429    7434
                        7445    7452    7460    7467    7476    7482    7490    7495    7505    7510    7515    7522    7532
                        7542    7550    7555    7562    7573    7580    7589    7601    7609    7614    7621    7627    7632
                        7640    7645    7654    7664    7670    7677    7684    7692    7700    7706    7714    7718    7727
                        7733    7740    7745    7748    7751    7759    7766    7773    7791    7799    8127    8155    8192
                        8205    8220    8689    8700    8704
LIMERR  001544          1648#
LKS     001346          1560#   2698    5737*   5758*
LPCNT   001466          1617#
LPTIM   001470          1618#
MCLK  = 000400          1203#
MDS   = 001000          1153#   2892    2985    4844    5290
```

```
MDSERR  001546          1649#
MEMBAS= 172100          1566#   2620    2629
MEMERR  031456          2623    5962#
MEMPAR  010360          2617#
MEMVEC= 000114          1565#   2623*   2624*
MERD   = 001000         1204#
MESMSK= 000017          1198#
MEWD   = 002000         1205#
MIND   = 000200         1202#
MSG1    036220          4927    7066#
MSG10   037523          2796    2842    7196#
MSG100  045531          4282    4287    4292    4297    7751#
MSG11   037550          6009    7200#
MSG12   037610          7206#
MSG13   037617          2722    7208#
MSG14   037705          5161    7218#
MSG15   037730          2912    3059    7222#
MSG16   037742          3086    7224#
MSG17   037766          3930    7228#
MSG18   040016          3344    7232#
MSG19   040050          3493    7237#
MSG2    037123          2648    7149#
MSG20   040101          3563    7242#
MSG21   040132          3639    7247#
MSG22   040173          3757    7253#
MSG23   040254          7262#
MSG24   040341          3976    7271#
MSG25   040365          7275#
MSG26   040371          4449    7276#
MSG27   040432          3045    7282#
MSG28   040462          4809    7286#
MSG29   040524          3741    5865    7292#
MSG3    037174          2663    7156#
MSG30   040570          3201    7298#
MSG31   040652          3408    5822    5836    7307#
MSG32   040660          3251    7308#
MSG33   040720          3225    7314#
MSG34   041005          3226    3266    7323#
MSG35   041073          3265    7333#
MSG36   041170          3276    7344#
MSG37   041265          3197    3202    3227    3252    3267    3365    3382    3473    3499    3509    3520    3529    3544
                        3583    3594    3763    3985    4123    4219    4255    4274    4464    4550    4815    4822    7355#
MSG38   041331          3385    7362#
MSG39   041467          3386    3459    3469    7379#
MSG4    037226          2665    7161#
MSG40   041532          3468    7385#
MSG41   041644          3498    7398#
MSG42   041724          3508    7406#
MSG43   042002          3519    7414#
MSG44   042032          3528    7418#
MSG45   042062          3543    7422#
MSG46   042133          3568    3605    7429#
MSG47   042336          3196    3582    7452#
MSG48   042416          3593    7460#
MSG49   042613          3644    4545    7482#
MSG5    037273          2667    7168#
```

```
MSG50   042666          3690    4581    4609    7490#
MSG51   042720          3762    7495#
MSG52   043041          3192    3345    3494    3564    3640    3756    3977    4450    4810    7510#
MSG53   043142          3984    4218    4254    4273    4463    7522#
MSG54   043233          4122    4226    7532#
MSG55   043322          3387    3453    7542#
MSG56   043402          3912    7550#
MSG57   043606          4546    7573#
MSG58   043654          4814    7580#
MSG59   043736          4821    7589#
MSG6    037354          5981    7177#
MSG60   044041          4860    7601#
MSG61   044151          4852    7614#
MSG62   044217          4806    7621#
MSG63   044313          4861    7632#
MSG64   044506          4204    7654#
MSG65   044637          3200    3250    3275    7670#
MSG66   044705          7677#
MSG67   045030          3381    7692#
MSG68   045105          4858    7700#
MSG69   045145          3278    3569    3606    3692    4582    4610    7706#
MSG7    037410          2682    7182#
MSG70   045222          3277    3691    7714#
MSG71   045250          3472    7718#
MSG72   045333          4071    4114    4200    4367    4426    4688    4747    7727#
MSG74   045367          5855    7733#
MSG75   045435          5857    7740#
MSG76   045472          5914    7745#
MSG77   045514          5918    7748#
MSG8    037445          3191    7187#
MSG9    037471          3364    7191#
MSP     =  000100       1201#
MULT6   030370          5641    5668#
M.ALGN=  040000         1277#
M.CADD=  017760         1276#   5413    5436
M.CDIF=  017760         1269#   5398    5423
M.DPID   031642         6024#
M.DP40   031700         6035#   6057
M.DP41   031734         6040    6048#
M.DP42   031744         6046    6052#
M.DP44   031776         6059    6065#
M.DP50   032010         6033    6070#
M.DRV =  000007         1268#
M.HEAD=  007000         1279#   5456
M.ID  =  000003         1275#
M.OFST=  017760         1270#
M.PAR =  100000         1280#   5787
M.SECT=  000760         1278#   5385    5446
M.SER =  077770         1271#
NED     =  010000       1156#   2898    2910    2991    2994    3376    3420    3451    3503    3515    3523    3535    3541
                        3684    5286
NEM     =  004000       1155#
NUDRV    012276         2823    2924    3025#   4773
NXF     =  000004       1165#
OFFERR   001506         1628#
OFFSET=  000015         1123#
```

```
OFST  = 000004          1184#
OPI   = 020000          1176#
OR    = 000200          1151#
0.ADR1  061774          9094*   9145*   9151*   9326    9327*   9365*   9380    9397    9588#
0.BACK  062520          9194    9235#
0.BD    064012          9384    9385    9533#
0.BKPT  062226          9140#   9198
0.BK1   063160          9088    9362#
0.BK2   063214          9367    9370#
0.BK3   063232          9369    9376#
0.BRK   063150          9105    9360#
0.BUF   064041          9442    9451    9456    9561#
0.B2    063314          9381    9390#
0.B3    063354          9389    9398#
0.CADV  063524          9211    9228    9251    9255    9289    9293    9398    9441#
0.CLGL  062342          9167    9169    9177#
0.CLGT= 000014          9181    9551#
0.CLS1  063750          9509    9511#
0.CR    064014          9518    9520    9521    9535#
0.CRET  062454          9189    9218#
0.CRLF  063760          9226    9287    9383    9422    9427    9518#
0.CRLS  063766          9162    9520#
0.CRS   063772          9519    9521#
0.CSR1  064010          9418*   9434    9529#
0.CSR2  064011          9419*   9435    9530#
0.CT    061776          9347*   9390*   9392*   9589#
0.C1    063126          9350#   9391
0.DCD   062272          9147    9152    9162#   9220    9275    9400
0.DCD1  062276          9163#   9212    9256
0.EFF   062602          9197    9266#
0.EFF1  062720          9278    9297#
0.ENTR  062006          9085#
0.ERR   062256          9132    9143    9159#   9182    9208    9270    9513
0.ERR1  062612          9270#   9343
0.ERR2  062432          9208#   9242    9319
0.FTYP  063666          9161    9230    9245    9291    9396    9467    9483    9488#
0.GET   063610          9111    9165    9464#   9469    9471    9473
0.GO    062764          9191    9317#
0.GO2   063042          9328#   9353
0.LG  = 000006          9115    9559#
0.LGCH  064017          9178    9539#   9551
0.LGDR  062372          9185    9188#   9200
0.LGL = 000030          9200#
0.LGL1  062344          9178#   9183
0.LGL2  062364          9179    9184#
0.MSK   061766          9271    9272    9274    9581#
0.ODT   062002          1297    9083#   9570
0.OFST  062530          9195    9241#
0.OF1   062576          9254    9256#
0.OP1   062466          9192    9224#
0.OP2   062474          9136    9226#   9237
0.ORPC  062200          9128#   9193
0.P     064007          9103*   9317*   9342    9344*   9393*   9527#
0.PRI   061764          9095    9366    9370    9580#
0.PROC  063072          9199    9342#
0.PR1   063114          9346    9348#
```

```
0.RALL  062246          9106    9141    9151#
0.RCSR= 177560          9070#   9418    9420*   9430    9432    9434*   9464
0.RDB = 177562          9069#   9466
0.REGT  062130          9111#   9190
0.RSE1  063506          9431    9434#
0.RSP   062140          9113#   9116
0.RST   062044          9084    9093#
0.RSTT  063456          9323    9349    9427#
0.RST1  062074          9092    9100#
0.RTIT  063070          9102*   9337#
0.SCAN  062302          9122    9165#   9176
0.SP    062172          9114    9123#
0.SPC   063536          9444#   9452
0.SP1   062160          9119#   9124
0.STM = 000340          9059#   9104    9322    9348    9350
0.STRT  062032          9083    9090#
0.SVR   063366          9093    9362    9404#
0.SVTT  063424          9382    9394    9418#
0.T     064006          9324*   9351*   9363    9526#
0.TBIT  063014          9324#   9364
0.TBT = 000020          9060#   9325    9352    9388
0.TC    063752          9506    9512#
0.TCLS  063734          9128    9218    9224    9235    9505#
0.TCSR= 177564          9072#   9419    9421*   9428    9435*   9488    9494
0.TDB = 177566          9071#   9490*   9496*
0.TL    064033          9112    9115    9123    9553#   9559
0.TRTC  064050          9151    9327    9566#
0.TVEC= 000014          9058#   9086    9104*   9105*
0.TYPE  063652          9386    9457    9480#   9484    9522
0.TYP1  063732          9481    9492    9500#
0.TYP2  063712          9494#   9495    9498
0.UIN   062000          9094    9326*   9365    9590#
0.UPC   061760          9087*   9321*   9336    9360*   9377    9379*   9578#
0.URO   061742          9090    9120    9571#
0.USP   061756          9091*   9405*   9406    9577#
0.UST   061762          9085*   9325*   9335    9352*   9361*   9368    9388*   9579#
0.WDS   062610          9267    9269#
0.WDS2  062626          9274#   9296
0.WDS3  062656          9285#   9298    9304    9313
0.WDS4  062714          9285    9295#
0.WRD   062422          9188    9204#
0.WRDA  062446          9205    9213#
0.WRD1  062430          9207#   9214    9231
0.WSCH  062606          9196    9268#
0.XXX   064004          9404*   9412    9525#
0.45    062106          9101    9103#
PACK  = 000003          1118#   3133    3303    3431    3714    3820    4623
PARAM   001356          1570#   2547*   2550*   2660
PARSRT  010050          1294    2547#
PAT   = 000020          1199#   3349
PCA   = 004000          1206#
PCD   = 010000          1207#
PCLKF   005552          1755#   2703*   2710*   5735    5756
PCVEC   001352          1563#   2704    2711
PCYL    001372          1580#
PFSRT   012434          3069#   6011
```

```
PGE    = 002000          1154#
PIP    = 020000          1192#
PIRQ   = 177772           997#
PIRQVE= 000240           1091#
PKRB     001344          1559#
PKS      001340          1557#  2702   2709   5740*  5760*
PKSB     001342          1558#  5739*
PPTP     005524          1739#  2651*
PRGSRT   010064          2548   2551#  4978
PRO    = 000000          1014#  2553   2741
PR1    = 000040          1015#
PR2    = 000100          1016#
PR3    = 000140          1017#
PR4    = 000200          1018#
PR5    = 000240          1019#  1556
PR6    = 000300          1020#  2714
PR7    = 000340          1021#  2618   2624   6006   6007
PS     = 177776           994#   995
PSEC     001416          1593#
PSW    = 177776           995#
PWRVEC= 000024          1086#  2587*  2588*  5996*  6005*  6006*
RCYLA    027330          5258   5435#
RCYLD    027256          5255   5422#
RDCHR  = 104410          5815   5832   6673   7055#
RDCYLA   027206          5410#
RDCYLD   027122          5394#  5490   5494   5497
RDDATA= 000021           1125#  3868
RDGATE= 100000           1210#
RDHEAD= 000025           1127#
RDLIN  = 104411          4946   6747   7056#
RDOCT  = 104412          4991   5004   7057#
RDSEC    027052          5382#  5466   5470   5473   5639
RDTAB    004370          1659#
RDY    = 000200          1133#  5032   5046
RECAL  = 000013          1122#  5882
RESREG= 104414           6807   6859   7059#  9029
RESVEC= 000010           1081#
RHEAD    027424          5263   5455#
RHTAB    001760          1654#  5646   5648   5653   5656   8807
RKASOF= 000016           1104#  2755   5071   5107   5134
RKBA   = 000004          1099#  2751   3862*  3903*  3921*  4001*  4029*  4079*  4133*  4165*  4236*  4239*  4326*
                         4331*  4335   4388*  4391*  4394   4480*  4504*  4562*  4565*  4647*  4652*  4656   4709*
                         4712*  4715   5067
RKCS1  = 000000          1097#  2748   2855*  2955*  3133*  3158*  3282*  3301*  3303*  3317*  3359*  3368*  3374*
                         3413*  3415*  3429*  3431*  3444*  3446*  3539*  3586*  3598*  3612*  3682*  3712*  3714*
                         3743*  3781*  3818*  3820*  3868*  3941*  4008*  4033*  4052*  4085*  4126*  4137*  4171*
                         4244*  4338*  4397*  4486*  4508*  4570*  4621*  4623*  4659*  4718*  4827*  4829*  4837*
                         4839*  5032   5046   5064   5305*  5317*  5320*  5327*  5330*  5337*  5340*  5347*  5349*
                         5867*  5880*  5882*  5898*  5900*
RKCS2  = 000010          1101#  2749   2829*  2850*  2854*  2950*  2954*  3302*  3376   3414*  3420   3430*  3445*
                         3451   3503   3515   3523   3535   3541   3684   3713*  3819*  4030*  4080*  4134*  4166*
                         4240*  4332*  4392*  4505*  4566*  4622*  4653*  4713*  4828*  4838*  5065   5304*  5318*
                         5328*  5338*  5348*  5366*  5370*  5881*  5899*
RKDA   = 000006          1100#  2752   3864*  3905*  3923*  4004*  4027*  4083*  4136*  4169*  4242*  4302*  4307*
                         4311*  4313   4334*  4376*  4380*  4384*  4385   4483*  4568*  4655*  4697*  4701*  4705*
                         4706   5068
```

```
RKDB   = 000024  1106#  2757  4094  4180  4347  4406  4668  4727
RKDC   = 000020  1105#  2756  3780*  3866*  4243*  4336*  4395*  4507*  4569*  4657*  4716*  5072
RKDS   = 000012  1102#  2753  5069
RKECPS= 000030  1110#  2761  5076
RKECPT= 000032  1111#  2762  5077
RKER   = 000014  1103#  2754  5070
RKMR1  = 000026  1107#  2758  3087*  3204*  3255*  3283*  3349*  3587*  3599*  5073  5113*  5118*  5138*
                 5142*  5319*  5329*  5339*  5371*  5382*  5394*  5410*  5515*  5533*  5889*
RKMR2  = 000034  1108#  2759  5050  5055  5074
RKMR3  = 000036  1109#  2760  5051  5056  5075
RKPR1    001336  1556#  5020
RKVEC    001334  1555#  2687*  5008*  5011*  5018
RKWC   = 000002  1098#  2750  3867*  4002*  4031*  4081*  4135*  4167*  4241*  4333*  4393*  4481*  4506*
                 4567*  4654*  4714*  5066
RLS    = 000010  1147#
RSEC     027366  5262  5445#
RTT    = 000006  9062#
SAVREG= 104413  6792  6833  7058#  8971
SBPAR    030642  5183  5187  5191  5195  5199  5203  5207  5773#
SCLR   = 000040  1149#  2829  2850  2950  5366
SCOP1  = 104415  2847  2947  3127  7060#
SCOP1$   030714  5797#  7060
SDC    = 000000  7065
SEC      001410  1590#  5749*
SECFLG   030134  5566  5586#
SECNT    001414  1592#
SECTOR   001422  1595#  4026*  4027  4046*  4047  4083  4136  4169  5384*  5385*  5386*  5387*  5388*
                 5389*  5445*  5446*  5447*  5448*  5449*  5450*  5467  5471  5474  5640*  5644  5645*
                 5646*  5658  5668*  5669  5670*  5671*
SEEK   = 000017  1124#  3781  3941
SELDRV= 000001  1117#  2855  2955  3415  3446  4829  4839  5305  5320  5330  5340  5349
SETINT   025044  2688  5009  5018#
SIZFLG   005556  1757#  2689*  2840  4964*  4984*
SKI    = 000002  1164#
SORT     030262  5637#
SPACE2   061716  9007  9016  9027  9037#
SPBAR  = 000040  1576#  5817  5834
SRTSPL= 000011  1121#  3743  5867
SRTTAB   002164  1655#  5649
ST     = 177776  9056#  9085  9086*  9099*  9322*  9348*  9350*  9376*
STACK  = 001100   985#  2552  2579  2739  2788  2848  2943  2948  3039  3079  3119  3128  3151
                 3184  3339  3488  3558  3634  3734  3853  3968  4444  4768  4798  5268  5971
                 6008  9034
START    010060  1292  2550#
START1   010466  2642#
STKLMT= 177774   996#
STOP     031034  5846#  6514  6597
STOP1    031036  3407  5848#
ST2      010534  2649  2660#
ST3      010600  2650  2678#
ST4      010632  2662  2683  2686#
ST5      010656  2669  2691#  2827  2890  4903  5920
SUBCLR   026772  3083  3121  3130  3155  3186  3222  3247  3262  3314  3341  3462  3490  3512
                 3532  3560  3609  3636  3695  3736  3754  3777  3815  3856  3991  3937  3973
                 4023  4076  4119  4162  4215  4231  4317  4446  4501  4542  4557  4613  4638
                 4818  4868  5366#  5848  5860
```

```
SUM     001472          1620#
SUM1    001476          1622#
SVAL  = 100000          1194#
SWR     001140          1463#  2577    2599*   2601    2607*   2614*   4933    5219    5229    5239    5249    5797    5934
                        5942   5947    5967    6090    6104    6106    6112    6119    6157    6164    6176    6180    6517
                        6556   6611*   8972    9030
SWREG   000176          1290#  2607    4933    6517    6556    6579
SW0   = 000001          1048#
SW00  = 000001          1039#  1049
SW01  = 000002          1038#  1048
SW02  = 000004          1037#  1047
SW03  = 000010          1036#  1046
SW04  = 000020          1035#  1045
SW05  = 000040          1034#  1044
SW06  = 000100          1033#  1043
SW07  = 000200          1032#  1042
SW08  = 000400          1031#  1041
SW09  = 001000          1030#  1040
SW1   = 000002          1048#
SW10  = 002000          1029#
SW11  = 004000          1028#
SW12  = 010000          1027#  9030
SW13  = 020000          1026#  5934    8972
SW14  = 040000          1025#  5947
SW15  = 100000          1024#
SW2   = 000004          1047#
SW3   = 000010          1046#
SW4   = 000020          1045#
SW5   = 000040          1044#
SW6   = 000100          1043#
SW7   = 000200          1042#
SW8   = 000400          1041#
SW9   = 001000          1040#  5219    5229    5239    5249    5797    5942    5967
TBITVE= 000014          1082#
TEMP1   005444          1698#  2835*   2851*   2856*   2861*   2862*   2863    2951*   2956*   2961*   2962*   2963    3134*
                        3159*  3304*   3318*   3416*   3432*   3447*   3613*   3647*   3651*   3681*   3686*   3715*   3744*
                        3782*  3786*   3821*   3869*   3942*   3946*   4009*   4034*   4086*   4138*   4172*   4245*   4339*
                        4398*  4487*   4509*   4571*   4624*   4660*   4719*   4830*   4834*   4840*   5034*   5048*   5105*
                        5109*  5136*   5152    5154*   5180    5182*   5184    5186*   5188    5190*   5192    5194*   5196
                        5198*  5200    5202*   5204    5206*   5208    5210*   5303    5306*   5309*   5315    5321*   5331*
                        5341*  5350*   5356*   5367*   5465*   5477*   5489*   5501*   5514*   5519*   5532*   5537*   5568*
                        5571*  5572    5779*   5787*   5868*   5885*   5901*   8801
TEMP2   005446          1699#  3646*   3653*   3654    3748*   3857*   3893*   3894    3919*   5111*   5491    5495    5498
                        5522*  5569*   5570*   5572*   5573    5872*   5894*
TEMP3   005450          1700#  3858*   3896    3916*   3917
TEMP4   005452          1701#  3861*   3862    3903    3920*   3921
TEMP5   005454          1702#  3863*   3864    3904*   3905    3922*   3923
TERR1   030464          5690   5694    5711#   5716    5723
TIMUP   001412          1591#  5734*   5745*   5751*
TIM1    001456          1612#
TIM2    001460          1613#
TIM3    001462          1614#
TIM4    001464          1615#
TITLE   024544          2644   2681    4924#
TKVEC = 000060          1089#  6494*   6495*
TOCYL   001366          1578#  3994*   4229*   4243    4263*   4264    4336    4395    4473*   4499*   4507    4534*   4535
```

```
                    4569    4657    4716    8794    8804    8807
TPVEC = 000064      1090#
TRAPPC  001354      1568#   5963*   8800
TRAPVE= 000034      1088#   2585*   2586*   6007*
TRT   = 000003      9061#   9566
TRTVEC= 000014      1083#
TRUERR  030412      4043    5682#
TSTATN  025344      3162    3321    3361    3370    3616    5085#   5904
TST1    011036      2715    2737#
TST10   012752      3182#
TST11   013630      3337#
TST12   014500      3486#
TST13   014756      3536    3542    3556#
TST14   015262      3632#
TST15   015666      3732#
TST16   016332      3851#
TST17   017056      3909    3966#
TST2    011212      2765    2786#
TST20   022016      4442#
TST21   024006      4766#
TST22   024042      4772    4796#
TST3    011752      2841    2941#
TST4    012302      2971    3037#
TST5    012434      3077#
TST6    012572      3082    3117#
TST7    012674      3125    3139    3149#
TYPDS = 104405      4892    7050#
TYPE  = 104401      2648    2663    2665    2667    2682    2722    2796    2806    2842    2871    2912    3045    3059
                    3067    3086    3108    3109    3191    3192    3196    3197    3200    3201    3202    3225    3226
                    3227    3250    3251    3252    3265    3266    3267    3275    3276    3277    3279    3344    3345
                    3364    3365    3381    3382    3385    3386    3387    3393    3408    3440    3453    3459    3468
                    3469    3472    3473    3493    3494    3498    3499    3508    3509    3519    3520    3528    3529
                    3543    3544    3563    3564    3568    3569    3582    3583    3593    3594    3605    3606    3639
                    3640    3644    3690    3691    3692    3741    3757    3758    3762    3763    3912    3930    3934
                    3935    3976    3977    3984    3985    4071    4114    4122    4123    4200    4204    4218    4219
                    4226    4254    4255    4273    4274    4282    4287    4292    4297    4367    4426    4449    4450
                    4463    4464    4545    4546    4550    4581    4582    4609    4610    4688    4747    4806    4809
                    4810    4814    4815    4821    4822    4852    4858    4860    4861    4890    4893    4927    4977
                    5161    5822    5836    5855    5857    5865    5914    5918    5936    5981    6009    6159    6167
                    6235    6329    6455    6511    6523    6577    6578    6581    6592    6602    6613    6632    6680
                    6686    6691    6695    6700    6701    6703    6706    6710    6776    6778    6918    7046#   8983
                    8984    8988    8989    8998    9007    9011    9016    9018    9024    9027
TYPERR  061432      6166    8971#
TYPOC = 104402      6580    7047#   9004
TYPON = 104404      7049#
TYPOS = 104403      2809    2874    2915    3063    3390    3437    3456    4855    5164    5939    5984    7048#
T.A2  = 000001      1728#   3806    4066    4109    4195    4362    4421    4527    4683    4742    5253
T.B2  = 000002      1729#   3806    4066    4109    4195    4362    4421    4527    4683    4742    5256
T.B3  = 000004      1730#   5260
T1      001424      1597#
T10     001426      1598#   2835    2851    2856    2951    2956    3159    3304    3318    3416    3432    3447    3613
                    3715    3744    3782    3821    4624    4830    4840    5306    5321    5331    5341    5350    5367
                    5868    5885    5894    5901
T100    001434      1601#   3748    5872
T50     001430      1599#   3942
T500    001432      1600#
```

```
T5000   001436          1602#   4227    5465    5489
T50000  001440          1603#   3681    3786    3869    3946    4009    4034    4086    4138    4172    4245    4339    4398
                        4487    4509    4554    4571    4660    4719
UFE   = 000400          1152#   2894    2987
UNLD    005370          1661#   2691*   3153*   3280*   3312*   3581*   3597*   3702*   3752*   4593*   4620*   4859*   4864*
                        4866*   5851
UNLOAD= 000007          1120#   3158    3317    3612
UNS   = 040000          1177#
UPE   = 020000          1157#
VV    = 000100          1188#
WCE   = 040000          1158#   4092    4178    4345    4404    4666    4725
WDCNT   001522          1636#   8801
WD1     001502          1625#   4094*   4180*   4347*   4406*   4668*   4727*   8791
WD2     001504          1626#   4082*   4168*   4335*   4394*   4656*   4715*   8791
WLE   = 004000          1174#
WORD    001536          1644#
WRDATA= 000023          1126#   4033    4137    4244    4508    4570
WRHEAD= 000027          1128#   4008    4486
WRL   = 004000          1191#
WRTCHK= 000031          1129#   4085    4171    4338    4397    4659    4718
WRTGAT= 040000          1209#
$APTHD  001000          1319    1325#
$ASTAT= ****** U        6372    6387
$ATYC   033232          6343    6345#
$ATY1   033206          6341#
$ATY3   033214          6220    6342#
$ATY4   033224          6172    6344#
$AUTOB  001134          1460#   4937*   6570    6727
$BASE   001264          1530#   2686*   2747    4995*   4997*
$BDADR  001122          1455#
$BDDAT  001126          1457#
$BELL   001200          1481#   6159    6192    6523    6720
$CDW1   001270          1532#
$CDW2   001272          1533#
$CHARC  032756          6237*   6247*   6254    6263*   6268#
$CKSWR  034200          6556#   7054
$CMTAG  001100          1443#   2574    2575    2583    2589    2590    2591
$CM3  = 000000          1473#
$CM4  = 000006          1473#   1474#   1475#   1476#   1477#   1478#   1479#
$CNTLC  035110          6511    6592    6720#
$CNTLG  035122          6577    6722#
$CNTLU  035115          6602    6695    6721#
$CPUOP  001236          1504#
$CRLF   001205          1483#   2806    2871    3067    3108    3109    3393    3440    3934    3935    6167    6192    6236
                        6271    6613    6700    6720    6781    8983    8988    8998    9011    9024
$DBLK   033176          6295    6329    6337#
$DB2D   035432          6833#   6895
$DB2O   035312          3932    6792#
$DDW0   001274          1534#
$DDW1   001276          1535#
$DDW10  001320          1544#
$DDW11  001322          1545#
$DDW12  001324          1546#
$DDW13  001326          1547#
$DDW14  001330          1548#
$DDW15  001332          1549#
```

```
$DDW2   001300          1536#
$DDW3   001302          1537#
$DDW4   001304          1538#
$DDW5   001306          1539#
$DDW6   001310          1540#
$DDW7   001312          1541#
$DDW8   001314          1542#
$DDW9   001316          1543#
$DECVL  035612          6835    6881#
$DEVCT  001220          1495#   2635*   2695*   3049    4770*   4771
$DEVM   001266          1531#   2800
$DOAGN  024500          4886    4895    4901#
$DTBL   033166          6298    6333#
$ENDAD  024470          1305    4897#   6187
$ENDCT  024436          2589    4888#
$ENDMG  024507          4890    4905#
$ENULL  024504          4893    4904#
$ENV    001230          1500#   4931    6169    6215    6350    6374
$ENVM   001231          1501#   2612    2792    6217    6222    6352
$EOP    024402          2769    3046    4072    4878#   5916    5958
$EOPCT  024430          2589*   4885#   4889    5915*
$ERFLG  001103          1446#   5799    6079    6108    6110    6116*   6138    6154*   6192    9032
$ERMAX  001115          1452#   2592*   6110    6133*   6138
$ERROR  032300          2583    6152#
$ERRPC  001116          1453#   6161*   6162*   6163    6192    8788    8791    8794    8797    8801    8804    8807    8810
                        8813    8820    8828
$ERRTB  005560          1772#   8980
$ERTTL  001112          1450#   6160*   6192
$ESCAP  001176          1480#   2591*   5211    5215*   5225*   5235*   5245*   5265*   5269*   5962*   5966*   6132*   6183
                        6185    6192
$ETABL  001230          1499#
$ETEND  001334          1331    1552#
$FATAL  001212          1492#   6378*
$FFLG   033452          6341*   6344*   6372    6381*   6389#
$FILLC  001156          1471#   6240    6271
$FILLS  001155          1470#   6271
$GDADR  001120          1454#
$GDDAT  001124          1456#
$GET42  024460          4894#
$GTSWR  034270          6578#   7052
$HD     =  000000        961
$HIBTS  001000          1326#
$HIOCT  035310          6769*   6780#
$ICNT   001104          1447#   6123*   6124    6126*   6137
$INTAG  001135          1461#   6575*   6594    6614    6727
$ITEMB  001114          1451#   6163*   6171    6192    8974
$LF     001206          1484#   6192    6271    6710    6720    6781
$LFLG   033451          6382*   6388#
$LPADR  001106          1448#   2593*   5945    6114*   6130*   6135    6137
$LPERR  001110          1449#   2594*   5270    5801    5804*   5972    6114    6131*   6137    6182
$MADR1  001242          1517#
$MADR2  001246          1521#
$MADR3  001252          1524#
$MADR4  001256          1527#
$MAIL   001210          1327    1331    1490#   2611    4931    6129    6169    6215
$MAMS1  001240          1511#
```

```
$MAMS2  001244          1519#
$MAMS3  001250          1522#
$MAMS4  001254          1525#
$MBADR  001002          1327#
$MFLG   033450          6342*   6348    6383*   6387#
$MNEW   035140          6581    6725#
$MSGAD  001224          1497#   6358*   6361
$MSGLG  001226          1498#   6363*
$MSGTY  001210          1491#   6356    6364*   6376    6380*
$MSWR   035127          6578    6723#
$MTYP1  001241          1512#
$MTYP2  001245          1520#
$MTYP3  001251          1523#
$MTYP4  001255          1526#
$MULT   035722          6937#
$MXCNT  032276          6127    6137#
$NULL   001154          1469#   6242    6271
$NWTST= 000001          2727#   2729    2772#   2774    2926#   2928    3028#   3030    3070#   3072    3111#   3113    3142#
                        3144    3168#   3170    3325#   3327    3478#   3480    3548#   3550    3620#   3622    3725#   3727
                        3831#   3833    3957#   3959    4433#   4435    4756#   4758    4775#   4777
$OCNT   033676          6427*   6456*   6469#
$OCTVL  035414          3090    6794    6819#
$OMODE  033700          6422*   6426*   6431    6434*   6445*   6471#
$OVER   032262          6091    6107    6115    6125    6134#
$PASS   001216          1494#   2611*   3081    3926    4800    4882*   4883*   4891    4904    6121    6138
$PASTM  001006          1329#
$PWRCT  031640          6002*   6003*   6013#
$PWRDN  031554          2587    5996#   6005
$PWRUP  031566          5996    6002#
$QUES   001204          1482#   6192    6271    6632    6703    6720    6778    6781
$RDCHR  034542          6645#   7055
$RDDEC= ****** U        7058
$RDLIN  034632          6668#   7056
$RDOCT  035152          6742#   7057
$RDSZ = 000022          6661#
$RESRE  036072          7003#   7059
$RTNAD  024502          4903#
$R2A  = ****** U        7060
$SAVRE  036034          6987#   7058
$SB2D   035626          6893#
$SCOPE  032020          2581    6088#
$SETUP= 000137          2544#   2580    2581    2583    2585    2587    2589    2590    2591    2593    4880    4928    6089
                        6153    6179    6187    6515    6520    6521    6551    6727
$STUP = 177777          2544#
$SUPRS  035662          3107    3933    6910#
$SVLAD  032226          6099    6128#
$SVPC = 000244          1303#   1308
$SWR  = 167400           947#    961     965     966     967     968     969     970     971     972    1479    1480    1481
                        2590    2591    2593    2594    2738    2787    2942    3038    3078    3118    3150    3183    3338
                        3487    3557    3633    3733    3852    3967    4443    4767    4797    4875    4881    4896    4902
                        4904    6080    6081    6082    6083    6084    6090    6102    6104    6105    6108    6109    6110
                        6117    6118    6119    6131    6134    6137    6144    6145    6146    6147    6148    6157    6164
                        6176    6180    6192
$SWREG  001232          1502#   2614
$SWRMK= 000000           972     973    6084    6085    6106
$TESTN  001214          1493#   3040*   6129*   8788    8791    8794    8797    8800    8801    8804    8807    8810    8813
```

```
                        8820    8828
$TIMES  001174          1479#   2590*   2738*   2787*   2942*   3038*   3078*   3118*   3150*   3183*   3338*   3487*   3557*
                        3633*   3733*   3852*   3967*   4443*   4767*   4797*   4881*   6117*   6124    6127*   6137
$TKB    001146          1466#   5814    5831    6475    6496    6507    6532    6560    6587
$TKCNT  033702          6476#   6491*   6521    6538*   6652    6654*
$TKINT  033712          2557    3406    3410    3443    6491#   6512    6573
$TKQEN= 033711          6480#   6546    6657
$TKQIN  033704          6477#   6492*   6493    6544*   6545*   6546    6548*
$TKQOU  033706          6478#   6493*   6655    6656*   6657    6659*
$TKQSR  033710          3395*   3398    3409*   6479#   6492    6548    6659
$TKS    001144          1465#   6475    6497*   6528*   6530    6536*   6558    6574*   6584    6596*   6616*
$TKSRV  033762          6494    6507#
$TMPO   001160          1473#   3396*   3398*   3399*   3400    3402    3404    4227*   4251*   4554*   4578*
$TMP1   001162          1474#   4228*   4234    4259*   4324    4386    4555*   4560    4586*   4645    4707
$TMP2   001164          1475#   4277*   4280    4285    4290    4295    4300    4305    4310*   4311    4320    4322    4374
                        4378    4383*   4384    4606*   4641    4643    4695    4699    4704*   4705
$TMP3   001166          1476#   4278*   4313*   4334    4385*   4607*   4655    4706*   8791
$TMP4   001170          1477#
$TMP5   001172          1478#   5211*   5265    5269
$TN     = 000023          948#    961   2715    2727    2738#   2765    2772    2787#   2841    2926    2942#   2971    3028
                        3038#   3040    3041    3070    3078#   3082    3111    3118#   3125    3139    3142    3150#   3168
                        3183#   3325    3338#   3478    3487#   3536    3542    3548    3557#   3620    3633#   3725    3733#
                        3831    3852#   3909    3957    3967#   4433    4443#   4756    4767#   4772    4775    4797#
$TNPWR  035542          6840    6841    6861#
$TPB    001152          1468#   6260*   6271
$TPFLG  001157          1472#   6209    6271
$TPS    001150          1467#   6258    6271
$TRAP   036130          2585    7023#
$TRAP2  036152          7034#   7045
$TRP    = 000016        7038#   7047#   7048#   7049#   7050#   7051#   7052    7053#   7054    7055#   7056#   7057#   7058#
                        7059#   7060#   7061#
$TRPAD  036164          7028    7045#
$TSTM   001004          1328#
$TSTNM  001102          1445#   3041*   4880*   6079    6106    6128*   6129    6134    6138    6156    6192
$TTYIN  035066          6670    6671    6683    6701    6715    6719#
$TYPBN= ****** U        7051
$TYPDS  032762          6283#   7050
$TYPE   032500          6209#   6369    7038    7046
$TYPEC  032712          6239    6246    6253    6258#   6259    6618
$TYPEX  032760          6264    6266    6269#
$TYPOC  033500          6425#   7047
$TYPON  033514          6424    6427#   7049
$TYPOS  033454          6420#   7048
$UNIT   001222          1496#   2636*   2696*   3051*   3056    3060    3302    3388    3445    3454    3713    3819    4622
                        5086    5106    5133    5175    5176    5177    5178    5304    5318    5328    5338    5348    5370
                        5881    5899
$UNITM  001010          1330#
$USWR   001234          1503#
$VECT1  001260          1528#
$VECT2  001262          1529#
$XTSTR  032032          6093#
$$GET4= 000000          4896#
$OFILL  033677          6421*   6425*   6435    6470#
$40CAT= ****** U        6090    6166
.       = 062002        1284#   1288#   1293#   1296#   1303    1304#   1306#   1308#   1309#   1315    1316#   1318#   1320#
                        1442#   1485    1653#   1654#   1655#   1657#   1658#   1659#   2578    2593    2594    4904    4908
```

```
                    5998    6054    6137    6138    6192    6271    6337#   6390#   6475    6479#   6480    6481#   6719#
                    6720    6727#   6781    6819#   6881#   9046#   9047#   9200    9429    9433    9465    9489    9551
                    9559    9562#   9570#
.$ASTA= ****** U    6342    6345
.$X   = 001000      1315#   1320
```

```
CALIB    1364#  5880
CHECK    1355#  3214  3239  3293  3673  3804  3884  4064  4107  4154  4193  4360  4419  4525  4598
                4681  4740
COMMEN   1092#
CWD2     1361#
DRCLR    1362#  5897
ENDCOM   1092#
ERROR     986#  2768  2825  2837  2853  2858  2880  2888  2903  2907  2918  2922  2953  2958  2974
                2982  3000  3004  3008  3012  3084  3122  3131  3136  3140  3156  3161  3163  3187  3208
                3217  3218  3219  3220  3223  3233  3242  3243  3244  3245  3248  3259  3263  3273  3287
                3296  3297  3298  3299  3306  3310  3315  3320  3322  3342  3353  3357  3362  3372  3378
                3418  3434  3449  3463  3491  3505  3513  3517  3525  3533  3537  3561  3575  3579  3591
                3603  3610  3615  3617  3637  3656  3667  3676  3677  3678  3679  3688  3696  3700  3705
                3709  3717  3721  3737  3746  3750  3755  3769  3774  3778  3784  3788  3792  3797  3807
                3808  3809  3810  3813  3816  3823  3827  3856  3871  3875  3887  3888  3889  3890  3892
                3898  3900  3938  3944  3948  3952  3974  3991  4011  4015  4024  4036  4049  4055  4067
                4068  4069  4070  4073  4077  4088  4095  4098  4110  4111  4112  4113  4120  4130  4140
                4145  4148  4157  4158  4159  4160  4163  4174  4181  4184  4196  4197  4198  4199  4216
                4232  4247  4253  4266  4271  4318  4341  4348  4351  4363  4364  4365  4366  4400  4407
                4410  4422  4423  4424  4425  4447  4470  4489  4493  4502  4511  4528  4529  4530  4531
                4537  4543  4558  4573  4580  4591  4601  4602  4603  4604  4614  4618  4626  4630  4639
                4662  4669  4672  4684  4685  4686  4687  4721  4728  4731  4743  4744  4745  4746  4819
                4832  4842  4849  4869  5283  5288  5292  5308  5323  5333  5343  5352  5369  5849  5861
                5870  5874  5887  5893  5896  5903  5906  5964
ESCAPE   1092#
F.EAB    1345#  3876  4056  4099  4185  4352  4411  4517  4673  4732
GETPRI   1092#
GETSWR   1092#  4928
HDCHK3   1377#
HDTBL    1381#
LOOP     1339#  2846  2946  3127
MSG      2727#  2729  2772#  2774  2926#  2928  3028#  3030  3070#  3072  3111#  3113  3142#  3144  3168#
                3170  3325#  3327  3478#  3480  3548#  3550  3620#  3622  3725#  3727  3831#  3833  3957#  3959
                4433#  4435  4756#  4758  4775#  4777
MULT     1092#
NEWTST   1092#  2727  2772  2926  3028  3070  3111  3142  3168  3325  3478  3548  3620  3725  3831
                3957  4433  4756  4775
OWNTAG   1333#  1554
POP      1092#  6324  6384  6385  6770  6966  7008
PUSH     1092#  6283  6345  6347  6368  6744  6937  6988
QKPACK   1426#  3301  3711  3818  4621
QKSEEK   1385#  3940
QKSRT    1422#  3735  5858
QKUNLD   1430#  3154  3313  3608
RDATA    1402#
RDHDR    1375#
REPORT   1092#
SBOUND   1418#  4316  4636
SCOPE     987#  2737  2786  2941  3037  3077  3117  3149  3182  3337  3486  3556  3632  3732  3851
                3966  4442  4766  4796  4879
SETPRI   1092#  6648
SETTRA   7038#  7047  7048  7049  7050  7052  7054  7055  7056  7057  7058  7059  7060
SETUP    1092#  2573
SKIP     1092#  2715  2765  2841  2971  3082  3125  3139  3536  3542  3909  4772
SLASH    1092#
SPACE    1092#
```

```
STARS    1092#  1301   1312   1314   1321   1438   1485   1488   2727   2736   2772   2785   2926   2940   3028
         3036   3070   3076   3111   3116   3142   3148   3168   3181   3325   3336   3478   3485   3548   3555
         3620   3631   3725   3731   3831   3850   3957   3965   4433   4441   4756   4765   4775   4795   4872
         6076   6140   6194   6273   6340   6397   6474   6551   6566   6637   6661   6730   6783   6822   6884
         6902   6925   6972   7017   8961   8970
SWRSU    1092#  2595#
TRMTRP   7038#
TYPBIN   1092#
TYPDEC   1092#  4891
TYPNAM   1092#
TYPNUM   1092#
TYPOCS   1092#  2807   2872   2913   3061   3388   3435   3454   4853   5162   5937   5982
TYPOCT   1092#  6579
TYPTXT   1092#
WDATA    1395#  4032
WRCHK    1411#  4084   4170   4337   4396   4658   4717
WRHDR    1370#  4007   4485
$$CMRE   1436#
$$CMTM   1436#  1473   1474   1475   1476   1477   1478
$$ESCA   1092#
$$NEWT   1092#  2727   2772   2926   3028   3070   3111   3142   3168   3325   3478   3548   3620   3725   3831
         3957   4433   4756   4775
$$SET    7038#  7047   7048   7049   7050   7052   7054   7055   7056   7057   7058   7059   7060
$$SETM   2611#
$$SKIP   1092#  2715   2765   2841   2971   3082   3125   3139   3536   3542   3909   4772
.EQUAT    947#   982
.HEADE    947#   951
.SETUP    947#  2544
.SWRHI    947#   961
.SWRLO    947#   973#
.$ACT1    947#  1299
.$APTB   1486#
.$APTH    947#  1310
.$APTY    947#  6338
.$CATC    947#  1282
.$CMTA    947#  1436
.$DB2D    947#  6820
.$DB2O    947#  6781
.$EOP     947#  4870
.$ERRO    947#  6138
.$MULT    947#  6923
.$RDOC    947#  6728
.$READ    947#  6472
.$SAVE    947#  6970
.$SB2D    947#  6882
.$SCOP    947#  6074
.$SUPR    947#  6900
.$TRAP    947#  7015
.$TYPD    947#  6271
.$TYPE    947#  6192
.$TYPO    947#  6395

. ABS.   064052     000
```

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

 DZR6JC,DZR6JC.SEQ/SOL/NL:MD/EQ:SDC/CRF/NL:TOC/DOC=DZR6JC.P11
 RUN-TIME: 76 69 9 SECONDS
 RUN-TIME RATIO: 236/156=1.5
 CORE USED:  31K  (61 PAGES)

 DOCUMENT PAGES:  206
```

Spooler runtime 32 Seconds, 135 KCS, 827 disk reads, 3 disk writes, 206 pages

[illegible] Date 01-Dec-76 17:40:39 Monitor IPC-D 602 [6B0] [illegible]