RK06

DISK DRIVE DIAG PART 3

MD-11-DZR6J-B

EP-DZR6J-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA

```
.REM    %
```

IDENTIFICATION

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

TABLE OF CONTENTS

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

CHAIN MODE OPERATION (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

DUMP MODE OPERATION (MANUAL)

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

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE. IT IS APT
COMPATABLE TO THE EXTENT THAT APT HOOKS WILL BE IN THE

PROGRAM & WILL WORK THRU THE 'UPTON INTERFACE'.

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

AUTOMATIC MODE (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

NOTE: SEVERAL APT CONSIDERATIONS ARE STILL TO BE DEFINED.

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

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

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

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

FAULT ISOLATION WILL NOT BE PERFORMED FOR THE FIRST STAGE
RELEASE BUT WILL BE INCLUDED FOR THE SECOND STAGE RELEASE.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OF 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	1260	177440
CONTROLLER INTERRUPT VECTOR	1330	210
CONTROLLER PRIORITY	1332	240
P-CLOCK STATUS REG	1334	172540
P-CLOCK SET BUFFER	1336	172542
P-CLOCK READ BUFFER	1340	172544
L-CLOCK STATUS REG	1342	177546
L-CLOCK INTERRUPT VECTOR	1344	100
P-CLOCK INTERRUPT VECTOR	1346	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564
TTY PRINTER BUFFER	1152	177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

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

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

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

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

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

4.1.4 PRESS 'START'

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

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

4.2 STARTING LOCATIONS

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

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED

C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

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

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

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

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

4.3.1 SW<15>

THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT THE ERROR MESSAGE AND PERTINENT INFORMATION, 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 ERROR'S HAVE
BEEN DETECTED.

4.3.5 SW<11>

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

4.3.6 SW<10>

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

4.3.7 SW<09>

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

4.3.8 SW<08>

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

4.4 'SOFTWARE' SWITCH REGISTER

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

```
482
483
484                                 4.5.3   CONTROLLER INTERRUPT VECTOR
485
486                                         THE REQUEST WILL BE:
487
488                                                 TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210
489
490                                         THE DEFAULT IS A CARRIAGE RETURN.
491
492
493                                 4.5.4   EXAMPLE OF PROGRAM DIALOGUE
494
495                                         THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220.
496                                         ALL OPERATOR RESPONSES ARE UNDERLINED.
497
498                                                 UNIBUS RK06 DRIVE DIAGNOSTIC
499                                                    PART 3
500                                                 MAINDEC-11-DZR6J-B-PB
501
502                                                 DRIVES TO BE TESTED: 1,3<CR>
503                                                                      -------
504
505                                                 TYPE IN BUSS ADDRESS IF NOT 177440 <CR>
506                                                                            ----
507
508                                                 TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>
509                                                                                           ----
510
511                                                 WILL TEST DRIVES:
512                                                 1
513                                                 3
514
515                                                 DRIVE 1
516
517                                                 (THE REST IS IDENTICAL TO THE EXAMPLE SHOWN IN 4.6 BELOW)
518
519
520                         4.6     PROGRAM EXAMPLE
521
522                                 THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE
523                                 DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.
524
525                                                 UNIBUS RK06 DRIVE DIAGNOSTIC
526                                                    PART 3
527                                                 MAINDEC-11-DZR6J-B-PB
528
529                                                 WILL TEST DRIVES:
530                                                 0
531                                                 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.


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-C (↑C). 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  UNLOAD ALL DRIVES TO BE TESTED

        THIS TEST UNLOADS ALL THE DRIVES TO BE TESTED,
        WAITS FOR ATTN & VERIFIES IT CAME BACK FROM THE CORRECT DRIVE
        WITHIN APPROX. 100 USEC.
        ATTN, DRIVE READY & DRIVE STATUS CHANGES ARE VERIFIED TO BE CORRECT


TEST 5  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 6  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 7  SET VV WITH PACK COMMAND

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


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


TEST 10 INTERLOCKS TESTS

        THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
        ARE OPERATING PROPERLY IN MESSAGE A0 & 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.
```

```
732
733                                     TEST 16 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
734
735                                             THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
736                                             THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
737                                             FACTORY DURING MANF.  ALL BAD SECTOR INFOR (BSE) WILL BE STORED
738                                             AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
739                                             IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
740                                             IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
741                                             A MESSAGE WILL BE TYPED INDICATING THAT ALL
742                                             FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
743                                             THIS IS DONE SO AS NOT TO DESTROY BSE INFOR OR AN ALIGNMENT PACK BY WRIT
744
745                                             THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
746
747
748                                     TEST 17 WRITE LOCK TEST
749
750                                             THIS TEST VERIFIES THAT DATA WRITTEN ON A SECTOR CANNOT BE
751                                             ALTERED ONCE THE WRITE LOCK SWITCH HAS BEEN ENABLED.
752                                             IT ALSO CHECKS THAT WRITE PROTECT TAKES EFFECT ONLY AT
753                                             SECTOR BOUNDRIES WHEN DOING CONTINUOUS WRITING ON CYL 0, HEAD 0
754
755
756                                     TEST 20 AC LOW DETECTION PART 2
757
758                                             THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
759                                             AT SECTOR BOUNDRIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
760                                             THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
761                                             WHEN THE INTERFACE SHUTS DOWN.
762
763
764
765
766                                     TEST 21 END OF PROGRAM
767
768                                             THIS IS NOT A TEST BUT A LINKAGE TO PERFORM ALL THE
769                                             ABOVE TESTS FOR THE NEXT DRIVE PRESENT.
770                                             THE NEXT TEST IS ENTERED ONLY AFTER ALL DRIVES PRESENT
771                                             HAVE BEEN TESTED.
772                                             DO NOT LOOP ON THIS 'TEST'.
773
774
775                                     TEST 22 MULTIPLE DRIVE DETECTION TEST
776
777                                             THIS TEST IS PERFORMED ONLY ONCE AT THE END OF PASS 1
778                                             AND IS BYPASSED IF ONLY ONE DRIVE IS PRESENT.
779
780                                             THE MULTIPLE DRIVE DETECTION LOGIC IS TESTED BY THE FOLLOWING METHOD:
781
```

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

```
MESSAGE A0 ERROR
AT END OF HEAD LOADING

  TEST NO.  PC
  000014  013432
RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2
140141  100000  000000  140101  040200  000101
  SHOULD BE
  150341

MESSAGE B1 ERROR
AFTER LIMIT DETECT

  TEST NO.  PC
  000023  023316
RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2
045720  030001  000000  040000  040200  000100
  SHOULD BE
  045720  120001
```

IN THE FIRST EXAMPLE, RKMR2 (MESSAGE A0) DID NOT READ BACK CORRECTLY. THE CORRECT "SHOULD BE" CONTENTS IS UNDER 'RKMR2'.

IN THE 2ND EXAMPLE, RKMR3 (MESSAGE B1) DID NOT READ BACK CORRECTLY WITH THE CORRECT 'SHOULD BE' CONTENTS UNDER 'RKMR3'.

```
                                        %


                                .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
```

```
                        SWD1=   2
                        SWD0=   1
                        .EQUIV  SWD9,SW9
                        .EQUIV  SWD8,SW8
                        .EQUIV  SWD7,SW7
                        .EQUIV  SWD6,SW6
                        .EQUIV  SWD5,SW5
                        .EQUIV  SWD4,SW4
                        .EQUIV  SWD3,SW3
                        .EQUIV  SWD2,SW2
                        .EQUIV  SWD1,SW1
                        .EQUIV  SWD0,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
```

```
1020
1021                            ;       $BASE=177440
1022
1023    000000                  RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1024    000002                  RKWC=   2               ;WORD COUNT REGISTER
1025    000004                  RKBA=   4               ;BUS ADDRESS REGISTER
1026    000006                  RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1027    000010                  RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1028    000012                  RKDS=   12              ;DRIVE STATUS REGISTER
1029    000014                  RKER=   14              ;ERROR REGISTER
1030    000016                  RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1031    000020                  RKDC=   20              ;DESIRED CYLINDER REGISTER
1032    000024                  RKDB=   24              ;DATA BUFFER
1033    000026                  RKMR1=  26              ;MAINTENANCE REGISTER 1
1034    000034                  RKMR2=  34              ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1035    000036                  RKMR3=  36              ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1036    000030                  RKECPS= 30              ;ECC POSITION INFORMATION
1037    000032                  RKECPT= 32              ;ECC PATTERN INFORMATION
1038
1039                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1040
1041                            ;       DRIVE COMMANDS
1042
1043    000001                  SELDRV= 1               ;SELECT DRIVE (GET STATUS)
1044    000003                  PACK=   3               ;PACK ACKNOWLEDGE
1045    000005                  CLEAR=  5               ;DRIVE CLEAR
1046    000007                  UNLOAD= 7               ;UNLOAD
1047    000011                  SRTSPL= 11              ;START SPINDLE
1048    000013                  RECAL=  13              ;RECALIBRATE
1049    000015                  OFFSET= 15              ;OFFSET
1050    000017                  SEEK=   17              ;SEEK
1051    000021                  RDDATA= 21              ;READ DATA
1052    000023                  WRDATA= 23              ;WRITE DATA
1053    000025                  RDHEAD= 25              ;READ HEADER
1054    000027                  WRHEAD= 27              ;WRITE HEADER AND DATA
1055    000031                  WRTCHK= 31              ;WRITE CHECK
1056
1057    000001                  GO=     BIT0            ;GO BIT
1058    000100                  IE=     BIT6            ;INTERRUPT ENABLE
1059    000200                  RDY=    BIT7            ;CONTROLLER READY
1060    000400                  BA16=   BIT8            ;BUS ADDRESS BIT 16
1061    001000                  BA17=   BIT9            ;BUS ADDRESS BIT 17
1062    002000                  CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06)
1063    004000                  CTO=    BIT11           ;CONTROLLER TIMEOUT
1064    010000                  CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1065    020000                  SPAR=   BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1066    040000                  DI=     BIT14           ;DRIVE INTERRUPT
1067    100000                  CERR=   BIT15           ;CONTROLLER ERROR
1068    100000                  CCLR=   BIT15           ;CONTROLLER CLEAR
1069
1070                            .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1071
1072    000007                  DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
1073    000010                  RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1074    000020                  BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
1075    000040                  SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
1076            000100          IR=     BIT6                    ;INPUT READY
1077            000200          OR=     BIT7                    ;OUTPUT READY
1078            000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1079            001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1080            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1081            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1082            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1083            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1084            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1085            100000          DLT=    BIT15                   ;DATA LATE ERROR
1086
1087                            .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1088
1089            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1090            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
1091            000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
1092            000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
1093            000020          FMTE=   BIT4                    ;FORMAT ERROR
1094            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1095            000100          ECH=    BIT6                    ;ECC HARD
1096            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
1097            000400          HVRC=   BIT8                    ;HEADER VRC ERROR
1098            001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1099            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1100            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1101            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
1102            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1103            040000          UNS=    BIT14                   ;DRIVE UNSAFE
1104            100000          DCK=    BIT15                   ;DATA CHECK
1105
1106                            .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1107
1108            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1109                                                            ;   THIS BIT IS RESET)
1110            000004          OFST=   BIT2                    ;DRIVE OFFSET
1111            000010          ACLO=   BIT3                    ;AC LOW
1112            000020          DCLO=   BIT4                    ;DC LOW
1113            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
1114            000100          VV=     BIT6                    ;VOLUME VALID
1115            000200          DRDY=   BIT7                    ;DRIVE READY
1116            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
1117            004000          WRL=    BIT11                   ;WRITE LOCK
1118            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
1119            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
1120            100000          SVAL=   BIT15                   ;STATUS VALID
1121
1122                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1123
1124            000017          MESMSK= 17                      ;MESSAGE MASK
1125            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
1126            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
1127            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1128            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
1129            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
1130            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1131            002000          MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
```

```
1132          004000                  PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
1133          010000                  PCD=    BIT12                   ;PRECOMPENSATION DELAY
1134          020000                  ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
1135          040000                  WRTGAT= BIT14                   ;WRITE GATE
1136          100000                  RDGATE= BIT15                   ;READ GATE
1137
1138                                  .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1139
1140          000040                  D.DRA=  BIT5                    ;DRIVE AVAILABLE
1141          000100                  D.VV=   BIT6                    ;VOLUME VALID
1142          000200                  D.DRDY= BIT7                    ;DRIVE READY
1143          000400                  D.DDT=  BIT8                    ;DRIVE TYPE (0=RK06)
1144          001000                  D.FORM= BIT9                    ;DRIVE FORMAT
1145          002000                  D.OFF=  BIT10                   ;OFFSET ON
1146          004000                  D.WRL=  BIT11                   ;WRITE LOCK
1147          010000                  D.SPIN= BIT12                   ;SPINDLE ON
1148          020000                  D.PIP=  BIT13                   ;POSITIONING IN PROGRESS
1149          040000                  D.DSC=  BIT14                   ;DRIVE STATUS CHANGE
1150
1151                                  .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1152
1153          000020                  D.TFOK= BIT4                    ;TRACK FOLLOWING OK
1154          000040                  D.HDHM= BIT5                    ;HEADS HOME
1155          000100                  D.BRHM= BIT6                    ;BRUSHES HOME
1156          000200                  D.DOOR= BIT7                    ;DOOR INTERLOCKED
1157          000400                  D.CART= BIT8                    ;CARTRAGE INTERLOCK
1158          001000                  D.SPOK= BIT9                    ;SPEED OK
1159          002000                  D.FWD=  BIT10                   ;FORWARD
1160          004000                  D.REV=  BIT11                   ;REVERSE
1161          010000                  D.LOAD= BIT12                   ;HEADS LOADING
1162          020000                  D.RTZ=  BIT13                   ;RETURN TO ZERO
1163          040000                  D.UNLD= BIT14                   ;HEADS UNLOADING
1164
1165                                  .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1166
1167          000040                  D.IDAE= BIT5                    ;INVALID DISK ADDRESS ERROR:HEAD/CYL
1168          000100                  D.ACLO= BIT6                    ;AC LOW
1169          000200                  D.FLT=  BIT7                    ;DRIVE FAULT
1170          000400                  D.NXF=  BIT8                    ;NON-EXECUTABLE FUNCTION CODE
1171          001000                  D.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
1172          002000                  D.SKI=  BIT10                   ;SEEK INCOMPLETE
1173          004000                  D.WLE=  BIT11                   ;WRITE LOCK ERROR
1174          010000                  D.SPLS= BIT12                   ;SPEED LOSS
1175          020000                  D.DROT= BIT13                   ;DRIVE OFF TRACK
1176          040000                  D.UNS=  BIT14                   ;R/W UNSAFE
1177
1178                                  .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1179
1180          000020                  D.SECT= BIT4                    ;SECTOR ERROR
1181          000040                  D.WCUR= BIT5                    ;WRITE CURRENT AND NO WRITE GATE
1182          000100                  D.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
1183          000200                  D.HDFL= BIT7                    ;HEAD FAULT
1184          000400                  D.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
1185          001000                  D.XERR= BIT9                    ;INDEX ERROR
1186          002000                  D.TIB=  BIT10                   ;TRIBIT ERROR
1187          004000                  D.PLO=  BIT11                   ;PLO ERROR
```

```
1188            010000                  D.NMOV= BIT12                   ;SEEK AND NO MOTION
1189            020000                  D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1190            040000                  D.SUNS= BIT14                   ;SERVO UNSAFE
1191
1192                                    .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)
1193
1194            000007                  M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1195            017760                  M.CDIF= 17760                   ;CYLINDER DIFF, BYTE 10
1196            017760                  M.OFST= 17760                   ;OFFSET VALUE, BYTE 10
1197            077770                  M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1198
1199                                    .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)
1200
1201            000003                  M.ID=   3                       ;BYTE ID, ALL BYTES
1202            017760                  M.CADD= 17760                   ;CYLINDER ADDRESS, BYTE 10
1203            040000                  M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1204            000760                  M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1205            007000                  M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1206            100000                  M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
1207
1208                                    .SBTTL  TRAP CATCHER
1209
1210            000000                          .=0
1211                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1212                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1213                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1214            000174                          .=174
1215  000174    000000                DISPREG:  .WORD   0             ;;SOFTWARE DISPLAY REGISTER
1216  000176    000000                SWREG:    .WORD   0             ;;SOFTWARE SWITCH REGISTER
1217                                    .SBTTL  STARTING ADDRESS(ES)
1218  000200    000137  010566                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1219            000220                          .=220
1220  000220    000137  010556                  JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING
1221
1222            000240                          .=240
1223  000240    000137  055550                  JMP     0.ODT           ;ENTER ODT11
1224
1225                                    .SBTTL  ACT11 HOOKS
1226
1227                                    ;;*****************************************************************
1228                                    ;HOOKS REQUIRED BY ACT11
1229            000244                          $SVPC=.                 ;SAVE PC
1230            000046                          .=46
1231  000046    023674                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1232            000052                          .=52
1233  000052    120000                          .WORD   120000          ;;2)SET LOC.52 TO 120000
1234            000244                          .=$SVPC                 ;; RESTORE PC
1235            001000                          .=1000
1236                                    .SBTTL  APT PARAMETER BLOCK
1237
1238                                    ;;*****************************************************************
1239                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1240                                    ;;*****************************************************************
1241            001000                          .$X=.   ;;SAVE CURRENT LOCATION
1242            000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1243  000024    000200                          200     ;;FOR APT START UP
1244            000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1245  000044    001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
1246            001000                          .=.$X   ;;RESET LOCATION COUNTER
1247                                    ;;*****************************************************************
1248                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1249                                    ;INTERFACE SPEC.
1250
1251  001000                            $APTHD:
1252  001000    000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1253  001002    001204                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1254  001004    000454                  $TSTM:  .WORD   300.    ;;RUN TIM OF LONGEST TEST
1255  001006    001130                  $PASTM: .WORD   600.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1256  001010    001130                  $UNITM: .WORD   600.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1257  001012    000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1258
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001166  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001170  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001172  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001174  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001200  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001201  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001202  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001204                  $MAIL:                          ;;APT MAILBOX
001204  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001206  000000          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001210  000000          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
```

```
1315  001212  000000          $PASS:  .WORD   APASS   ;;PASS COUNT
1316  001214  000000          $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
1317  001216  000000          $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
1318  001220  000000          $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
1319  001222  000000          $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1320  001224                  $ETABLE:                ;;APT ENVIRONMENT TABLE
1321  001224      000         $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1322  001225      000         $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1323  001226  000000          $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1324  001230  000000          $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1325  001232  000000          $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1326                          ;*                      BITS 15-11=CPU TYPE
1327                          ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1328                          ;*                              11/70=06,PDQ=07,Q=10
1329                          ;*                      BIT 10=REAL TIME CLOCK
1330                          ;*                      BIT  9=FLOATING POINT PROCESSOR
1331                          ;*                      BIT  8=MEMORY MANAGEMENT
1332  001234      000         $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1333  001235      000         $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1334                          ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1335                          ;*                              900 NSEC CORE=001
1336                          ;*                              300 NSEC BIPOLAR=002
1337                          ;*                              500 NSEC MOS=003
1338  001236  000000          $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1339                          ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1340  001240      000         $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1341  001241      000         $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1342  001242  000000          $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1343  001244      000         $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1344  001245      000         $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1345  001246  000000          $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1346  001250      000         $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1347  001251      000         $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1348  001252  000000          $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1349  001254  000000          $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1350  001256  000000          $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1351  001260  177440          $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1352  001262  000000          $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1353  001264  000000          $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1354  001266  000000          $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1355  001270  000000          $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
1356  001272  000000          $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
1357  001274  000000          $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
1358  001276  000000          $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1359  001300  000000          $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1360  001302  000000          $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1361  001304  000000          $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1362  001306  000000          $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1363  001310  000000          $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
1364  001312  000000          $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
1365  001314  000000          $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
1366  001316  000000          $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
1367  001320  000000          $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
1368  001322  000000          $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
1369  001324  000000          $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
1370  001326  000000          $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
```

```
1371
1372
1373  001330                    $ETEND:
1374
1375          177440            ABASE=  177440          ;DEFAULT BUSS ADDRESS
1376  001330  000210            RKVEC:  210             ;DEFAULT CONTROLLER INTERRUPT VECTOR
1377  001332  000240            RKPRI:  PR5             ;PRIORITY
1378  001334  172540            PKS:    172540          ;P-CLOCK STATUS REG
1379  001336  172542            PKSB:   172542          ;P-CLOCK SET BUFFER
1380  001340  172544            PKRB:   172544          ;P-CLOCK READ BUFFER
1381  001342  177546            LKS:    177546          ;L-CLOCK STATUS REG.
1382
1383  001344  000100            LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
1384  001346  000104            PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.
1385
1386          000114            MEMVEC= 114             ;MEMORY PARITY VECTOR
1387          172100            MEMBAS= 172100          ;MEMORY PARITY OPTION
1388
1389  001350  000000            TRAPPC: 0               ;PC FOR MEMORY PARITY ERROR TRAP
1390
1391  001352  000000            PARAM:  0               ;1 FOR 220 START, NO DEFAULT
1392  001354  000000            FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1393
1394  001356  000000            DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
1395                                                    ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1396
1397          000040            SPBAR=  40              ;SPACE BAR
1398          000003            CONTC=  3               ;CONTROL-C
1399  001360  000000            FRCYL:  0               ;FROM CYLINDER
1400  001362  000000            TOCYL:  0               ;TO CYLINDER
1401  001364  000000            CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
1402  001366  000000            PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
1403  001370  000000            CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
1404  001372  000000            CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1405  001374  000000            CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
1406  001376  000000            CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
1407
1408  001400  000074            HZ:     60.             ;60 FOR 60 CPS
1409                                                    ;50 FOR 50 CPS
1410  001402  000000            COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1411                                                    ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1412  001404  000000            SEC:    0               ;SECOND COUNTER
1413  001406  000000            TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
1414  001410  000000            SECNT:  0               ;SECTOR COUNT
1415  001412  000000            PSEC:   0               ;PREVIOUS SECTOR
1416  001414  000000            ESEC:   0               ;EXPECTED SECTOR
1417  001416  000000            SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1418
1419  001420  000001            T1:     1               ;TIMEOUT CONSTANTS
1420  001422  000012            T10:    10.
1421  001424  000062            T50:    50.
1422  001426  000764            T500:   500.
1423  001430  000144            T100:   100.
1424  001432  011610            T5000:  5000.
1425  001434  141520            T50000: 50000.
1426
```

```
1427  001436  000077                CYL:    63.             ;CYLINDER NUMBERS USED IN
1428  001440  000177                        127.            ;CURRENT CROSSOVER TEST
1429  001442  000277                        191.
1430  001444  000377                        255.
1431  001446  000477                        319.
1432  001450  000577                        383.
1433
1434  001452  000000                TIM1:   0               ;USED IN TIMING TESTS
1435  001454  000000                TIM2:   0
1436  001456  000000                TIM3:   0
1437  001460  000000                TIM4:   0
1438
1439  001462  000000                LPCNT:  0               ;LOOP CTR USED IN CALCLK
1440  001464  000000                LPTIM:  0               ;LOOP TIME IN USEC
1441
1442  001466  000000                SUM:    0               ;LO ORDER FOR TIMING TESTS
1443  001470  000000                        0               ;HI ORDER FOR TIMING TESTS
1444  001472  000000                SUM1:   0               ;LO ORDER FOR TIMING TESTS
1445  001474  000000                        0               ;HI ORDER FOR TIMING TESTS
1446
1447  001476  000000                WD1:    0               ;ACTUAL HEADER/DATA WORD
1448  001500  000000                WD2:    0               ;EXPECTED DATA WORD
1449
1450  001502  000000                OFFERR: 0               ;SET WHEN WRITE CHECK ERROR ON OFFSET
1451
1452
1453  001504  000000                HEAD:   0               ;HEAD NUMBER
1454  001506  000000                HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
1455  001510  000000                FORMAT: 0               ;FORMAT TYPE
1456  001512  000000                FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
1457  001514  000000                WDCNT:  0               ;WORD COUNT
1458
1459  001516  000000                DATA0:  0               ;ALL 0'S
1460  001520  052525                DATA01: 52525           ;0101 PATT
1461  001522  177777                DATA1:  177777          ;ALL 1'S
1462  001524  133467                DPAT1:  133467
1463  001526  070627                DPAT2:  70627
1464
1465  001530  000000                WORD:   0               ;HEADER/DATA WORD
1466  001532  000000                HDWD:   0               ;HEADER WORD FROM RKDB
1467
1468  001534  000000                BSERR:  0               ;CANNOT READ BSE INFO WHEN SET
1469  001536  000000                LIMERR: 0               ;LIMIT DETECT ERROR FLAG
1470  001540  000000                MDSERR: 0               ;MULT DRIVE SEL ERROR FLAG
1471
1472  001542  000102                HDTAB:  .BLKW 66.       ;CALCULATED HEADER WORD TABLE
1473  001746  000102                RHTAB:  .BLKW 66.       ;FILLED AFTER READ HEADER CMD
1474  002152  000102                SRTTAB: .BLKW 66.       ;ABOVE RHTAB SORTED STARTING FORM
1475                                                        ;SECTOR 0 BY SORT ROUTINE
1476  002356  000400                BSE20:  .BLKW 256.      ;20 SECTOR BSE INFO
1477  003356  000400                BSE22:  .BLKW 256.      ;22 SECTOR BSE INFO.
1478  004356  000400                RDTAB:  .BLKW 256.      ;FILLED AFTER READ DATA CMD
1479
1480  005356  000633                INVCYL: 411.            ;INVALID CYLINDER ADDR
1481  005360  000634                        412.
1482  005362  000640                        416.
```

```
1483  005364  000700                        448.
1484  005366  000740                        480.
1485
1486
1487  005370    001     002     004  ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
1488  005373    010     020     040
1489  005376    100     200
1490                                        .EVEN
1491
1492                                        .LIST   MD
1493
1494                                ;
1495                                ;USE LOOP  X TO OMIT SUBCLR
1496                                ;
1497
1498                                ;
1499                                ;A=MSG A0 ERROR#, B=MSG B0 ERROR#
1500                                ;C=MSG A1 ERROR#, D=MSG B1 ERROR#
1501                                ;
1502                                ;E=<ERROR CONDITION DESCRIPTION>
1503                                ;F=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1504                                ;
1505                                ;       NOTE: F CAN BE ANY BIT COMBINATION DESIRED
1506                                ;
1507
1508                                ;
1509                                ;A=CYL DIFF/OFFSET ERROR#
1510                                ;B=CYL ADDR ERROR#
1511                                ;C=<ERROR CONDITION DESCRIPTION>
1512                                ;
1513
1514
1515
1516                                ;
1517                                ;USE CALIB      X   TO OMIT CKWD12 & CKWD3
1518                                ;
1519
1520
1521                                ;
1522                                ; A=WRHEAD/<CFMT!WRHEAD>
1523                                ; USE WRHDR     <A>,X   TO OMIT CKWD12
1524                                ;
1525
1526                                ;
1527                                ; A=RDHEAD/<CFMT!RDHEAD>
1528                                ;USE RDHDR <A>,X TO OMIT CKWD12
1529                                ;
1530
1531
1532
1533                                ;
1534                                ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1535                                ;
1536
1537                                ;
1538                                ;QUICK SEEK.    ENTER WITH CYL# IN RKDC
```

```
1539                                    ;
1540
1541
1542                                    ;
1543                                    ;A=WRDATA/<CFMT!WRDATA>
1544                                    ;USE WDATA <A>,X TO OMIT CKWD12
1545                                    ;
1546
1547
1548                                    ;
1549                                    ;A=RDDATA/<CFMT!RDDATA>
1550                                    ;USE RDATA      <A>,X TO OMIT CKWD12
1551                                    ;
1552
1553
1554                                    ;
1555                                    ;A=WRTCHK/<CFMT!WRTCHK>
1556                                    ;C=16 FOR STD ERROR MSG
1557                                    ;C=134/135/136/137 FOR ERROR MSG USED IN 'SBOUND' ROUTINE
1558                                    ;USE WRCHK      <A>,C,X TO OMIT CKWD12
1559                                    ;
1560
1561                                    ;
1562                                    ;MACRO TO TEST THAT WRITE CHECK OCCURRED AT SECTOR BOUNDRY
1563                                    ;A&B=134,135 FOR WRITE PROTECT SW TEST
1564                                    ;A&B=136,137 FOR AC LOW TEST PART 2
1565                                    ;C=JUMP ADDR TO REPEAT TEST
1566                                    ;
1567
1568                                    ;
1569                                    ;QUICK START SPINDLE
1570                                    ;
1571
1572                                    ;
1573                                    ;ROUTINE TO ISSUE PACK COMMAND
1574                                    ;
1575
1576                                    ;
1577                                    ;QUICK UNLOAD
1578                                    ;
1579
1580
1581
1582                                            .NLIST  MD
1583
1584
1585                                    ;
1586                                    ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
1587                                    ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
1588                                    ;
1589
1590  005400  000000                    HCS1:   0                       ;HOLD RKCS1
1591  005402  000000                    HCS2:   0                       ;HOLD RKCS2
1592  005404  000000                    HWC:    0                       ;HOLD RKWC
1593  005406  000000                    HBA:    0                       ;ETC.
1594  005410  000000                    HDA:    0
```

```
1595  005412  000000                HDS:    0
1596  005414  000000                HER:    0
1597  005416  000000                HASOF:  0
1598  005420  000000                HDC:    0
1599  005422  000000                HDB:    0
1600  005424  000000                HMR1:   0
1601  005426  000000                HMR2:   0
1602  005430  000000                HMR3:   0
1603  005432  000000                HPOS:   0
1604  005434  000000                HPAT:   0
1605
1606                                ;
1607                                ;THE FOLLOWING ARE "SHOULD BE" REGISTERS FOR THE ABOVE REGISTERS
1608                                ;
1609  005436  000000                SBMR2:  0                       ;'SHOULD BE' FOR HMR2
1610  005440  000000                SBMR3:  0                       ;ETC
1611
1612  005442  000000                TEMP1:  0                       ;TEMPORARY STORAGE.
1613  005444  000000                TEMP2:  0
1614  005446  000000                TEMP3:  0
1615  005450  000000                TEMP4:  0
1616  005452  000000                TEMP5:  0
1617
1618                                ;
1619                                ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
1620                                ;
1621
1622  005454  000000                DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
1623  005456  000000                DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
1624  005460  000000                ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
1625  005462  000000                PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
1626  005464  000000                DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
1627
1628                                ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
1629                                ;IS PRESENT AND IS TO BE TESTED.
1630
1631  005466  000000                DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
1632  005470  000000                DRIV1:  0                       ;FOR DRIVE 1
1633  005472  000000                DRIV2:  0                       ;FOR DRIVE 2
1634  005474  000000                DRIV3:  0                       ;FOR DRIVE 3
1635  005476  000000                DRIV4:  0                       ;FOR DRIVE 4
1636  005500  000000                DRIV5:  0                       ;FOR DRIVE 5
1637  005502  000000                DRIV6:  0                       ;FOR DRIVE 6
1638  005504  000000                DRIV7:  0                       ;FOR DRIVE 7
1639
1640  005506  000000                LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
1641  005510  000000                PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
1642  005512  000000                DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
1643  005514  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

005516                          $ERRTB:

                                ;ERROR 1
005516  043237                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
005520  050367                          DH1
005522  054460                          DT1
005524  054720                          DF1

                                ;ERROR 2
005526  043456                          EM5                     ;DETECTED MDS
005530  050367                          DH1
005532  054460                          DT1
005534  054720                          DF1

                                ;ERROR 3
005536  043477                          EM6                     ;DETECTED UFE
005540  050367                          DH1
005542  054460                          DT1
005544  054720                          DF1

                                ;ERROR 4
005546  043520                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
005550  050367                          DH1
005552  054460                          DT1
005554  054720                          DF1
                                ;ERROR 5
005556  043607                          EM8                     ;DR PRESENT BUT NOT SPECIFIED BY CPERATOR
005560  050367                          DH1
005562  054460                          DT1
005564  054720                          DF1

                                ;ERROR 6
005566  043663                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
005570  050367                          DH1
005572  054460                          DT1
005574  054720                          DF1

                                ;ERROR 7
005576  043737                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
005600  050367                          DH1
005602  054460                          DT1
005604  054720                          DF1
```

```
1700
1701                                    ;ERROR 10
1702  005606  044022                        EM11            ;DRA & NED BOTH SET
1703  005610  050367                        DH1
1704  005612  054460                        DT1
1705  005614  054720                        DF1
1706                                    ;ERR 11
1707  005616  044066                        EM12            ;NO RDY
1708  005620  051702                        DH27            ;AFTER WRITE DATA CMD
1709  005622  054460                        DT1
1710  005624  055064                        DF10
1711                                    ;ERR 12
1712  005626  044356                        EM21            ;CERR SET
1713  005630  051702                        DH27
1714  005632  054460                        DT1
1715  005634  055064                        DF10
1716                                    ;ERR 13
1717  005636  044066                        EM12            ;NO RDY
1718  005640  051652                        DH26            ;AFTER READ DATA CMD
1719  005642  054460                        DT1
1720  005644  055064                        DF10
1721                                    ;ERR 14
1722  005646  044356                        EM21            ;CERR SET
1723  005650  051652                        DH26
1724  005652  054460                        DT1
1725  005654  055064                        DF10
1726                                    ;ERR 15
1727  005656  044066                        EM12            ;NO RDY
1728  005660  052111                        DH32            ;AFTER WRITE CHECK CMD
1729  005662  054460                        DT1
1730  005664  055064                        DF10
1731                                    ;ERR 16
1732  005666  047142                        EM80            ;WRITE CHECK ERROR SET
1733  005670  052111                        DH32            ;AFTER WRITE CHECK CMD
1734  005672  054460                        DT1
1735  005674  055064                        DF10
1736                                    ;ERR 17
1737  005676  044066                        EM12            ;CONTR NOT RDY
1738  005700  051365                        DH18            ;AFTER UNLD CMD
1739  005702  054460                        DT1
1740  005704  055064                        DF10
1741                                    ;ERR 20
1742  005706  044124                        EM13            ;NO ATTN
1743  005710  051365                        DH18
1744  005712  054504                        DT2
1745  005714  055100                        DF11
1746                                    ;ERR 21
1747  005716  047317                        EM83            ;DATA CHECK ERROR
1748  005720  051652                        DH26            ;AFTER READ DATA CMD
1749  005722  054460                        DT1
1750  005724  055064                        DF10
1751                                    ;ERR 22
1752  005726  044356                        EM21            ;CERR SET
1753  005730  052111                        DH32            ;AFTER WRITE CHECK CMD
1754  005732  054460                        DT1
1755  005734  055064                        DF10
```

```
1756
1757  005736  044273          ;ERR 23
1758  005740  051702                EM18                    ;MSG B0 ERROR
1759  005742  054460                DH27                    ;AFTER WRITE DATA CMD
1760  005744  055044                DT1
1761                           DF9
1762  005746  044356          ;ERROR 24
1763  005750  051470                EM21                    ;CERR SET
1764  005752  054460                DH21                    ;AFTER SCLR
1765  005754  055064                DT1
1766                           DF10
1767  005756  044335          ;ERR 25
1768  005760  051702                EM20                    ;MSG B1 ERROR
1769  005762  054460                DH27
1770  005764  055044                DT1
1771                           DF9
1772  005766  044273          ;ERR 26
1773  005770  051652                EM18
1774  005772  054460                DH26                    ;AFTER READ DATA CMD
1775  005774  055044                DT1
1776                           DF9
1777
1778  005776  044471          ;ERROR 27
1779  006000  051412                EM24                    ;VOL VALID NOT SET
1780  006002  054460                DH19                    ;AFTER PACK CMD
1781  006004  055064                DT1
1782                           DF10
1783  006006  044335          ;ERR 30
1784  006010  051652                EM20                    ;MSG B1 ERROR
1785  006012  054460                DH26                    ;AFTER READ DATA CMD.
1786  006014  055044                DT1
1787                           DF9
1788  006016  044273          ;ERR 31
1789  006020  052111                EM18                    ;MSG B0 ERROR
1790  006022  054460                DH32                    ;AFTER WRITE CHECK CMD
1791  006024  055044                DT1
1792                           DF9
1793  006026  044335          ;ERR 32
1794  006030  052111                EM20                    ;MSG B1 ERROR
1795  006032  054460                DH32
1796  006034  055044                DT1
1797                           DF9
1798  006036  044066          ;ERR 33
1799  006040  051602                EM12                    ;CONTR NOT READY
1800  006042  054504                DH24                    ;AFTER OFFSET CMD
1801  006044  055100                DT2
1802                           DF11
1803  006046  044124          ;ERR 34
1804  006050  051602                EM13                    ;NO ATTN
1805  006052  054504                DH24
1806  006054  055100                DT2
1807                           DF11
1808  006056  044252          ;ERR 35
1809  006060  053356                EM17                    ;MSG A0 ERROR
1810  006062  054460                DH53                    ;DURING OFFSET COMMAND
1811  006064  055024                DT1
                               DF8
```

```
1812
1813  006066  044314          ;ERR 36
1814  006070  053356                EM19                    ;MSG A1 ERROR
1815  006072  054460                DH53
1816  006074  055024                DT1
1817                           DF8
1818  006076  044314          ;ERR 37
1819  006100  051602                EM19                    ;MSG A1 ERROR
1820  006102  054460                DH24                    ;AFTER OFFSET CMD
1821  006104  055024                DT1
1822                           DF8
1823  006106  044335          ;ERR 40
1824  006110  051602                EM20                    ;MSG B1 ERROR
1825  006112  054460                DH24
1826  006114  055044                DT1
1827                           DF9
1828  006116  044146          ;ERR 41
1829  006120  051365                EM14                    ;WRONG ATTN
1830  006122  054504                DH18                    ;AFTER UNLOAD CMD
1831  006124  055100                DT2
1832                           DF11
1833  006126  044173          ;ERR 42
1834  006130  051365                EM15                    ;DRDY NOT CLEARED
1835  006132  054460                DH18
1836  006134  055064                DT1
1837                           DF10
1838  006136  044225          ;ERR 43
1839  006140  051365                EM16                    ;DSC NOT SET
1840  006142  054460                DH18
1841  006144  055064                DT1
1842                           DF10
1843  006146  044400          ;ERR 44
1844  006150  050667                EM22                    ;DOOR NOT CLEARED
1845  006152  054460                DH8                     ;AFTER DRIVE UNLOADED & DOOR OPENED
1846  006154  055064                DT1
1847                           DF10
1848  006156  044252          ;ERR 45
1849  006160  050667                EM17                    ;MSG A0 ERROR
1850  006162  054460                DH8
1851  006164  055024                DT1
1852                           DF8
1853  006166  044273          ;ERR 46
1854  006170  050667                EM18                    ;MSG B0 ERROR
1855  006172  054460                DH8
1856  006174  055044                DT1
1857                           DF9
1858  006176  045344          ;ERROR 47
1859  006200  051341                EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
1860  006202  054460                DH17                    ;AFTER RECAL CMD
1861  006204  055064                DT1
1862                           DF10
1863  006206  045413          ;ERROR 50
1864  006210  051341                EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
1865  006212  054460                DH17                    ;AFTER RECAL COMD
1866  006214  055064                DT1
1867                           DF10
                               ;ERR 51
```

```
1868  006216  047762                  EM93            ;WRONG CYL# IN HEADER WORD
1869  006220  051627                  DH25            ;AFTER SEEK CMD
1870  006222  054662                  DT9
1871  006224  055164                  DF20
1872                          ;ERR 52
1873  006226  044252                  EM17            ;MSG AD ERROR
1874  006230  051702                  DH27            ;AFTER WRITE DATA CMD
1875  006232  054460                  DT1
1876  006234  055024                  DF8
1877                          ;ERR 53
1878  006236  044314                  EM19            ;MSG A1 ERROR
1879  006240  051702                  DH27
1880  006242  054460                  DT1
1881  006244  055024                  DF8
1882                          ;ERR 54
1883  006246  044252                  EM17            ;MSG AD ERROR
1884  006250  051652                  DH26            ;AFTER READ DATA CMD
1885  006252  054460                  DT1
1886  006254  055024                  DF8
1887                          ;ERROR 55
1888  006256  044124                  EM13            ;NO ATTN
1889  006260  051341                  DH17            ;AFTER RECAL CMD
1890  006262  054460                  DT1
1891  006264  055064                  DF10
1892                          ;ERR 56
1893  006266  044314                  EM19            ;MSG A1 ERROR
1894  006270  051652                  DH26
1895  006272  054460                  DT1
1896  006274  055024                  DF8
1897                          ;ERR 57
1898  006276  044252                  EM17            ;MSG AD ERROR
1899  006300  052111                  DH32            ;AFTER WRITE CHECK CMD
1900  006302  054460                  DT1
1901  006304  055024                  DF8
1902                          ;ERR 60
1903  006306  044314                  EM19            ;MSG A1 ERROR
1904  006310  052111                  DH32
1905  006312  054460                  DT1
1906  006314  055024                  DF8
1907                          ;ERR 61
1908  006316  044273                  EM18            ;MSG BD ERROR
1909  006320  053356                  DH53            ;DURING OFFSET CMD
1910  006322  054460                  DT1
1911  006324  055044                  DF9
1912                          ;ERR 62
1913  006326  044335                  EM20            ;MSG B1 ERROR
1914  006330  053356                  DH53
1915  006332  054460                  DT1
1916  006334  055044                  DF9
1917                          ;ERR 63
1918  006336  044314                  EM19            ;MSG A1 ERROR
1919  006340  050667                  DH8             ;AFTER DRIVE UNLOADED & DOOR OPENED
1920  006342  054460                  DT1
1921  006344  055024                  DF8
1922                          ;ERR 64
1923  006346  044335                  EM20            ;MSG B1 ERROR
```

```
1924  006350  050667                    DH8
1925  006352  054460                    DT1
1926  006354  055044                    DF9
1927                          ;ERR 65
1928  006356  044441                    EM23          ;SPIN SET
1929  006360  051034                    DH11          ;AFTER LOADING HEADS WITH DOOR OPEN
1930  006362  054460                    DT1
1931  006364  055064                    DF10
1932                          ;ERR 66
1933  006366  044252                    EM17          ;MSG A0 ERROR
1934  006370  051034                    DH11
1935  006372  054460                    DT1
1936  006374  055024                    DF8
1937                          ;ERR 67
1938  006376  044273                    EM18          ;MSG B0 ERROR
1939  006400  051034                    DH11
1940  006402  054460                    DT1
1941  006404  055044                    DF9
1942                          ;ERR 70
1943  006406  044314                    EM19          ;MSG A1 ERROR
1944  006410  051034                    DH11
1945  006412  054460                    DT1
1946  006414  055024                    DF8
1947                          ;ERR 71
1948  006416  044335                    EM20          ;MSG B1 ERROR
1949  006420  051034                    DH11
1950  006422  054460                    DT1
1951  006424  055044                    DF9
1952                          ;ERR 72
1953  006426  044527                    EM25          ;CARTRIDGE NOT CLEARED
1954  006430  051110                    DH12          ;AFTER DISK PACK REMOVED
1955  006432  054460                    DT1
1956  006434  055064                    DF10
1957                          ;ERR 73
1958  006436  044575                    EM26          ;VV NOT CLEARED
1959  006440  051110                    DH12
1960  006442  054460                    DT1
1961  006444  055064                    DF10
1962                          ;ERR 74
1963  006446  044252                    EM17          ;MSG A0 ERROR
1964  006450  051110                    DH12
1965  006452  054460                    DT1
1966  006454  055024                    DF8
1967                          ;ERR 75
1968  006456  044273                    EM18          ;MSG B0 ERROR
1969  006460  051110                    DH12
1970  006462  054460                    DT1
1971  006464  055044                    DF9
1972                          ;ERR 76
1973  006466  044314                    EM19          ;MSG A1 ERROR
1974  006470  051110                    DH12
1975  006472  054460                    DT1
1976  006474  055024                    DF8
1977                          ;ERR 77
1978  006476  044335                    EM20          ;MSG B1 ERROR
1979  006500  051110                    DH12
```

```
 1980  006502  054460                    DT1
 1981  006504  055044                    DF9
 1982                            ;ERR 100
 1983  006506  044441                    EM23                    ;SPIN SET
 1984  006510  051140                    DH13                    ;AFTER LOADING HEADS WITH CART. OUT
 1985  006512  054460                    DT1
 1986  006514  055064                    DF10
 1987                            ;ERR 101
 1988  006516  044252                    EM17                    ;MSG A0 ERROR
 1989  006520  051140                    DH13
 1990  006522  054460                    DT1
 1991  006524  055024                    DF8
 1992                            ;ERR 102
 1993  006526  044273                    EM18                    ;MSG B0 ERROR
 1994  006530  051140                    DH13
 1995  006532  054460                    DT1
 1996  006534  055044                    DF9
 1997                            ;ERR 103
 1998  006536  044314                    EM19                    ;MSG A1 ERROR
 1999  006540  051140                    DH13
 2000  006542  054460                    DT1
 2001  006544  055024                    DF8
 2002                            ;ERR 104
 2003  006546  044335                    EM20                    ;MSG B1 ERROR
 2004  006550  051140                    DH13
 2005  006552  054460                    DT1
 2006  006554  055044                    DF9
 2007                            ;ERR 105
 2008  006556  046250                    EM52                    ;UNS NOT SET
 2009  006560  054233                    DH64                    ;AFTER MDS FOUND
 2010  006562  054460                    DT1
 2011  006564  055064                    DF10
 2012                            ;ERR 106
 2013  006566  044664                    EM28                    ;VV SET
 2014  006570  051255                    DH15                    ;WITHOUT PACK CMD
 2015  006572  054460                    DT1
 2016  006574  055064                    DF10
 2017                            ;ERR 107
 2018  006576  044716                    EM29                    ;DSC NOT SET
 2019  006600  053613                    DH59                    ;AFTER EVEN PARITY ISSUED
 2020  006602  054460                    DT1
 2021  006604  055064                    DF10
 2022                            ;ERR 110
 2023  006606  046037                    EM48                    ;WRL NOT CLEARED
 2024  006610  053120                    DH48                    ;AFTER WRITE LOCK SWITCH DISABLED
 2025  006612  054460                    DT1
 2026  006614  055064                    DF10
 2027                            ;ERR 111
 2028  006616  044743                    EM30                    ;ATTN NOT CLEARED
 2029  006620  051302                    DH16                    ;AFTER UNIT SELECT PLUG REMOVED
 2030  006622  054504                    DT2
 2031  006624  055100                    DF11
 2032                            ;ERR 112
 2033  006626  044637                    EM27                    ;NED NOT SET
 2034  006630  051302                    DH16
 2035  006632  054460                    DT1
```

```
2036  006634  055064                    DF10
2037                            ;ERR 113
2038  006636  044124                    EM13          ;ATTN NOT SET
2039  006640  053613                    DH59          ;AFTER EVEN PARITY ISSUED
2040  006642  054460                    DT1
2041  006644  055024                    DF8
2042                            ;ERROR 114
2043  006646  044273                    EM18          ;MSG BD ERROR
2044  006650  053613                    DH59
2045  006652  054460                    DT1
2046  006654  055044                    DF9
2047                            ;ERR 115
2048  006656  046077                    EM49          ;WRL NOT SET
2049  006660  053161                    DH49          ;AFTER WRITE LOCK SW ENABLED
2050  006662  054460                    DT1
2051  006664  055064                    DF10
2052                            ;ERROR 116
2053  006666  044066                    EM12          ;CONT NOT RDY
2054  006670  051412                    DH19          ;AFTER PACK CMD
2055  006672  054460                    DT1
2056  006674  055064                    DF10
2057                            ;ERROR 117
2058  006676  044066                    EM12          ;CONT NOT RDY
2059  006700  051435                    DH20          ;AFTER SEL DR CMD
2060  006702  054460                    DT1
2061  006704  055064                    DF10
2062                            ;ERROR 120
2063  006706  044066                    EM12
2064  006710  051470                    DH21          ;AFTER SUBSYS CLEAR
2065  006712  054460                    DT1
2066  006714  055064                    DF10
2067                            ;ERR 121
2068  006716  046133                    EM50          ;WLE NOT SET
2069  006720  053221                    DH50          ;AFTER WRITING WITH WRITE LOCK SET
2070  006722  054460                    DT1
2071  006724  055064                    DF10
2072                            ;ERR 122
2073  006726  044637                    EM27          ;NED NOT SET
2074  006730  051550                    DH23          ;AFTER WRONG PORT SELECTED
2075  006732  054460                    DT1
2076  006734  055064                    DF10
2077                            ;ERR 123
2078  006736  044252                    EM17          ;MSG AO ERROR
2079  006740  053221                    DH50          ;AFTER WRITING WITH WRITE LOCK ENABLED
2080  006742  054460                    DT1
2081  006744  055024                    DF8
2082                            ;ERROR 124
2083  006746  044066                    EM12
2084  006750  051341                    DH17          ;AFTER RECAL CMD
2085  006752  054460                    DT1
2086  006754  055064                    DF10
2087                            ;ERR 125
2088  006756  044273                    EM18          ;MSG BD ERROR
2089  006760  053221                    DH50          ;AFTER WITING WITH WRL ENABLED
2090  006762  054460                    DT1
2091  006764  055044                    DF9
```

```
2092
2093  006766  044314          ;ERR 126
2094  006770  053221                EM19                    ;MSG A1 ERROR
2095  006772  054460                DH50
2096  006774  055024                DT1
2097                           DF8
2098  006776  044335          ;ERR 127
2099  007000  053221                EM20                    ;MSG B1 ERROR
2100  007002  054460                DH50
2101  007004  055044                DT1
2102                           DF9
2103  007006  044774          ;ERR 130
2104  007010  051767                EM31                    ;NED NOT CLEARED
2105  007012  054460                DH29                    ;AFTER CORRECT PORT SELECTED
2106  007014  055064                DT1
2107                           DF10
2108  007016  044066          ;ERROR 131
2109  007020  051627                EM12                    ;NO RDY
2110  007022  054460                DH25                    ;AFTER SEEK CMD
2111  007024  055064                DT1
2112                           DF10
2113  007026  044124          ;ERROR 132
2114  007030  051627                EM13                    ;NO ATTN
2115  007032  054460                DH25
2116  007034  055064                DT1
2117                           DF10
2118  007036  044637          ;ERR 133
2119  007040  051733                EM27                    ;NED NOT SET
2120  007042  054460                DH28                    ;AFTER BOTH PORTS DESELECTED
2121  007044  055064                DT1
2122                           DF10
2123  007046  046175          ;ERR 134
2124  007050  053221                EM51                    ;WRITE LOCK NOT SET SECTOR BOUNDRY
2125  007052  054526                DH50                    ;AFTER WRITING WITH WRL ENABLED
2126  007054  054744                DT3
2127                           DF3
2128  007056  046175          ;ERR 135
2129  007060  053667                EM51
2130  007062  054526                DH60                    ;AFTER WRITE LOCK ENABLED WHILE WRITING
2131  007064  054760                DT3
2132                           DF4
2133  007066  046175          ;ERR 136
2134  007070  054162                EM51
2135  007072  054526                DH63                    ;AFTER WRITE LOCK ENABLED FROM AC OFF
2136  007074  054744                DT3
2137                           DF3
2138  007076  046175          ;ERR 137
2139  007100  054162                EM51
2140  007102  054526                DH63
2141  007104  054760                DT3
2142                           DF4
2143  007106  045025          ;ERR 140
2144  007110  052055                EM32                    ;SPINDLE ON NOT SET
2145  007112  054460                DH31                    ;AFTER DRIVE MANUALLY LOADED
2146  007114  055064                DT1
2147                           DF10
                               ;ERR 141
```

```
2148  007116  045061                EM33                    ;DRIVE NOT READY
2149  007120  052372                DH38                    ;AFTER AC POWERED UP
2150  007122  054460                DT1
2151  007124  055064                DF10
2152                          ;ERR 142
2153  007126  045112                EM34
2154  007130  052055                DH31
2155  007132  054460                DT1
2156  007134  055064                DF10
2157                          ;ERR 143
2158  007136  044066                EM12                    ;CONT NOT READY
2159  007140  052167                DH34                    ;AFTER ST SPIN. CMD
2160  007142  054460                DT1
2161  007144  055064                DF10
2162                          ;ERR 144
2163  007146  044124                EM13                    ;NO ATTN
2164  007150  052167                DH34
2165  007152  054504                DT2
2166  007154  055100                DF11
2167                          ;ERR 145
2168  007156  045153                EM35                    ;HEADS NOT HOME
2169  007160  052223                DH35                    ;AFTER MANUAL UNLOAD
2170  007162  054460                DT1
2171  007164  055064                DF10
2172                          ;ERR 146
2173  007166  046474                EM57                    ;CERR NOT SET
2174  007170  052336                DH37                    ;AFTER TIMEOUT TO POWER DOWN
2175  007172  054460                DT1
2176  007174  055064                DF10
2177                          ;ERR 147
2178  007176  045252                EM37                    ;AC LOW NOT SET
2179  007200  052336                DH37
2180  007202  054460                DT1
2181  007204  055064                DF10
2182                          ;ERR 150
2183  007206  045522                EM42                    ;NED NOT SET
2184  007210  052336                DH37
2185  007212  054460                DT1
2186  007214  055064                DF10
2187                          ;ERROR 151
2188  007216  044066                EM12                    ;NO RDY
2189  007220  051516                DH22                    ;AFTER CLEAR CMD
2190  007222  054460                DT1
2191  007224  055064                DF10
2192                          ;ERR 152
2193  007226  045547                EM43                    ;AC LO NOT CLEARED
2194  007230  052372                DH38                    ;AFTER AC POWERED UP
2195  007232  054460                DT1
2196  007234  055064                DF10
2197                          ;ERR 153
2198  007236  045601                EM44                    ;VV NOT CLEARED
2199  007240  052372                DH38
2200  007242  054460                DT1
2201  007244  055064                DF10
2202                          ;ERROR 154
2203  007246  046375                EM55                    ;ATTN NOT CLEARED
```

```
2204  007250  051516                    DH22
2205  007252  054504                    DT2
2206  007254  055100                    DF11
2207                            ;ERR 155
2208  007256  045643                    EM45          ;VV SET AFTER HDS LOADED
2209  007260  051255                    DH15          ;WITHOUT 'PACK' CMD
2210  007262  054460                    DT1
2211  007264  055064                    DF10
2212                            ;ERR 156
2213  007266  045720                    EM46          ;NXF=0
2214  007270  052753                    DH45          ;AFTER SEEK WITH VV=0
2215  007272  054460                    DT1
2216  007274  055064                    DF10
2217                            ;ERR 157
2218  007276  045777                    EM47          ;CYL ADDR CHANGED FROM 0
2219  007300  052753                    DH45
2220  007302  054460                    DT1
2221  007304  055064                    DF10
2222                            ;ERR 160
2223  007306  044252                    EM17          ;MSG A0 ERROR
2224  007310  052753                    DH45
2225  007312  054460                    DT1
2226  007314  055024                    DF8
2227                            ;ERR 161
2228  007316  044273                    EM18          ;MSG B0 ERROR
2229  007320  052753                    DH45
2230  007322  054460                    DT1
2231  007324  055044                    DF9
2232                            ;ERR 162
2233  007326  044314                    EM19          ;MSG A1 ERROR
2234  007330  052753                    DH45
2235  007332  054460                    DT1
2236  007334  055024                    DF8
2237                            ;ERR 163
2238  007336  044335                    EM20          ;MSG B1 ERROR
2239  007340  052753                    DH45
2240  007342  054460                    DT1
2241  007344  055044                    DF9
2242                            ;ERR 164
2243  007346  045720                    EM46          ;NXF NOT SET
2244  007350  053012                    DH46          ;AFTER WRITE DATA WITH VV=0
2245  007352  054460                    DT1
2246  007354  055064                    DF10
2247                            ;ERR 165
2248  007356  044252                    EM17          ;MSG A0 ERROR
2249  007360  053012                    DH46
2250  007362  054460                    DT1
2251  007364  055024                    DF8
2252                            ;ERR 166
2253  007366  044273                    EM18          ;MSG B0 ERROR
2254  007370  053012                    DH46
2255  007372  054460                    DT1
2256  007374  055044                    DF9
2257                            ;ERR 167
2258  007376  044314                    EM19          ;MSG A1 ERROR
2259  007400  053012                    DH46
```

```
2260  007402  054460                    DT1
2261  007404  055024                    DF8
2262                            ;ERR 170
2263  007406  044335                    EM20                    ;MSG B1 ERROR
2264  007410  053012                    DH46
2265  007412  054460                    DT1
2266  007414  055024                    DF8
2267                            ;ERROR 171
2268  007416  044066                    EM12                    ;NO RDY
2269  007420  052023                    DH30                    ;AFTER READ HEADER CMD
2270  007422  054460                    DT1
2271  007424  055064                    DF10
2272                            ;ERROR 172
2273  007426  046560                    EM61                    ;NXF DID NOT SET FAULT
2274  007430  052753                    DH45                    ;AFTER SEEK WITH VV=0
2275  007432  054460                    DT1
2276  007434  055064                    DF10
2277                            ;ERROR 173
2278  007436  046626                    EM63                    ;DLT SET
2279  007440  052023                    DH30
2280  007442  054570                    DT5
2281  007444  055130                    DF15
2282                            ;ERROR 174
2283  007446  044356                    EM21                    ;CERR SET
2284  007450  052023                    DH30
2285  007452  054570                    DT5
2286  007454  055130                    DF15
2287                            ;ERR 175
2288  007456  046275                    EM53                    ;UNLD NOT SET
2289  007460  054233                    DH64                    ;AFTER MDS FOUND
2290  007462  054460                    DT1
2291  007464  055064                    DF10
2292                            ;ERR 176
2293  007466  046325                    EM54                    ;CANNOT FIND MDS
2294  007470  054267                    DH65                    ;AFTER SEARCHING ALL DRIVES
2295  007472  054460                    DT1
2296  007474  055064                    DF10
2297                            ;ERROR 177
2298  007476  044575                    EM26                    ;VV NOT CLEARED
2299  007500  051302                    DH16                    ;AFTER UNIT SEL PLUG REMOVED
2300  007502  054460                    DT1
2301  007504  055064                    DF10
2302                            ;ERROR 200
2303  007506  044066                    EM12                    ;NO RDY
2304  007510  052416                    DH39                    ;AFTER WRITE HEADER CMD
2305  007512  054570                    DT5
2306  007514  055130                    DF15
2307                            ;ERROR 201
2308  007516  044356                    EM21                    ;CERR SET
2309  007520  052416                    DH39
2310  007522  054570                    DT5
2311  007524  055130                    DF15
2312                            ;ERROR 202
2313  007526  046430                    EM56                    ;UNEXP MEMORY PARITY ERROR
2314  007530  054332                    DH66                    ;TEST #,PRAP PC
2315  007532  054616                    DT6
```

```
2316  007534  054774                    DF5
2317
2318  007536  044273          ;ERROR 203
2319  007540  052336                    EM18                ;MSG BO ERROR
2320  007542  054460                    DH37                ;AFTER TIMEOUT TO POWER DOWN
2321  007544  055044                    DT1
2322                                     DF9
2323  007546  046474          ;ERR 204
2324  007550  054353                    EM57                ;CERR NOT SET
2325  007552  054460                    DH67                ;AFTER TIMEOUT TO ENABLE WRL
2326  007554  055064                    DT1
2327                                     DF10
2328  007556  046133          ;ERR 205
2329  007560  054353                    EM50                ;WRL NOT SET
2330  007562  054460                    DH67
2331  007564  055064                    DT1
2332                                     DF10
2333  007566  046647          ;ERROR 206
2334  007570  050367                    EM64                ;WCE AT CYL 411,TRK 2, SEC 21
2335  007572  054700                    DH1
2336  007574  055014                    DT10
2337                                     DF7
2338  007576  046474          ;ERROR 207
2339  007600  053221                    EM57                ;CERR NOT SET
2340  007602  054460                    DH50                ;AFTER WRITING WITH WRL ENABLED
2341  007604  055064                    DT1
2342                                     DF10
2343  007606  044356          ;ERROR 210
2344  007610  051627                    EM21                ;CERR SET
2345  007612  054460                    DH25
2346  007614  055064                    DT1
2347                                     DF10
2348  007616  000000          ;ERROR 211
2349  007620  000000                    0
2350  007622  000000                    0
2351  007624  000000                    0
2352                                     0
2353  007626  000000          ;ERROR 212
2354  007630  000000                    0
2355  007632  000000                    0
2356  007634  000000                    0
2357                                     0
2358  007636  000000          ;ERROR 213
2359  007640  000000                    0
2360  007642  000000                    0
2361  007644  000000                    0
2362                                     0
2363  007646  000000          ;ERROR 214
2364  007650  000000                    0
2365  007652  000000                    0
2366  007654  000000                    0
2367                                     0
2368  007656  000000          ;ERROR 215
2369  007660  000000                    0
2370  007662  000000                    0
2371  007664  000000                    0
```

```
2372                                    ;ERROR 216
2373   007666   000000                            0
2374   007670   000000                            0
2375   007672   000000                            0
2376   007674   000000                            0
2377                                    ;ERROR 217
2378   007676   000000                            0
2379   007700   000000                            0
2380   007702   000000                            0
2381   007704   000000                            0
2382                                    ;ERROR 220
2383   007706   000000                            0
2384   007710   000000                            0
2385   007712   000000                            0
2386   007714   000000                            0
2387                                    ;ERROR 221
2388   007716   044252                            EM17                    ;MSG A0 ERROR
2389   007720   051341                            DH17
2390   007722   054460                            DT1
2391   007724   055024                            DF8
2392                                    ;ERROR 222
2393   007726   044314                            EM19                    ;MSG A1 ERROR
2394   007730   051341                            DH17
2395   007732   054460                            DT1
2396   007734   055024                            DF8
2397                                    ;ERROR 223
2398   007736   000000                            0
2399   007740   000000                            0
2400   007742   000000                            0
2401   007744   000000                            0
2402                                    ;ERROR 224
2403   007746   000000                            0
2404   007750   000000                            0
2405   007752   000000                            0
2406   007754   000000                            0
2407                                    ;ERROR 225
2408   007756   000000                            0
2409   007760   000000                            0
2410   007762   000000                            0
2411   007764   000000                            0
2412                                    ;ERROR 226
2413   007766   044066                            EM12                    ;NO RDY
2414   007770   051652                            DH26                    ;AFTER READ DATA CMD
2415   007772   054460                            DT1
2416   007774   055064                            DF10
2417                                    ;ERROR 227
2418   007776   044356                            EM21                    ;CERR SET
2419   010000   051652                            DH26
2420   010002   054570                            DT5
2421   010004   055130                            DF15
2422                                    ;ERROR 230
2423   010006   046606                            EM62                    ;DTE SET
2424   010010   051652                            DH26
2425   010012   054570                            DT5
2426   010014   055130                            DF15
2427                                    ;ERROR 231
```

```
2428  010016  046626                    EM63                    ;DLT SET
2429  010020  051652                    DH26
2430  010022  054570                    DT5
2431  010024  055130                    DF15
2432                              ;ERROR 232
2433  010026  000000                    0
2434  010030  000000                    0
2435  010032  000000                    0
2436  010034  000000                    0
2437                              ;ERROR 233
2438  010036  047052                    EM68                    ;CANNOT READ BSE INFO
2439  010040  052553                    DH42                    ;ON SECT 0,2,4,6,8
2440  010042  054460                    DT1
2441  010044  055144                    DF17
2442                              ;ERROR 234
2443  010046  047052                    EM68
2444  010050  052624                    DH43                    ;ON SECT 1,3,5,7,9
2445  010052  054460                    DT1
2446  010054  055144                    DF17
2447                              ;ERROR 235
2448  010056  000000                    0
2449  010060  000000                    0
2450  010062  000000                    0
2451  010064  000000                    0
2452                              ;ERROR 236
2453  010066  000000                    0
2454  010070  000000                    0
2455  010072  000000                    0
2456  010074  000000                    0
2457                              ;ERROR 237
2458  010076  000000                    0
2459  010100  000000                    0
2460  010102  000000                    0
2461  010104  000000                    0
2462                              ;ERROR 240
2463  010106  000000                    0
2464  010110  000000                    0
2465  010112  000000                    0
2466  010114  000000                    0
2467                              ;ERROR 241
2468  010116  000000                    0
2469  010120  000000                    0
2470  010122  000000                    0
2471  010124  000000                    0
2472                              ;ERROR 242
2473  010126  000000                    0
2474  010130  000000                    0
2475  010132  000000                    0
2476  010134  000000                    0
2477
2478                              ;ERROR 243
2479  010136  000000                    0
2480  010140  000000                    0
2481  010142  000000                    0
2482  010144  000000                    0
2483                              ;ERR 244
```

```
2484  010146  047115                    EM74                    ;RTZ NOT SET
2485  010150  052526                    DH41                    ;DURING RECAL CMD
2486  010152  054460                    DT1
2487  010154  055064                    DF10
2488                              ;ERR 245
2489  010156  000000                      0
2490  010160  000000                      0
2491  010162  000000                      0
2492  010164  000000                      0
2493                              ;ERR 246
2494  010166  000000                      0
2495  010170  000000                      0
2496  010172  000000                      0
2497  010174  000000                      0
2498                              ;ERR 247
2499  010176  000000                      0
2500  010200  000000                      0
2501  010202  000000                      0
2502  010204  000000                      0
2503                              ;ERR 250
2504  010206  000000                      0
2505  010210  000000                      0
2506  010212  000000                      0
2507  010214  000000                      0
2508                              ;ERR 251
2509  010216  000000                      0
2510  010220  000000                      0
2511  010222  000000                      0
2512  010224  000000                      0
2513                              ;ERR 252
2514  010226  000000                      0
2515  010230  000000                      0
2516  010232  000000                      0
2517  010234  000000                      0
2518                              ;ERR 253
2519  010236  000000                      0
2520  010240  000000                      0
2521  010242  000000                      0
2522  010244  000000                      0
2523                              ;ERR 254
2524  010246  000000                      0
2525  010250  000000                      0
2526  010252  000000                      0
2527  010254  000000                      0
2528                              ;ERR 255
2529  010256  000000                      0
2530  010260  000000                      0
2531  010262  000000                      0
2532  010264  000000                      0
2533                              ;ERR 256
2534  010266  000000                      0
2535  010270  000000                      0
2536  010272  000000                      0
2537  010274  000000                      0
2538                              ;ERR 257
2539  010276  000000                      0
```

```
2540  010300  000000                    0
2541  010302  000000                    0
2542  010304  000000                    0
2543                          ;ERR 260
2544  010306  044252                    EM17          ;MSG A0 ERROR
2545  010310  051602                    DH24          ;AFTER OFFSET CMD
2546  010312  054460                    DT1
2547  010314  055024                    DF8
2548                          ;ERR 261
2549  010316  044273                    EM18          ;MSG B0 ERROR.
2550  010320  051602                    DH24
2551  010322  054460                    DT1
2552  010324  055044                    DF9
2553                          ;ERR 262
2554  010326  000000                    0
2555  010330  000000                    0
2556  010332  000000                    0
2557  010334  000000                    0
2558                          ;ERR 263
2559  010336  000000                    0
2560  010340  000000                    0
2561  010342  000000                    0
2562  010344  000000                    0
2563                          ;ERR 264
2564  010346  000000                    0
2565  010350  000000                    0
2566  010352  000000                    0
2567  010354  000000                    0
2568                          ;ERR 265
2569  010356  000000                    0
2570  010360  000000                    0
2571  010362  000000                    0
2572  010364  000000                    0
2573                          ;ERR 266
2574  010366  000000                    0
2575  010370  000000                    0
2576  010372  000000                    0
2577  010374  000000                    0
2578                          ;ERR 267
2579  010376  044273                    EM18          ;MSG B0 ERROR
2580  010400  052416                    DH39          ;AFTER WRITE HEADER CMD
2581  010402  054460                    DT1
2582  010404  055044                    DF9
2583                          ;ERR 270
2584  010406  044335                    EM20          ;MSG B1 ERROR.
2585  010410  052416                    DH39
2586  010412  054460                    DT1
2587  010414  055044                    DF9
2588                          ;ERR 271
2589  010416  000000                    0
2590  010420  000000                    0
2591  010422  000000                    0
2592  010424  000000                    0
2593                          ;ERR 272
2594  010426  000000                    0
2595  010430  000000                    0
```

```
2596  010432  000000                    0
2597  010434  000000                    0
2598                          ;ERR 273
2599  010436  000000                    0
2600  010440  000000                    0
2601  010442  000000                    0
2602  010444  000000                    0
2603                          ;ERR 274
2604  010446  000000                    0
2605  010450  000000                    0
2606  010452  000000                    0
2607  010454  000000                    0
2608                          ;ERR 275
2609  010456  044273                    EM18        ;MSG BO ERROR
2610  010460  051341                    DH17        ;AFTER RECAL CMD
2611  010462  054460                    DT1
2612  010464  055044                    DF9
2613                          ;ERR 276
2614  010466  044335                    EM20        ;MSG B1 ERROR
2615  010470  051341                    DH17
2616  010472  054460                    DT1
2617  010474  055044                    DF9
2618                          ;ERR 277
2619  010476  044252                    EM17        ;MSG AO ERROR
2620  010500  052416                    DH39        ;AFTER WRITE HEADER CMD
2621  010502  054460                    DT1
2622  010504  055024                    DF8
2623                          ;ERR 300
2624  010506  044314                    EM19        ;MSG A1 ERROR
2625  010510  052416                    DH39
2626  010512  054460                    DT1
2627  010514  055024                    DF8
2628                          ;ERR 301
2629  010516  000000                    0
2630  010520  000000                    0
2631  010522  000000                    0
2632  010524  000000                    0
2633                          ;ERR 302
2634  010526  000000                    0
2635  010530  000000                    0
2636  010532  000000                    0
2637  010534  000000                    0
2638                          ;ERR 303
2639  010536  000000                    0
2640  010540  000000                    0
2641  010542  000000                    0
2642  010544  000000                    0
2643                          ;ERR 304
2644  010546  000000                    0
2645  010550  000000                    0
2646  010552  000000                    0
2647  010554  000000                    0
2648
```

```
2649
2650                                    .SBTTL  PROGRAM SETUP
2651
2652  010556  012737  000001  001352  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
2653  010564  000402                          BR      PRGSRT          ;START PROGRAM
2654
2655  010566  105037  001352          START:  CLRB    PARAM           ;CLEAR FOR 200 START
2656  010572  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
2657  010574  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
2658  010600  012746  000340                  MOV     #PR7,-(SP)      ;PSW LOADED TO BE
2659  010604  012746  010612                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
2660  010610  000002                          RTI                     ;LOCKOUT ALL INTERRUPTS
2661
2662  010612                          1$:
2663                                  .SBTTL  INITIALIZE THE COMMON TAGS
2664                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2665  010612  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2666  010616  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2667  010620  022706  001140                  CMP     #SWR,R6 ;;DONE?
2668  010624  001374                          BNE     .-6             ;;LOOP BACK IF NO
2669  010626  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2670                                  ;;INITIALIZE A FEW VECTORS
2671  010632  012737  030216  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2672  010640  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2673  010646  012737  030476  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2674  010654  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2675  010662  012737  033772  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2676  010670  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
2677  010676  012737  027752  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2678  010704  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2679  010712  013737  023642  023634          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2680  010720  005037  001170                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2681  010724  005037  001172                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2682  010730  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2683  010736  012737  010736  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2684  010744  012737  010744  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2685                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2686                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2687  010752  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2688  010756  012737  011012  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2689  010764  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2690  010772  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2691  011000  022777  177777  170132          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2692  011006  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2693                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
2694  011010  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
2695  011012  012716  011020          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2696  011016  000002                          RTI
2697  011020  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2698  011026  012737  000174  001142          MOV     #DISPREG,DISPLAY
2699  011034  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2700
2701  011040  005037  001212                  CLR     $PASS           ;;CLEAR PASS COUNT
2702  011044  132737  000200  001225          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2703  011052  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
2704  011054  012737  001226  001140          MOV     #SWREG,SWR      ;;NO,USE APT SWITCH REGISTER
```

```
2705  011062                                  67$:
2706
2707  011062  012737  011122  000004  MEMPAR: MOV     #1$,ERRVEC      ;TIMEOUT VECTOR
2708  011070  012737  000340  000006          MOV     #PR7,ERRVEC+2
2709
2710  011076  012737  000001  172100          MOV     #1,MEMBAS       ;LD REG TO DETERMINE IF
2711                                                                  ;MEMORY CHECK ENABLE AVAILABLE
2712  011104  012737  027654  000114          MOV     #MEMERR,MEMVEC  ;LD MEMORY CHK VECTOR IF DONT TIMEOUT
2713  011112  012737  000340  000116          MOV     #PR7,MEMVEC+2
2714  011120  000401                          BR      2$
2715
2716  011122  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
2717  011124  012737  000006  000004  2$:     MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
2718  011132  005037  000006                  CLR     ERRVEC+2
2719
2720  011136  004737  023730                  JSR     PC,CLRFLG       ;CLEAR DDUMP THRU SIZFLG
2721  011142  005037  001214                  CLR     $DEVCT
2722  011146  005037  001216                  CLR     $UNIT
2723
2724                                          ;
2725                                          ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
2726                                          ;
2727
2728  011152  005737  000042          START1: TST     42
2729  011156  001014                          BNE     1$              ;BR IF AUTO
2730  011160  004737  023750                  JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
2731  011164  123727  000041  000013          CMPB    41,#13          ;13=LOADED BY XXDP
2732  011172  001010                          BNE     2$
2733  011174  005237  005454                  INC     DDUMP           ;SET RK06 DUMP MODE FLAG
2734  011200  104401  034755                  TYPE    ,MSG2           ;REPLACE DRO PACK W/SCRATCH & DO<CR>
2735  011204  000137  011220                  JMP     ST2
2736  011210  000137  011264          1$:     JMP     ST3
2737  011214  005237  005462          2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG
2738
2739                                          ;
2740                                          ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
2741                                          ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
2742                                          ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
2743                                          ;       EX:     DRIVES TO BE TESTED:  1,2,4<CR>
2744                                          ;
2745
2746  011220  005737  001352          ST2:    TST     PARAM
2747  011224  001002                          BNE     1$              ;BR IF 220 START
2748  011226  000137  011316                  JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
2749  011232  104401  035026          1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
2750  011236  004737  023770                  JSR     PC,GDRVS        ;GET DR NOS.
2751  011242  104401  035060                  TYPE    ,MSG4           ;BUSS ADDR
2752  011246  004737  024130                  JSR     PC,GBA          ;GET BA
2753  011252  104401  035123                  TYPE    ,MSG5           ;CONT INT VECTOR
2754  011256  004737  024156                  JSR     PC,GINT         ;GET INT VECTOR
2755  011262  000427                          BR      ST5
2756
2757                                          ;
2758                                          ;AUTO MODE
2759                                          ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
2760                                          ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
```

```
2761                                    ;ON THE BUSS
2762                                    ;
2763
2764  011264  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
2765  011272  001007                    BNE     1$
2766  011274  005237  005456                  INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
2767  011300  004737  023750                  JSR     PC,TITLE
2768  011304  104401  035236                  TYPE    ,MSG7           ;DRO NOT TSTD
2769  011310  000402                    BR      ST4
2770  011312  005237  005460          1$:     INC     ACT11           ;SET ACT AUTO FLAG.
2771
2772  011316  012737  177440  001260  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
2773  011324  012737  000210  001330          MOV     #210,RKVEC      ;DEFAULT VALUE
2774  011332  004737  024210                  JSR     PC,SETINT
2775  011336  005237  005514                  INC     SIZFLG          ;DO "SIZE THE BUSS" TEST
2776
2777  011342  012706  001100          ST5:    MOV     #STACK,SP       ;INIT STACK
2778  011346  012746  000340                  MOV     #PR7,-(SP)      ;PSW LOADED TO BE
2779  011352  012746  011360                  MOV     #5$,-(SP)       ;LSI-11 COMPATABLE
2780  011356  000002                    RTI                     ;LOCK OUT ALL INTERRUPTS
2781  011360  012737  005466  001356  5$:     MOV     #DRIVD,DRVPTR   ;SETUP
2782  011366  005037  001214                  CLR     $DEVCT          ;NO. OF DRVS DONE
2783  011372  005037  001216                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
2784  011376  012737  011444  000004          MOV     #1$,ERRVEC              ;SETUP TIMEOUT ERROR VECTOR
2785  011404  005777  167732                  TST     @LKS            ;SEE IF L-CLOCK THERE
2786  011410  005237  005506                  INC     LCLKF           ;PRESENT, SET FLAG.
2787  011414  013700  001344                  MOV     LCVEC,R0        ;VECTOR ADDR
2788  011420  012737  011506  000004          MOV     #2$,ERRVEC
2789  011426  005777  167702                  TST     @PKS            ;SEE IF P-CLOCK THERE
2790  011432  005237  005510                  INC     PCLKF           ;PRESENT, SET FLAG
2791  011436  013700  001346                  MOV     PCVEC,R0        ;VECTOR ADDR
2792  011442  000412                    BR      3$
2793
2794  011444  022626                  1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
2795  011446  012737  011512  000004          MOV     #4$,ERRVEC
2796  011454  005777  167654                  TST     @PKS            ;SEE IF P-CLOCK THERE
2797  011460  005237  005510                  INC     PCLKF           ;PRESENT, SET FLAG
2798  011464  013700  001346                  MOV     PCVEC,R0        ;VECTOR ADDR
2799  011470  005237  005512          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
2800  011474  012720  027036                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
2801  011500  012710  000340                  MOV     #PR7,(R0)
2802  011504  000407                    BR      TST1            ;;GO TO NEXT TEST
2803
2804  011506  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
2805  011510  000767                    BR      3$
2806
2807  011512  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
2808  011514  005037  005512                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
2809  011520  104401  035445                  TYPE    ,MSG13          ;ALL TIMING TESTS BYPASSED
2810
2811
```

```
                                .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'
                                ;*
                                ;;*****************************************************************
011524  000004                TST1:   SCOPE
011526  012737  000001  001170        MOV     #1,$TIMES       ;;DO 1 ITERATION
011534  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR


011540  012737  011656  000004      MOV     #1$,ERRVEC              ;SETUP TIMOUT ERROR VECTOR
011546  013705  001260              MOV     $BASE,R5        ;SETUP INDEX REG.
011552  005765  000000              TST     RKCS1(R5)       ;REFERENCE ALL THE
011556  005765  000010              TST     RKCS2(R5)       ;CONTROLLER REGISTERS
011562  005765  000002              TST     RKWC(R5)
011566  005765  000004              TST     RKBA(R5)
011572  005765  000006              TST     RKDA(R5)
011576  005765  000012              TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
011602  005765  000014              TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
011606  005765  000016              TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
011612  005765  000020              TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
011616  005765  000024              TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
011622  005765  000026              TST     RKMR1(R5)
011626  005765  000034              TST     RKMR2(R5)
011632  005765  000036              TST     RKMR3(R5)
011636  005765  000030              TST     RKECPS(R5)
011642  005765  000032              TST     RKECPT(R5)

011646  012737  027500  000004      MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
011654  000404                      BR      TST2            ;;GO TO NEXT TEST

011656  022626              1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
011660  104007                      ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
011662  000137  023606              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.
```

```
2868                                        ;*
2869                                        ;;*****************************************************************
2870  011666  000004                       TST2:   SCOPE
2871  011670  012737  000001  001170               MOV     #1,$TIMES       ;;DO 1 ITERATION
2872  011676  012706  001100                       MOV     #STACK,SP       ;RESTORE STK PTR
2873
2874  011702  012765  000040  000010               MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
2875  011710  013737  001422  005442               MOV     T10,TEMP1       ;SET TIMEOUT
2876  011716  004737  024226                       JSR     PC,FRDY         ;FIND RDY
2877  011722  104120                             ERROR   120             ;RDY NOT SET BY END OF SCLR
2878  011724  005737  005514                       TST     SIZFLG
2879  011730  001562                             BEQ     TST3            ;;DO NOT SIZE, GOTO NEXT TEST
2880  011732  104401  035351                     TYPE    ,MSG10          ;WILL TEST DRIVES
2881  011736  005037  005464                     CLR     DRIVS           ;# OF DRIVES PRESENT
2882  011742  005000                             CLR     R0              ;DRV ADDR
2883  011744  012701  005466                     MOV     #DRIV0,R1       ;DRV FLAG
2884  011750                                   1$:
2885  011750  104415                             SCOP1
2886  011752  012706  001100                     MOV     #STACK,SP       ;RESTORE STK PTR
2887
2888  011756  012765  000040  000010               MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
2889  011764  013737  001422  005442               MOV     T10,TEMP1       ;SET TIMEOUT
2890  011772  004737  024226                       JSR     PC,FRDY         ;FIND RDY
2891  011776  104120                             ERROR   120             ;RDY NOT SET BY END OF SCLR
2892  012000  010065  000010                     MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
2893  012004  012765  000001  000000               MOV     #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
2894  012012  013737  001422  005442               MOV     T10,TEMP1
2895  012020  004737  024226                       JSR     PC,FRDY         ;FIND RDY
2896  012024  104117                             ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
2897  012026  032737  100000  005400               BIT     #CERR,HCS1
2898  012034  001046                             BNE     2$
2899  012036  013737  005426  005442               MOV     HMR2,TEMP1
2900  012044  042737  177770  005442               BIC     #^C<DRVMSK>,TEMP1
2901  012052  020037  005442                       CMP     R0,TEMP1        ;S/B SAME
2902  012056  001016                             BNE     3$
2903  012060  005700                             TST     R0
2904  012062  001003                             BNE     4$
2905  012064  005737  005456                     TST     DDPCH           ;SEE IF XXDP CHAIN MODE
2906  012070  001014                             BNE     5$
2907  012072  005237  005464             4$:     INC     DRIVS           ;INC DRIVE COUNT.
2908  012076  005211                             INC     (R1)            ;SET DRIVE PRESENT FLAG
2909  012100  104401  001201                     TYPE    ,$CRLF
2910  012104  010046                             MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2911                                                                       ;;TYPE DR #
2912  012106  104403                             TYPOS                   ;;GO TYPE--OCTAL ASCII
2913  012110     001                           .BYTE   1               ;;TYPE 1 DIGIT(S)
2914  012111     000                           .BYTE   0               ;;SUPPRESS LEADING ZEROS
2915  012112  000403                             BR      5$
2916
2917  012114  004737  024636             3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
2918  012120  104001                             ERROR   1               ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
2919
2920  012122  005721                     5$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR. FLAG
2921  012124  005200                             INC     R0              ;INC DR #
2922  012126  022700  000010                     CMP     #8.,R0
2923  012132  001306                             BNE     1$              ;MORE LEFT.
```

```
2924  012134  005737  005464          TST     DRIVS
2925  012140  001054                  BNE     10$
2926  012142  104401  035436          TYPE    ,MSG12          ;NONE
2927  012146  000137  023606          JMP     $EOP
2928
2929  012152  032737  001000  005402  2$:     BIT     #MDS,HCS2
2930  012160  001015                  BNE     6$
2931  012162  032737  000400  005402          BIT     #UFE,HCS2
2932  012170  001015                  BNE     7$
2933  012172  032737  000001  005412          BIT     #DRA,HDS
2934  012200  001015                  BNE     8$
2935  012202  032737  010000  005402          BIT     #NED,HCS2
2936  012210  001424                  BEQ     9$
2937  012212  000743                  BR      5$
2938
2939  012214  004737  024636          6$:     JSR     PC,BYP          ;TYPE BYP DR #
2940  012220  104002                  ERROR   2               ;MDS DETECTED
2941  012222  000737                  BR      5$
2942
2943  012224  004737  024636          7$:     JSR     PC,BYP
2944  012230  104003                  ERROR   3               ;UFE DETECTED
2945  012232  000733                  BR      5$
2946
2947  012234  032737  010000  005402  8$:     BIT     #NED,HCS2
2948  012242  001713                  BEQ     4$
2949  012244  104401  035556          TYPE    ,MSG15          ;DRV#
2950  012250  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2951                                                          ;;TYPE DR#
2952  012252  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
2953  012254     001                  .BYTE   1               ;;TYPE 1 DIGIT(S)
2954  012255     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
2955  012256  104010                  ERROR   10              ;DRA & NED BOTH SET
2956  012260  000720                  BR      5$
2957
2958  012262  004737  024636          9$:     JSR     PC,BYP
2959  012266  104004                  ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
2960  012270  000714                  BR      5$
2961  012272  000137  012622          10$:    JMP     UNLDRV
2962
2963                                  ;;*****************************************************************
2964                                  ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
2965                                  ;*
2966                                  ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
2967                                  ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
2968                                  ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
2969                                  ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
2970                                  ;*      TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
2971                                  ;*      BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
2972                                  ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
2973                                  ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
2974                                  ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
2975                                  ;*      VERIFY IT WAS NOT SPECIFIED.
2976                                  ;*
2977                                  ;*****************************************************************
2978  012276  000004                  TST3:   SCOPE
2979  012300  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
2980  012306  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
2981  012312  005000                          CLR     R0              ;DRIVE ADDR
2982  012314  012701  005466                  MOV     #DRIVD,R1       ;DRIVE FLAG
2983  012320                          1$:
2984  012320  104415                          SCOP1
2985  012322  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
2986
2987  012326  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
2988  012334  013737  001422  005442          MOV     T10,TEMP1       ;SET TIME OUT
2989  012342  004737  024226                  JSR     PC,FRDY         ;FIND RDY
2990  012346  104120                          ERROR   120             ;NO RDY AFTER SCLR
2991  012350  010065  000010                  MOV     R0,RKCS2(R5)    ;DRV ADDR
2992  012354  012765  000001  000000          MOV     #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
2993  012362  013737  001422  005442          MOV     T10,TEMP1
2994  012370  004737  024226                  JSR     PC,FRDY         ;FIND RDY
2995  012374  104117                          ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
2996  012376  032737  100000  005400          BIT     #CERR,HCS1
2997  012404  001036                          BNE     2$
2998  012406  013737  005426  005442          MOV     HMR2,TEMP1
2999  012414  042737  177770  005442          BIC     #^C<DRVMSK>,TEMP1
3000  012422  020037  005442                  CMP     R0,TEMP1        ;S/B SAME
3001  012426  001010                          BNE     3$
3002  012430  005711                  11$:    TST     (R1)
3003  012432  001417                          BEQ     5$
3004  012434  005721                  4$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR FLAG
3005  012436  005200                          INC     R0              ;INC DR#
3006  012440  022700  000010                  CMP     #8.,R0
3007  012444  001325                          BNE     1$              ;MORE LEFT
3008  012446  000465                          BR      TST4            ;;GO TO NEXT TEST
3009
3010  012450  004737  024636          3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
3011  012454  104001                          ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
3012  012456  005711                          TST     (R1)
3013  012460  001765                          BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
3014  012462  005337  005464          12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
3015  012466  005011                          CLR     (R1)            ;CLEAR DRIVE FLAG
3016  012470  000761                          BR      4$
3017
3018  012472  004737  024636          5$:     JSR     PC,BYP
3019  012476  104005                          ERROR   5               ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
3020  012500  000755                          BR      4$
3021
3022  012502  032737  001000  005402  2$:     BIT     #MDS,HCS2
3023  012510  001027                          BNE     6$
3024  012512  032737  000400  005402          BIT     #UFE,HCS2
3025  012520  001027                          BNE     7$
3026  012522  032737  000001  005412          BIT     #DRA,HDS
3027  012530  001005                          BNE     8$
3028  012532  032737  010000  005402          BIT     #NED,HCS2
3029  012540  001423                          BEQ     9$
3030  012542  000404                          BR      10$
3031  012544  032737  010000  005402  8$:     BIT     #NED,HCS2
3032  012552  001726                          BEQ     11$
3033  012554  005711                  10$:    TST     (R1)
3034  012556  001726                          BEQ     4$
3035
```

```
3036  012560  004737  024636          JSR     PC,BYP          ;TYPE BYPASS DRIVE#
3037  012564  104006                  ERROR   6
3038  012566  000735                  BR      12$
3039
3040  012570  004737  024636  6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
3041  012574  104002                  ERROR   2               ;MDS DETECTED
3042  012576  000762                  BR      8$
3043
3044  012600  004737  024636  7$:     JSR     PC,BYP
3045  012604  104003                  ERROR   3               ;UFE DETECTED
3046  012606  000756                  BR      8$
3047
3048  012610  004737  024636  9$:     JSR     PC,BYP
3049  012614  104004                  ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
3050  012616  000752                  BR      8$
3051
3052  012620  001237                  BNE     1$              ;BRANCH IF MORE LEFT.
3053
3054  012622                  UNLDRV:
3055
3056                          ;;*****************************************************************
3057                          ;*TEST 4        UNLOAD ALL DRIVES TO BE TESTED
3058                          ;*
3059                          ;*      THIS TEST UNLOADS ALL THE DRIVES TO BE TESTED,
3060                          ;*      WAITS FOR ATTN & VERIFIES IT CAME BACK FROM THE CORRECT DRIVE
3061                          ;*      WITHIN APPROX. 100 USEC.
3062                          ;*      ATTN, DRIVE READY & DRIVE STATUS CHANGES ARE VERIFIED TO BE CORRECT
3063                          ;*
3064                          ;;*****************************************************************
3065  012622  000004          TST4:   SCOPE
3066  012624  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
3067
3068  012632  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
3069  012636  005000                  CLR     R0              ;DRV ADDR
3070  012640  012701  005466          MOV     #DRIV0,R1       ;DRV FLAG
3071
3072  012644                  1$:
3073  012644  104415                  SCOP1
3074  012646  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
3075
3076  012652  012765  000040  000010          MOV     #SCLR,RKCS2(R5)
3077  012660  013737  001422  005442          MOV     T10,TEMP1
3078  012666  004737  024226                  JSR     PC,FRDY         ;FIND RDY
3079  012672  104120                  ERROR   120             ;NO RDY AFTER SCLR
3080  012674  005721                  TST     (R1)+
3081  012676  001436                  BEQ     4$
3082  012700  010065  000010          MOV     R0,RKCS2(R5)    ;SELECT DRV ADDR.
3083  012704  012765  000007  000000          MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
3084  012712  013737  001422  005442          MOV     T10,TEMP1
3085  012720  004737  024226                  JSR     PC,FRDY         ;FIND RDY
3086  012724  104017                  ERROR   17              ;RDY NOT SET AFTER UNLOAD CMD
3087  012726  105737  005417          TSTB    HASOF+1         ;TEST FOR ANY ATTN BITS
3088  012732  001001                  BNE     5$              ;BR IF ANY
3089  012734  104020                  ERROR   20              ;NO ATTN AFTER UNLOAD CMD
3090  012736  136037  005370  005417  5$:     BITB    ATTN(R0),HASOF+1        ;TEST FOR SPECIFIC ATTN
3091  012744  001001                  BNE     2$              ;BR IF THERE
```

```
                                        ERROR   41              ;WRONG ATTN AFTER UNLOAD CMD
012746  104041
012750  032737  000200  005426  2$:     BIT     #D.DRDY,HMR2
012756  001401                          BEQ     3$
012760  104042                          ERROR   42              ;D.DRDY SET AFTER UNLOAD CMD
012762  032737  040000  005426  3$:     BIT     #D.DSC,HMR2
012770  001001                          BNE     4$
012772  104043                          ERROR   43              ;D.DSC NOT SET AFTER UNLOAD CMD

012774  005200                  4$:     INC     R0              ;INCR DRIVE ADDR.
012776  020027  000010                  CMP     R0,#8.
013002  001320                          BNE     1$              ;BRANCH IF MORE LEFT.

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

013004                          NUDRV:                          ;ENTER HERE FROM LAST TEST

                                ;;*****************************************************************
                                ;*TEST 5        FIND NEXT DRIVE TO BE TESTED
                                ;*
                                ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                ;*      ADDRESS IN '$UNIT'.
                                ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                                ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
                                ;*
                                ;;*****************************************************************
013004  000004           TST5:  SCOPE
013006  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
013014  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
013020  012737  000005  001210          MOV     #$TN-1,$TESTN
013026  012737  000005  001102          MOV     #$TN-1,$TSTNM

013034  005737  005464                  TST     DRIVS           ;ANY DRIVES PRESENT?
013040  001004                          BNE     4$              ;YES BRANCH
013042  104401  036273                  TYPE    ,MSG27          ;ALL DRIVES TESTED
013046  000137  023606                  JMP     $EOP            ;NO, GO TO END

013052  013701  001356          4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
013056  005737  001214                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
013062  001402                          BEQ     2$              ;YES, BRANCH
013064  005237  001216          1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
013070  005721                  2$:     TST     (R1)+           ;IS DRIVE PRESENT?
013072  001774                          BEQ     1$              ;NO,FIND NEXT DRIVE PRESENT
013074  005737  005456                  TST     DDPCH           ;DDP CHAIN MODE?
013100  001403                          BEQ     3$              ;NO, BRANCH
013102  005737  001216                  TST     $UNIT           ;YES, IS IT DRIVE 0?
013106  001766                          BEQ     1$              ;IF YES, DON'T TEST DR 0
013110  010137  001356          3$:     MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT DR. FLAG
013114  104401  035556                  TYPE    ,MSG15          ;"DRIVE"
013120  013700  001216                  MOV     $UNIT,R0
013124  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                                ;;DRIVE #
```

```
                        TYPOS                   ;;GO TYPE--OCTAL ASCII
                        .BYTE   1               ;;TYPE 1 DIGIT(S)
                        .BYTE   0               ;;SUPPRESS LEADING ZEROS

                        TYPE    ,$CRLF

                PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
                ;;*************************************************************
                ;*TEST 6         PRINT DRIVE SERIAL NUMBER
                ;*
                ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
                ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
                ;*
                ;*************************************************************
013136 000004   TST6:   SCOPE
013140 012737 000001 001170     MOV     #1,$TIMES       ;;DO 1 ITERATION
013146 012706 001100            MOV     #STACK,SP       ;RESTORE STK PTR

013152 005737 001212            TST     $PASS
013156 001046                   BNE     TST7            ;;GO TO NEXT IF NOT FIRST PASS
013160 004737 024714            JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
013164 104024                   ERROR   24              ;CERR AFTER SCLR

013166 104401 035570            TYPE    ,MSG16          ;DRIVE SERIAL NO.
013172 012765 000003 000026     MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
013200 004737 024652            JSR     PC,GSTAT        ;GET STATUS
013204 013701 005426            MOV     HMR2,R1         ;GET SERIAL #
013210 012704 033256            MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
013214 010446                   MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
013216 012703 000003            MOV     #3,R3           ;SETUP CHAR COOUNT
013222 006101                   ROL     R1              ;INITIALIZE BIT POSITIONS
013224 006101                   ROL     R1
013226 006101           1$:     ROL     R1              ;GET NEXT 4 BITS
013230 006101                   ROL     R1
013232 006101                   ROL     R1
013234 006101                   ROL     R1
013236 010100                   MOV     R1,R0           ;GET WORKING COPY
013240 042700 177760            BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
013244 052700 000060            BIS     #60,R0          ;CONVERT TO ASCII DIGIT
013250 110024                   MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
013252 005303                   DEC     R3
013254 001364                   BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
013256 105014                   CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR

013260 004737 033524            JSR     PC,$SUPRS       ;TYPE
013264 104401 001201            TYPE    ,$CRLF
013270 104401 001201            TYPE    ,$CRLF

                ;;*************************************************************
                ;*TEST 7         SET VV WITH PACK COMMAND
                ;*
                ;*      IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
                ;*
                ;*************************************************************
013274 000004   TST7:   SCOPE
013276 012737 000001 001170     MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
3204  013304  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
3205
3206  013310  004737  024714          JSR     PC,SUBCLR
3207  013314  104024                  ERROR   24              ;CERR AFTER SCLR
3208
3209  013316  032737  000100  005426  BIT     #D.VV,HMR2
3210  013324  001024                  BNE     TST10           ;;GO TO NEXT TEST IF VV SET
3211
3212  013326  104415                  SCOP1
3213  013330  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
3214
3215  013334  004737  024714          JSR     PC,SUBCLR
3216  013340  104024                  ERROR   24              ;CERR AFTER SCLR
3217
3218  013342  012765  000003  000000  MOV     #PACK,RKCS1(R5) ;CMD TO SET VV
3219  013350  012737  000010  005442  MOV     #10,TEMP1
3220  013356  004737  024226          JSR     PC,FRDY         ;FIND RDY
3221  013362  104116                  ERROR   116             ;RDY NOT SET AFTER PACK CMD
3222
3223  013364  032737  000100  005426  BIT     #D.VV,HMR2
3224  013372  001001                  BNE     TST10           ;;GO TO NEXT TEST IF VV NOW SET
3225  013374  104027                  ERROR   27              ;PACK DID NOT SET V.V.
3226
3227
3228                          .SBTTL OPERATOR INTERVENTION TESTS
3229
3230                          ;;*****************************************************************
3231                          ;*TEST 10       INTERLOCKS TESTS
3232                          ;*
3233                          ;*      THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
3234                          ;*      ARE OPERATING PROPERLY IN MESSAGE A0 & THAT THE REMOVAL
3235                          ;*      OF THE CARTRIDGE CLEARS VOLUME VALID.
3236                          ;*      IT FURTHUR VERIFIES ALL REMAINING STATUS BITS OF
3237                          ;*      MESSAGE A & B, WORDS 0 & 1.
3238                          ;*      THIS TEST ALSO CHECKS THAT THE SPINDLE CANNOT BE STARTED
3239                          ;*      WHEN THE DOOR IS OPEN OR THE CARTRIDGE IS REMOVED.
3240                          ;*      IT ALSO VERIFIES THAT LOSS OF VOLUME VALID RESETS SACK WHICH
3241                          ;*      ASSERTS NON EXISTENT DRIVE IN RKCS2
3242                          ;*
3243                          ;;*****************************************************************
3244  013376  000004          TST10:  SCOPE
3245  013400  012737  000001  001170  MOV     #1,$TIMES       ;;DO 1 ITERATION
3246  013406  012706  001100          MOV     #STACK,SP       ;RESET STACK PTR
3247
3248  013412  004737  024714          JSR     PC,SUBCLR
3249  013416  104024                  ERROR   24              ;CERR AFTER SCLR
3250
3251  013420  104401  035273          TYPE    ,MSG8           ;INTERLOCKS TEST
3252  013424  104401  040671          TYPE    ,MSG52          ;CONT-C TO EXIT OR SPACE TO CONTINUE
3253  013430  004737  027412          JSR     PC,CCSP         ;INPUT CONT-C OR SPACE
3254  013434  000137  014504          JMP     8$              ;RET HERE FOR CONT-C
3255
3256  013440  104401  040177          TYPE    ,MSG47          ;VERIFY DOOR CANNOT BE OPENED
3257  013444  104401  037126          TYPE    ,MSG37          ;DEPRESS SPACE WHEN DONE
3258  013450  004737  027452          JSR     PC,GETSP        ;GET SPACE
3259
```

```
3260  013454  104401  042467                    TYPE    ,MSG65          ;DEPRESS RUN/STOP SW TO 'STOP'
3261  013460  104401  036431                    TYPE    ,MSG30          ;OPEN DOOR & LEAVE IT OPEN
3262  013464  104401  037126                    TYPE    ,MSG37          ;DEPRESS SPACE BAR  WHEN DONE
3263  013470  004737  027452                    JSR     PC,GETSP        ;INPUT  SPACE
3264  013474  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3265  013502  004737  024652                    JSR     PC,GSTAT
3266  013506  032737  000200  005426            BIT     #D.DOOR,HMR2
3267  013514  001401                            BEQ     1$
3268  013516  104044                            ERROR   44              ;DOOR STATUS BIT NOT CLEARED
3269
3270  013520  012765  100000  000000  1$:       MOV     #CCLR,RKCS1(R5)
3271  013526  005065  000026                    CLR     RKMR1(R5)       ;CHECK WORD 0 OF MSG A & B
3272  013532  004737  024652                    JSR     PC,GSTAT
3273  013536  012737  000140  005436            MOV     #<D.DRA!D.VV>,SBMR2
3274  013544  004737  027126                    JSR     PC,CKMR2
3275  013550  104045                            ERROR   45              ;MSG A0 ERROR AFTER DRIVE UNLOADED
3276                                            ;& DOOR OPENED
3277  013552  005037  005440                    CLR     SBMR3
3278  013556  004737  027210                    JSR     PC,CKMR3
3279  013562  104046                            ERROR   46              ;MSG B0 ERROR
3280
3281  013564  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
3282  013572  012765  000001  000026            MOV     #1,RKMR1(R5)    ;CHECK WORD 1 OF MSG A & B
3283  013600  004737  024652                    JSR     PC,GSTAT
3284  013604  012737  000540  005436            MOV     #<D.HDHM!D.BRHM!D.CART>,SBMR2
3285  013612  004737  027126                    JSR     PC,CKMR2
3286  013616  104063                            ERROR   63              ;MSG A1 ERROR AFTER DRIVE UNLOADED
3287                                            ;& DOOR OPENED
3288  013620  005037  005440                    CLR     SBMR3
3289  013624  004737  027210                    JSR     PC,CKMR3
3290  013630  104064                            ERROR   64              ;MSG B1 ERROR
3291
3292  013632  004737  024714                    JSR     PC,SUBCLR
3293  013636  104024                            ERROR   24              ;CERR AFTER SCLR
3294
3295  013640  104401  036561                    TYPE    ,MSG33          ;PRESS 'RUN-STOP' TO 'RUN'
3296  013644  104401  036646                    TYPE    ,MSG34          ;VERIFY DOES NOT START
3297  013650  104401  037126                    TYPE    ,MSG37
3298  013654  004737  027452                    JSR     PC,GETSP        ;GET SPACE FROM TTY
3299
3300  013660  004737  024652                    JSR     PC,GSTAT
3301  013664  032737  010000  005426            BIT     #D.SPIN,HMR2
3302  013672  001401                            BEQ     2$
3303  013674  104065                            ERROR   65              ;SPIN SET IN MSGA0
3304
3305  013676  012737  000140  005436  2$:       MOV     #<D.DRA!D.VV>,SBMR2
3306  013704  004737  027126                    JSR     PC,CKMR2
3307  013710  104066                            ERROR   66              ;MSG A0 ERROR AFTER ATTEMPTING TO
3308                                            ;START SPINDLE WITH DOOR OPEN.
3309  013712  005037  005440                    CLR     SBMR3
3310  013716  004737  027210                    JSR     PC,CKMR3
3311  013722  104067                            ERROR   67              ;MSG B0 ERROR.
3312
3313  013724  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
3314  013732  012765  000001  000026            MOV     #1,RKMR1(R5)    ;CHECK WORD 1 OF MSG A & B
3315  013740  004737  024652                    JSR     PC,GSTAT
```

```
3316  013744  012737  000540  005436        MOV     #<D.HDHM!D.BRHM!D.CART>,SBMR2
3317  013752  004737  027126                JSR     PC,CKMR2
3318  013756  104070                        ERROR   70                ;MSG A1 ERROR AFTER ATTEMPTING TO
3319                                                                  ;START SPINDLE WITH DOOR OPEN
3320  013760  005037  005440                CLR     SBMR3
3321  013764  004737  027210                JSR     PC,CKMR3
3322  013770  104071                        ERROR   71                ;MSG B1 ERROR
3323
3324  013772  004737  024714                JSR     PC,SUBCLR
3325  013776  104024                        ERROR   24                ;CERR AFTER SCLR
3326
3327  014000  104401  042467                TYPE    ,MSG65            ;DEPRESS 'RUN-STOP' SW TO STOP
3328  014004  104401  036521                TYPE    ,MSG32            ;REMOVE PACK & CLOSE DOOR
3329  014010  104401  037126                TYPE    ,MSG37
3330  014014  004737  027452                JSR     PC,GETSP          ;GET SPACE FROM TTY
3331
3332  014020  012765  000001  000026        MOV     #1,RKMR1(R5)      ;SELECT WORD 1
3333  014026  004737  024652                JSR     PC,GSTAT
3334  014032  032737  000400  005426        BIT     #D.CART,HMR2
3335  014040  001401                        BEQ     3$
3336  014042  104072                        ERROR   72                ;CARTRIDGE STATUS BIT NOT RESET
3337
3338  014044  012765  100000  000000  3$:   MOV     #CCLR,RKCS1(R5)
3339  014052  005065  000026                CLR     RKMR1(R5)         ;CHECK WORD 0 OF MSG A & B
3340  014056  004737  024652                JSR     PC,GSTAT
3341  014062  032737  000100  005426        BIT     #D.VV,HMR2
3342  014070  001401                        BEQ     4$
3343  014072  104073                        ERROR   73                ;VV NOT RESET AFTER CARTRIDGE REMOVED
3344
3345  014074  012737  000040  005436  4$:   MOV     #D.DRA,SBMR2
3346  014102  004737  027126                JSR     PC,CKMR2
3347  014106  104074                        ERROR   74                ;MSG A0 ERROR AFTER DRIVE UNLOADED
3348                                                                  ;& CARTRIDGE REMOVED
3349
3350  014110  005037  005440                CLR     SBMR3
3351  014114  004737  027210                JSR     PC,CKMR3
3352  014120  104075                        ERROR   75                ;MSG B0 ERROR
3353
3354  014122  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3355  014130  012765  000001  000026        MOV     #1,RKMR1(R5)      ;CHECK WORD 1 OF MSG A & B
3356  014136  004737  024652                JSR     PC,GSTAT
3357  014142  012737  000140  005436        MOV     #<D.HDHM!D.BRHM>,SBMR2  ;DOOR = 0 WHEN RUN-STOP SW IN STOP
3358  014150  004737  027126                JSR     PC,CKMR2
3359  014154  104076                        ERROR   76                ;MSG A1 ERROR AFTER DRIVE UNLOADED
3360                                                                  ;& CARTRIDGE REMOVED
3361  014156  005037  005440                CLR     SBMR3
3362  014162  004737  027210                JSR     PC,CKMR3
3363  014166  104077                        ERROR   77                ;MSG B1 ERROR.
3364
3365  014170  004737  024714                JSR     PC,SUBCLR
3366  014174  104024                        ERROR   24                ;CERR AFTER SCLR
3367
3368  014176  104401  036734                TYPE    ,MSG35            ;PRESS 'RUN-STOP' TO 'RUN'
3369  014202  104401  036646                TYPE    ,MSG34            ;VERIFY DOES NOT START
3370  014206  104401  037126                TYPE    ,MSG37
3371  014212  004737  027452                JSR     PC,GETSP          ;GET SPACE FROM TTY
```

```
3372
3373  014216  004737  024652          JSR     PC,GSTAT
3374  014222  032737  010000  005426  BIT     #D.SPIN,HMR2
3375  014230  001401                  BEQ     5$
3376  014232  104100                  ERROR   100             ;SPIN SET IN MSG A0
3377
3378  014234  012737  000040  005436  5$:     MOV     #D.DRA,SBMR2
3379  014242  004737  027126          JSR     PC,CKMR2
3380  014246  104101                  ERROR   101             ;MSG A0 ERROR AFTER ATTEMPTING TO
3381                                                          ;START SPINDLE WITH CARTRIDGE REMOVED
3382  014250  005037  005440          CLR     SBMR3
3383  014254  004737  027210          JSR     PC,CKMR3
3384  014260  104102                  ERROR   102             ;MSG B0 ERROR
3385
3386  014262  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
3387  014270  012765  000001  000026  MOV     #1,RKMR1(R5)    ;CHECK WORD 1 OF MSG A & B
3388  014276  004737  024652          JSR     PC,GSTAT
3389  014302  012737  000340  005436  MOV     #<D.HDHM!D.BRHM!D.DOOR>,SBMR2
3390  014310  004737  027126          JSR     PC,CKMR2
3391  014314  104103                  ERROR   103             ;MSG A1 ERROR AFTER ATTEMPTING TO
3392                                                          ;START SPINDLE WITH CARTRIDGE REMOVED
3393  014316  005037  005440          CLR     SBMR3
3394  014322  004737  027210          JSR     PC,CKMR3
3395  014326  104104                  ERROR   104             ;MSG B1 ERROR
3396
3397  014330  104401  042467          TYPE    ,MSG65          ;PRESS 'RUN-STOP' TO 'STOP'
3398  014334  104401  037031          TYPE    ,MSG36          ;INSERT CARTRIDGE & CLOSE DOOR
3399  014340  104401  043052          TYPE    ,MSG70          ;VERIFY HEADS LOAD
3400  014344  104401  042775          TYPE    ,MSG69          ;DEPRESS SPACE WHEN READY GOES ON
3401  014350  004737  027452          JSR     PC,GETSP        ;GET SPACE FROM TTY
3402
3403  014354  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
3404  014362  005065  000026          CLR     RKMR1(R5)       ;SELECT WORD 0
3405  014366  004737  024652          JSR     PC,GSTAT
3406  014372  032737  000100  005426  BIT     #D.VV,HMR2
3407  014400  001401                  BEQ     7$
3408  014402  104106                  ERROR   106             ;VOLUME VALID SET WITHOUT PACK CMD
3409
3410  014404                  7$:
3411  014404  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
3412  014412  012765  000003  000000  MOV     #PACK,RKCS1(R5) ;PACK CMD
3413  014420  013737  001422  005442  MOV     T10,TEMP1
3414  014426  004737  024226          JSR     PC,FRDY         ;FIND CONTR RDY
3415  014432  104116                  ERROR   116             ;CONTR NOT RDY
3416
3417  014434  032737  000100  005426  BIT     #D.VV,HMR2
3418  014442  001001                  BNE     64$
3419  014444  104027                  ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
3420  014446                  64$:
3421
3422  014446  004737  024714          JSR     PC,SUBCLR
3423  014452  104024                  ERROR   24              ;CERR AFTER SCLR
3424
3425  014454  012765  000007  000000  MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
3426  014462  013737  001422  005442  MOV     T10,TEMP1
3427  014470  004737  024226          JSR     PC,FRDY         ;FIND CONTR RDY
```

```
014474  104017                          ERROR   17              ;NO RDY AFTER UNLD CMD
014476  004737  024412                  JSR     PC,TSTATN
014502  104020                          ERROR   20              ;NO ATTN AFTER UNLOAD CMD
014504                          B$:

                                ;;*****************************************************************
                                ;*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
                                ;*
                                ;;*****************************************************************
014504  000004                TST11:  SCOPE
014506  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
014514  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR

014520  004737  024714                  JSR     PC,SUBCLR
014524  104024                          ERROR   24              ;CERR AFTER SCLR

014526  104401  035644                  TYPE    ,MSG18          ;UNIT SELECT PLUG TEST
014532  104401  040671                  TYPE    ,MSG52          ;CONT-C TO EXIT OR SPACE TO CONTINUE
014536  004737  027412                  JSR     PC,CCSP         ;INPUT CONT-C OR SPACE
014542  000137  015312                  JMP     12$             ;RETURN HERE FOR CONT-C

014546  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY TO GET DSC & ATTN
014554  004737  024652                  JSR     PC,GSTAT
014560  012737  001200  005440          MOV     #<D.FLT!D.PAR>,SBMR3
014566  004737  027210                  JSR     PC,CKMR3
014572  104114                          ERROR   114             ;MSG 80 ERROR AFTER SEL DRV WITH EVEN PARITY ISSUED

014574  032737  040000  005426          BIT     #D.DSC,HMR2
014602  001001                          BNE     1$
014604  104107                          ERROR   107             ;DSC NOT SET AFTER SEL DRV WITH EVEN PARITY

014606  012765  100000  000000  1$:     MOV     #CCLR,RKCS1(R5)
014614  004737  024652                  JSR     PC,GSTAT
014620  004737  024412                  JSR     PC,TSTATN
014624  104113                          ERROR   113             ;NO ATTN AFTER SEL DRV WITH EVEN PARITY

014626  104401  035317                  TYPE    ,MSG9           ;REMOVE UNIT SELECT PLUG
014632  104401  037126                  TYPE    ,MSG37          ;DEPRESS SPACE WHEN DONE
014636  004737  027452                  JSR     PC,GETSP        ;GET SPACE

014642  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
014650  004737  024652                  JSR     PC,GSTAT
014654  004737  024412                  JSR     PC,TSTATN       ;RETURN HERE FOR SPACE
014660  000401                          BR      3$              ;NO ATTN
014662  104111                          ERROR   111             ;REMOVING UNIT SEL PLUG, DID NOT
                                                                ;DISABLE ATTN.
014664  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
014672  004737  024652                  JSR     PC,GSTAT
```

```
3484  014676  032765  010000  000010          BIT     #NED,RKCS2(R5)
3485  014704  001001                          BNE     4$
3486  014706  104112                          ERROR   112             ;REMOVING UNIT SEL PLUG DID NOT
3487                                                                  ;ASSERT NON-EXISTENT DRIVE
3488
3489  014710  104401  042660          4$:     TYPE    ,MSG67          ;RDESELECT ALL OTHER PORTS
3490  014714  104401  037126                  TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
3491  014720  004737  027452                  JSR     PC,GETSP        ;GET SPACE
3492
3493  014724  104401  037172                  TYPE    ,MSG38          ;INSERT UNIT SELECT PLUGS
3494  014730  104401  037330                  TYPE    ,MSG39          ;DEPRESS CONTROL-C WHEN FINISHED
3495  014734  104401  041152                  TYPE    ,MSG55          ;EXIT WITH CORRECT UNIT SELECT PLUG #
3496  014740  013746  001216                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
3497                                                                  ;;TYPE CORRECT#
3498  014744  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3499  014746     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3500  014747     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3501  014750  104401  001201                  TYPE    ,$CRLF
3502
3503  014754  105777  164164          5$:     TSTB    @$TKS           ;SEE IF CHAR TYPED
3504  014760  100014                          BPL     6$              ;BR IF NO
3505  014762  117737  164160  001160          MOVB    @$TKB,$TMP0     ;ELSE READ CHAR
3506  014770  042737  177600  001160          BIC     #^C<177>,$TMP0  ;GET RID OF JUNK, IF ANY
3507  014776  023727  001160  000003          CMP     $TMP0,#CONTC    ;SEE IF CONTROL-C
3508  015004  001462                          BEQ     9$              ;BR IF YES
3509  015006  104401  036513                  TYPE    ,MSG31          ;?
3510
3511  015012  005000                  6$:     CLR     R0
3512  015014  012765  100000  000000  7$:     MOV     #CCLR,RKCS1(R5)
3513  015022  010065  000010                  MOV     R0,RKCS2(R5)    ;DRIVE NO.
3514  015026  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;SELECT DRIVE CMD
3515  015034  013737  001422  005442          MOV     T10,TEMP1
3516  015042  004737  024226                  JSR     PC,FRDY         ;FIND CONTR RDY
3517  015046  104117                          ERROR   117             ;CONTR RDY NOT SET
3518
3519  015050  032765  010000  000010          BIT     #NED,RKCS2(R5)  ;SEE IF UNIT SELECT PLUG INSERTED
3520  015056  001405                          BEQ     8$              ;& IF MATCH DRIVE NO IN CS2. BR IF YES
3521  015060  005200                          INC     R0
3522  015062  020027  000010                  CMP     R0,#8.          ;ALL 8 DRIVE NOS TRIED?
3523  015066  001352                          BNE     7$              ;BR IF NO
3524  015070  000731                          BR      5$              ;ELSE RETURN
3525
3526  015072  032737  000100  005426  8$:     BIT     #D.VV,HMR2
3527  015100  001325                          BNE     5$              ;TYPE SAME UNIT SELECT PLUG RDY ONCE
3528  015102  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3529  015110  010065  000010                  MOV     R0,RKCS2(R5)    ;DRIVE ADDR
3530  015114  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
3531  015122  013737  001422  005442          MOV     T10,TEMP1
3532  015130  004737  024226                  JSR     PC,FRDY         ;FIND CONTR RDY
3533  015134  104116                          ERROR   116             ;CONTR NOT RDY
3534  015136  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3535                                                                  ;;TYPE UNIT SEL PLUG NO.
3536  015140  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3537  015142     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3538  015143     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3539  015144  104401  001201                  TYPE    ,$CRLF
```

```
3540  015150  000701                          BR      5$
3541
3542  015152  012765  100000  000000  9$:     MOV     #CCLR,RKCS1(R5)
3543  015160  013765  001216  000010          MOV     $UNIT,RKCS2(R5)
3544  015166  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
3545  015174  013737  001422  005442          MOV     T10,TEMP1
3546  015202  004737  024226                  JSR     PC,FRDY
3547  015206  104117                          ERROR   117             ;CONT NOT RDY AFTER SEL DRV CMD
3548
3549  015210  032765  010000  000010          BIT     #NED,RKCS2(R5)  ;SEE IF CORRECT PLUG FOR EXIT
3550  015216  001411                          BEQ     10$             ;BR IF YES
3551  015220  104401  041152                  TYPE    ,MSG55          ;EXIT WITH CORRECT UNIT SELECT PLUG NO.
3552  015224  013746  001216                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
3553                                                                  ;;TYPE CORRECT UNIT SEL PLUG #
3554  015230  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3555  015232     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3556  015233     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3557  015234  104401  037330                  TYPE    ,MSG39          ;DEPRESS CONT-C WHEN DONE
3558  015240  000645                          BR      5$
3559
3560  015242  004737  024714          10$:    JSR     PC,SUBCLR
3561  015246  104024                          ERROR   24              ;CERR AFTER SCLR
3562
3563  015250  004737  024652                  JSR     PC,GSTAT
3564  015254  032737  000100  005426          BIT     #D.VV,HMR2
3565  015262  001005                          BNE     11$
3566  015264  104401  037373                  TYPE    ,MSG40          ;VV NOT SET, INSERT UNIT SELECT PLUG
3567  015270  104401  037330                  TYPE    ,MSG39          ;DEPRESS CONT-C TO EXIT TEST
3568  015274  000627                          BR      5$
3569
3570  015276  104401  043100          11$:    TYPE    ,MSG71          ;SELECT CORRECT PORT ON OTHER DRIVES
3571  015302  104401  037126                  TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
3572  015306  004737  027452                  JSR     PC,GETSP        ;GET SPACE
3573  015312                          12$:
3574
3575
3576                                          ;;*****************************************************************
3577                                          ;*TEST 12       PORT SELECTION TESTS
3578                                          ;*
3579                                          ;*      THE OPERATOR IS ASKED TO SWITCH TO THE WRONG PORT
3580                                          ;*      & THEN DESELECT BOTH PORTS.
3581                                          ;*      IN BOTH CASES, NON EXISTENT DRIVE SHOULD ASSERT IN RKCS2
3582                                          ;*
3583                                          ;;*****************************************************************
3584  015312  000004                  TST12:  SCOPE
3585  015314  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
3586  015322  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR
3587
3588  015326  004737  024714                  JSR     PC,SUBCLR
3589  015332  104024                          ERROR   24              ;CERR AFTER SCLR
3590
3591  015334  104401  035676                  TYPE    ,MSG19          ;PORT SELECTION TESTS
3592  015340  104401  040671                  TYPE    ,MSG52          ;DEPRESS CONTROL-C TO EXIT OR SPACE BAR TO CONTINUE
3593  015344  004737  027412                  JSR     PC,CCSP         ;INPUT CONTROL-C OR SPACE
3594  015350  000137  015530                  JMP     4$              ;RETURN HERE FOR CONTROL-C
3595
```

```
3596  015354  104401  037505                  TYPE    ,MSG41          ;SWITCH TO OTHER PORT
3597  015360  104401  037126                  TYPE    ,MSG37          ;DEPRESS SPACE WHEN DONE
3598  015364  004737  027452                  JSR     PC,GETSP        ;GET SPACE
3599
3600  015370  004737  024652                  JSR     PC,GSTAT
3601  015374  032765  010000  000010          BIT     #NED,RKCS2(R5)
3602  015402  001001                          BNE     1$
3603  015404  104122                          ERROR   122             ;NED NOT SET AFTER WRONG PORT SELECTED
3604
3605
3606  015406  104401  037565          1$:     TYPE    ,MSG42          ;SELECT CORRECT PORT
3607  015412  104401  037126                  TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
3608  015416  004737  027452                  JSR     PC,GETSP        ;GET SPACE
3609
3610  015422  004737  024714                  JSR     PC,SUBCLR
3611  015426  104024                          ERROR   24              ;CERR AFTER SCLR
3612
3613  015430  032765  010000  000010          BIT     #NED,RKCS2(R5)
3614  015436  001401                          BEQ     2$
3615  015440  104130                          ERROR   130             ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
3616
3617  015442  104401  037643          2$:     TYPE    ,MSG43          ;DESELECT BOTH PORTS
3618  015446  104401  037126                  TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
3619  015452  004737  027452                  JSR     PC,GETSP
3620  015456  004737  024652                  JSR     PC,GSTAT
3621  015462  032765  010000  000010          BIT     #NED,RKCS2(R5)
3622  015470  001001                          BNE     3$
3623  015472  104133                          ERROR   133             ;NED NOT SET AFTER BOTH PORTS DESELECTED
3624
3625
3626  015474  104401  037673          3$:     TYPE    ,MSG44          ;SELECT CORRECT PORT
3627  015500  104401  037126                  TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
3628  015504  004737  027452                  JSR     PC,GETSP
3629
3630  015510  004737  024714                  JSR     PC,SUBCLR
3631  015514  104024                          ERROR   24              ;CERR AFTER SCLR
3632
3633  015516  032765  010000  000010          BIT     #NED,RKCS2(R5)
3634  015524  001417                          BEQ     TST13           ;;GOTO NEXT TEST
3635  015526  104130                          ERROR   130             ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
3636
3637  015530  012765  100000  000000  4$:     MOV     #CCLR,RKCS1(R5)
3638  015536  004737  024652                  JSR     PC,GSTAT
3639  015542  032765  010000  000010          BIT     #NED,RKCS2(R5)
3640  015550  001405                          BEQ     TST13           ;;GOTO NEXT TEST
3641  015552  104401  037723                  TYPE    ,MSG45          ;CORRECT PORT NOT SELECTED, TRY AGAIN
3642  015556  104401  037126                  TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
3643  015562  000762                          BR      4$
3644
3645                                  ;;*****************************************************************
3646                                  ;*TEST 13       FRONT PANEL RUN/STOP SWITCH TEST
3647                                  ;*
3648                                  ;*      THIS TEST ALLOWS THE HEADS TO LOAD. THE OPERATOR IS
3649                                  ;*      ASKED TO VISUALLY VERIFY THE SEQUENCE OF HEADS LOADING &
3650                                  ;*      UNLOADING BOTH MECHANICALLY & BY THE LIGHTS ON THE FRONT PANEL
3651                                  ;*
```

```
                                ;;*****************************************************************
015564 000004                   TST13:  SCOPE
015566 012737 000001 001170             MOV     #1,$TIMES       ;;DO 1 ITERATION
015574 012706 001100                    MOV     #STACK,SP       ;RESET STACK POINTER.

015600 004737 024714                    JSR     PC,SUBCLR
015604 104024                           ERROR   24              ;CERR AFTER SCLR

015606 104401 035727                    TYPE    ,MSG20          ;RUN/STOP SWITCH TEST
015612 104401 040671                    TYPE    ,MSG52          ;PRESS CONTROL-C TO EXIT OR SPACE TO CONTINUE
015616 004737 027412                    JSR     PC,CCSP         ;INPUT CONTROL-C OR SPACE
015622 000137 016060                    JMP     6$              ;RETURN HERE IF CONTROL-C

015626 104401 037774                    TYPE    ,MSG46          ;DO MANUAL DRIVE LOAD
015632 104401 042775                    TYPE    ,MSG69          ;DEPRESS SPACE WHEN DRIVE 'READY' ON
015636 004737 027452                    JSR     PC,GETSP        ;GET SPACE

015642 004737 024652                    JSR     PC,GSTAT
015646 032737 010000 005426             BIT     #D.SPIN,HMR2
015654 001001                           BNE     1$
015656 104140                           ERROR   140             ;SPINDLE NOT ON AFTER DRIVE MANUALLY LOADED

015660 032737 000200 005426     1$:     BIT     #D.DRDY,HMR2    ;SEE IF DRIVE READY (LOADED)
015666 001001                           BNE     3$              ;BR IF YES
015670 104141                           ERROR   141             ;DRIVE NOT READY AFTER MANUAL LOADING

015672 104401 040177            3$:     TYPE    ,MSG47          ;ATTEMPT TO OPEN DOOR
015676 104401 037126                    TYPE    ,MSG37          ;DEPRESS SPACE WHEN FINISHED
015702 004737 027452                    JSR     PC,GETSP        ;GET SPACE

015706 012765 100000 000000             MOV     #CCLR,RKCS1(R5)
015714 012765 000001 000026             MOV     #1,RKMR1(R5)    ;SELECT WORD 1
015722 004737 024652                    JSR     PC,GSTAT
015726 032737 000200 005426             BIT     #D.DOOR,HMR2
015734 001001                           BNE     4$
015736 104142                           ERROR   142             ;DOOR STATUS BIT NOT SET

015740 104401 040257            4$:     TYPE    ,MSG48          ;UNLOAD HEADS MANUALLY
015744 104401 037126                    TYPE    ,MSG37          ;SPACE BAR WHEN DONE
015750 004737 027452                    JSR     PC,GETSP        ;GET SPACE

015754 012765 100000 000000             MOV     #CCLR,RKCS1(R5)
015762 012765 000001 000026             MOV     #1,RKMR1(R5)    ;SELECT WORD 1
015770 004737 024652                    JSR     PC,GSTAT
015774 032737 000040 005426             BIT     #D.HDHM,HMR2    ;CHECK HEAD HOME
016002 001001                           BNE     5$
016004 104145                           ERROR   145             ;HEADS HOME NOT SET AFTER MANUAL UNLD

016006 104401 037774            5$:     TYPE    ,MSG46          ;DO MANUAL LOAD
016012 104401 042775                    TYPE    ,MSG69          ;PRESS SPACE AFTER READY ON
016016 004737 027452                    JSR     PC,GETSP

016022 004737 024714                    JSR     PC,SUBCLR
016026 104024                           ERROR   24              ;CERR AFTER SCLR

016030 012765 000007 000000             MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD
```

```
3708  016036  013737  001422  005442          MOV     T10,TEMP1
3709  016044  004737  024226                  JSR     PC,FRDY         ;FIND CONTR RDY
3710  016050  104017                          ERROR   17              ;NO RDY AFTER UNLD CMD
3711  016052  004737  024412                  JSR     PC,TSTATN
3712  016056  104020                          ERROR   20              ;NO ATTN AFTER UNLOAD CMD
3713  016060                          6$:
3714
3715                                  ;;*****************************************************************
3716                                  ;*TEST 14       AC LOW DETECTION  PART 1
3717                                  ;*
3718                                  ;*      A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.
3719                                  ;*      BATTERY RETRACT WILL BE TESTED LATER.
3720                                  ;*      THE PROGRAM WAITS FOR AC LOW TO ASSERT IN RKMR3.
3721                                  ;*      FROM THIS POINT, THERE IS APPROX 4 MS AVAILABLE BEFORE DC LOW ASSERTS
3722                                  ;*      & THE INTERFACE SHUTS DOWN.
3723                                  ;*      THE INDICATION OF DC LOW WILL BE NON-EXISTENT DRIVE ASSERTING IN RKCS2.
3724                                  ;*      AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.
3725                                  ;*
3726                                  ;;*****************************************************************
3727  016060  000004                  TST14:  SCOPE
3728  016062  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
3729  016070  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3730
3731  016074  004737  024714                  JSR     PC,SUBCLR
3732  016100  104024                          ERROR   24              ;CERR AFTER SCLR
3733
3734  016102  104401  035760                  TYPE    ,MSG21          ;AC LOW TEST
3735  016106  104401  040671                  TYPE    ,MSG52          ;CONTROL-C TO BYPASS TEST
3736                                                                  ;OR SPACE TO CONTINUE
3737  016112  004737  027412                  JSR     PC,CCSP         ;INPUT CONT-C OR SPACE
3738  016116  000137  016400                  JMP     10$             ;RETURN HERE IF CONT-C
3739  016122  104401  040454                  TYPE    ,MSG49          ;TURN OFF AC(RET HERE IF SPACE)
3740
3741  016126  012737  177777  005442          MOV     #-1,TEMP1       ;SETUP TIMEOUT
3742  016134  004737  024652          1$:     JSR     PC,GSTAT
3743  016140  032737  100000  005400          BIT     #CERR,HCS1
3744  016146  001004                          BNE     9$
3745  016150  005337  005442                  DEC     TEMP1
3746  016154  001367                          BNE     1$
3747  016156  104146                          ERROR   146             ;CERR NOT DETECTED BEFORE TIMEOUT
3748
3749
3750  016160  000137  016212          9$:     JMP     3$              ;***THIS IS A TEMP JUMP***
3751                                  ;ACLO REQUIRES SECTOR PULSE AND SINCE HEADS ARE UNLOADED
3752                                  ;ACLO WILL NOT ASSERT & CERR WILL BE DUE TO NED
3753                                  ;THE PROBLEM IS BEING LOOKED INTO
3754                                  ;THIS TEST WILL EITHER BE ELIMINATED OR AC LO WILL BE MODIFIED TO ASSERT
3755
3756  016164  032737  000100  005430          BIT     #D.ACLO,HMR3
3757  016172  001001                          BNE     2$
3758  016174  104147                          ERROR   147             ;AC LOW NOT SET
3759
3760  016176  012737  000300  005440  2$:     MOV     #<D.ACLO!D.FLT>,SBMR3
3761  016204  004737  027210                  JSR     PC,CKMR3
3762  016210  104203                          ERROR   203             ;MSG BO ERROR AFTER AC SWITCH OFF
3763
```

```
3764  016212  013737  001434  005442  3$:     MOV     T50000,TEMP1
3765  016220  012765  100000  000000  4$:     MOV     #CCLR,RKCS1(R5)
3766  016226  004737  024652                  JSR     PC,GSTAT
3767  016232  032765  010000  000010          BIT     #NED,RKCS2(R5)
3768  016240  001004                          BNE     5$
3769  016242  005337  005442                  DEC     TEMP1
3770  016246  001364                          BNE     4$
3771  016250  104150                          ERROR   150             ;NED NOT SET BEFORE TIMEOUT
3772
3773  016252  104401  040527          5$:     TYPE    ,MSG50          ;SWITCH AC BACK ON
3774  016256  104401  043052                  TYPE    ,MSG70          ;VERIFY HEADS LOAD
3775  016262  104401  042775                  TYPE    ,MSG69          ;PRESS SPACE AFTER READY ON
3776  016266  004737  027452                  JSR     PC,GETSP        ;GET SPACE
3777
3778  016272  004737  024714                  JSR     PC,SUBCLR
3779  016276  104024                          ERROR   24              ;CERR AFTER SCLR
3780
3781  016300  032737  000200  005426          BIT     #D.DRDY,HMR2    ;SEE IF LOADED
3782  016306  001001                          BNE     6$              ;BR IN YES
3783  016310  104141                          ERROR   141             ;DRV NOT RDY AFTER AC UP
3784
3785  016312  032737  000100  005430  6$:     BIT     #D.ACLO,HMR3
3786  016320  001401                          BEQ     7$
3787  016322  104152                          ERROR   152             ;ACLO NOT RESET AFTER POWER UP
3788
3789  016324  032737  000100  005426  7$:     BIT     #D.VV,HMR2
3790  016332  001401                          BEQ     8$
3791  016334  104153                          ERROR   153             ;VV NOT RESET AFTER POWER UP
3792
3793  016336                          8$:
3794  016336  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3795  016344  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
3796  016352  013737  001422  005442          MOV     T10,TEMP1
3797  016360  004737  024226                  JSR     PC,FRDY         ;FIND CONTR RDY
3798  016364  104116                          ERROR   116             ;CONTR NOT RDY
3799
3800  016366  032737  000100  005426          BIT     #D.VV,HMR2
3801  016374  001001                          BNE     64$
3802  016376  104027                          ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
3803  016400                          64$:
3804  016400                          10$:
3805
3806                                  ;;*****************************************************************
3807                                  ;*TEST 15       CHECK NXF LOGIC
3808                                  ;*
3809                                  ;*      THIS TEST VERIFIES NON-EXECUTABLE FUNCTION (NXF) IS DETECTED
3810                                  ;*      AS A RESULT OF DOING A SEEK WITH VOLUME VALID RESET.
3811                                  ;*
3812                                  ;;*****************************************************************
3813  016400  000004          TST15:  SCOPE
3814  016402  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
3815  016410  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR
3816
3817  016414  004737  024714                  JSR     PC,SUBCLR
3818  016420  104024                          ERROR   24              ;CERR AFTER SCLR
3819
```

```
3820  016422  032737  010000  005426          BIT     #D.SPIN,HMR2    ;SEE IF SPINDLE ALREADY ON
3821  016430  001021                          BNE     64$             ;BR IF YES
3822  016432  104401  036365                  TYPE    ,MSG29          ;PLEASE WAIT, HEADS BEING LOADED
3823
3824  016436  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5)  ;START SPINDLE CMD
3825  016444  013737  001422  005442          MOV     T10,TEMP1
3826  016452  004737  024226                  JSR     PC,FRDY         ;FIND CONTR RDY
3827  016456  104143                          ERROR   143             ;CONTR RDY NOT SET AFTER CMD
3828
3829  016460  013737  001430  005444          MOV     T100,TEMP2
3830  016466  004737  024444                  JSR     PC,FATT1        ;FIND ATTN
3831  016472  104144                          ERROR   144             ;NO ATTN AFTER CMD
3832  016474                          64$:
3833
3834  016474  004737  024714                  JSR     PC,SUBCLR
3835  016500  104024                          ERROR   24              ;CERR AFTER SCLR
3836
3837  016502  104401  036021                  TYPE    ,MSG22          ;NXF TEST
3838  016506  104401  040671                  TYPE    ,MSG52          ;PRESS CONT-C TO EXIT OR SPACE TO CONTINUE
3839  016512  004737  027412                  JSR     PC,CCSP         ;INPUT CONTROL-C OR SPACE
3840  016516  000137  017112                  JMP     7$              ;RETURN HERE FOR CONTROL-C
3841
3842  016522  104401  040561                  TYPE    ,MSG51          ;REMOVE UNIT SELECT PLUG TO CLEAR VV
3843  016526  104401  037126                  TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
3844  016532  004737  027452                  JSR     PC,GETSP        ;GET SPACE
3845
3846  016536  004737  024652                  JSR     PC,GSTAT
3847  016542  032737  000100  005426          BIT     #D.VV,HMR2
3848  016550  001403                          BEQ     1$
3849  016552  104177                          ERROR   177             ;VV NOT CLEARED AFTER UNIT SEL PLUG
3850  016554  000137  017112                  JMP     7$              ;EXIT TEST
3851
3852  016560  032737  000100  005426  1$:     BIT     #D.VV,HMR2
3853  016566  001406                          BEQ     2$
3854  016570  104155                          ERROR   155             ;VV SET AFTER HEADS LOADED
3855  016572  000137  017112                  JMP     7$              ;EXIT TEST
3856
3857  016576  004737  024714                  JSR     PC,SUBCLR
3858  016602  104024                          ERROR   24              ;CERR AFTER SCLR
3859
3860  016604  012765  000001  000020  2$:     MOV     #1,RKDC(R5)
3861  016612  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
3862  016620  013737  001422  005442          MOV     T10,TEMP1
3863  016626  004737  024226                  JSR     PC,FRDY         ;FIND RDY
3864  016632  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
3865
3866  016634  013737  001434  005442          MOV     T50000,TEMP1
3867  016642  004737  024540                  JSR     PC,FATT2        ;FIND ATTN
3868  016646  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
3869
3870  016650  032737  000400  005430          BIT     #D.NXF,HMR3
3871  016656  001003                          BNE     3$
3872  016660  104156                          ERROR   156             ;NXF NOT SET AFTER SEEK WITH VV=0
3873  016662  000137  017112                  JMP     7$              ;EXIT TEST
3874
3875  016666  032737  000200  005430  3$:     BIT     #D.FLT,HMR3
```

```
3876  016674  001003                        BNE     4$
3877  016676  104172                        ERROR   172             ;NXF DID NOT SET FAULT
3878  016700  000137  017112                JMP     7$              ;EXIT TEST
3879
3880  016704  012765  100000  000000  4$:   MOV     #CCLR,RKCS1(R5)
3881  016712  004737  025130                JSR     PC,RDCYLA
3882  016716  005737  001374                TST     CYLADD
3883  016722  001401                        BEQ     5$
3884  016724  104157                        ERROR   157             ;HEADS MOVED WITH SEEK & NXF
3885
3886  016726  012765  100000  000000  5$:   MOV     #CCLR,RKCS1(R5)
3887  016734  005065  000026                CLR     RKMR1(R5)
3888  016740  004737  024652                JSR     PC,GSTAT
3889  016744  012737  050240  005436        MOV     #<D.DSC!D.SPIN!D.DRDY!D.DRA>,SBMR2
3890  016752  004737  027126                JSR     PC,CKMR2
3891  016756  104160                        ERROR   160             ;MSG A0 ERROR AFTER SEEK WITH VV=0
3892
3893  016760  012737  000600  005440        MOV     #<D.NXF!D.FLT>,SBMR3
3894  016766  004737  027210                JSR     PC,CKMR3
3895  016772  104161                        ERROR   161             ;MSG B0 ERROR
3896
3897  016774  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3898  017002  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3899  017010  004737  024652                JSR     PC,GSTAT
3900  017014  012737  001720  005436        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3901  017022  004737  027126                JSR     PC,CKMR2
3902  017026  104162                        ERROR   162             ;MSG A1 ERROR AFTER SEEK WITH VV=0
3903
3904  017030  005037  005440                CLR     SBMR3
3905  017034  004737  027210                JSR     PC,CKMR3
3906  017040  104163                        ERROR   163             ;MSG B1 ERROR
3907
3908  017042  004737  024714                JSR     PC,SUBCLR
3909  017046  104024                        ERROR   24              ;CERR AFTER SCLR
3910
3911  017050  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3912  017056  012765  000003  000000        MOV     #PACK,RKCS1(R5) ;PACK CMD
3913  017064  013737  001422  005442        MOV     T10,TEMP1
3914  017072  004737  024226                JSR     PC,FRDY         ;FIND CONTR RDY
3915  017076  104116                        ERROR   116             ;CONTR NOT RDY
3916
3917  017100  032737  000100  005426        BIT     #D.VV,HMR2
3918  017106  001001                        BNE     65$
3919  017110  104027                        ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
3920  017112                          65$:
3921  017112                          7$:
3922
3923                                  ;;*****************************************************************
3924                                  ;*TEST 16    .   READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
3925                                  ;*
3926                                  ;*      THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
3927                                  ;*      THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
3928                                  ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFOR (BSE) WILL BE STORED
3929                                  ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
3930                                  ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
3931                                  ;*      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 WRITING
                                        ;*
                                        ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
                                        ;*
                                        ;;*****************************************************************
017112  000004                  TST16:  SCOPE
017114  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
017122  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

017126  004737  024714                  JSR     PC,SUBCLR
017132  104024                          ERROR   24              ;CERR AFTER SCLR
017134  005037  005444                  CLR     TEMP2           ;SECTOR CTR
017140  005037  005446                  CLR     TEMP3           ;0=22 SEC, 1=20 SEC, 2=DONE
017144  005037  001534                  CLR     BSERR           ;BSE INFO NO GOOD IF SET

017150  012765  003356  000004          MOV     #BSE22,RKBA(R5) ;BSE TABLE FOR 22 SECTOR FORMAT
017156  012765  001000  000006          MOV     #1000,RKDA(R5)  ;HEAD 2, SECTOR 0
017164  012765  000632  000020          MOV     #410.,RKDC(R5)  ;CYL 410

017172  012765  177400  000002  1$:     MOV     #-256.,RKWC(R5) ;LOAD WORD CT
017200  012765  000021  000000          MOV     #RDDATA,RKCS1(R5);READ DATA COMMAND
017206  013737  001434  005442          MOV     T50000,TEMP1    ;SETUP TIMEOUT
017214  004737  024226                  JSR     PC,FRDY         ;FIND RDY
017220  104226                          ERROR   226             ;NO RDY AFTER READ DATA CMD
017222  004737  024652                  JSR     PC,GSTAT        ;GET FRESH DATA
017226  032737  100000  005400          BIT     #CERR,HCS1
017234  001416                          BEQ     3$
017236  104227                          ERROR   227             ;CERR AFTER READ DATA CMD
017240  005237  001534                  INC     BSERR           ;SET BSE ERROR FLAG
017244  032737  010000  005414          BIT     #DTE,HER
017252  001401                          BEQ     2$
017254  104230                          ERROR   230             ;DTE AFTER READ DATA CMD
017256  032737  100000  005402  2$:     BIT     #DLT,HCS2
017264  001404                          BEQ     4$
017266  104231                          ERROR   231             ;DLT AFTER READ DATA CMD
017270  000402                          BR      4$

017272  005037  001534          3$:     CLR     BSERR           ;BSE INFO OK
017276  005737  001534          4$:     TST     BSERR           ;BSE READ OK?
017302  001420                          BEQ     8$              ;BR IF YES
017304  062765  000002  000006          ADD     #2,RKDA(R5)     ;TRY NEXT SECTOR
017312  005237  005444                  INC     TEMP2
017316  023727  005444  000005          CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
017324  001322                          BNE     1$              ;BR IF NO
017326  005737  005446                  TST     TEMP3
017332  001002                          BNE     6$
017334  104233                          ERROR   233             ;CANT READ BSE ON SECTORS 0,2,4,6,8
017336  000502                          BR      TST17           ;;GO TO NEXT TEST
017340  104234                  6$:     ERROR   234             ;CANT READ BSE ON SECTORS 1,3,8,7,9
017342  000500                          BR      TST17           ;;GO TO NEXT TEST

017344  012700  000006          8$:     MOV     #6,R0           ;SETUP FOR WORD 3 OF BSE INFO
017350  016037  003356  005450          MOV     BSE22(R0),TEMP4 ;PULL OUT CARTRIDGE TYPE INFO.
017356  001405                          BEQ     9$              ;BRANCH IF DATA CARTRIDGE
```

```
                                        TYPE    ,MSG56          ;ALIGNMENT CARTRIDGE USED
                                        INC     BSERR           ;SET BSE ERROR FLAG
                                        BR      10$

                                9$:     INC     TEMP3
                                        CMP     TEMP3,#2
                                        BEQ     10$

                                        CLR     TEMP2
                                        MOV     #BSE20,RKBA(R5) ;BSE TABLE FOR 20 SECTOR FORMAT
                                        MOV     #1001,RKDA(R5)  ;HEAD 2, SECTOR 1
                                        BR      1$

                                10$:    TST     $PASS
                                        BNE     12$             ;TYPE CART # ONLY ON 1'ST PASS
                                        TYPE    ,MSG17          ;CART SERIAL #
                                        MOV     #BSE22,-(SP)
                                        JSR     PC,$DB20        ;CONVERT DBL BINARY WORD TO OCTAL
                                        JSR     PC,$SUPRS       ;TYPE SERIAL #
                                        TYPE    ,$CRLF
                                        TYPE    ,$CRLF

                                12$:    JSR     PC,SUBCLR
                                        ERROR   24              ;CERR AFTER SCLR
                                                                ;RETURN TO CYL 0

                                        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
                                        MOV     T50,TEMP1       ;SETUP TIMEOUT
                                        JSR     PC,FRDY         ;FIND RDY
                                        ERROR   131             ;NO RDY AFTER SEEK CMD

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

                                        BIT     #CERR,HCS1
                                        BEQ     64$
                                        ERROR   210             ;CERR AFTER SEEK CMD

                                64$:


                                ;;*****************************************************************
                                ;*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
                                ;*
                                ;;*****************************************************************
                                TST17:  SCOPE
                                        MOV     #1,$TIMES       ;;DO 1 ITERATION
                                        MOV     #STACK,SP       ;RESTORE STACK PTR
                                        TST     BSERR           ;SEE IF ALIGN CART
                                        BEQ     15$             ;BR IF NO
```

```
4044  017566  000137  021650                JMP     12$             ;ELSE EXIT TEST
4045
4046  017572  004737  024714        15$:    JSR     PC,SUBCLR
4047  017576  104024                        ERROR   24              ;CERR AFTER SCLR
4048
4049  017600  104401  036167                TYPE    ,MSG24          ;WRITE LOCK TEST
4050  017604  104401  040671                TYPE    ,MSG52          ;PRESS CONT-C TO EXIT OR SPACE TO CONTINUE
4051  017610  004737  027412                JSR     PC,CCSP         ;INPUT CONT-C OR SPACE
4052  017614  000137  021650                JMP     12$             ;RETURN HERE IF CONT-C
4053
4054  017620  004737  024652                JSR     PC,GSTAT
4055  017624  032737  004000  005426        BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK IS ON
4056  017632  001416                        BEQ     2$              ;BR IF NO
4057  017634  104401  040772        1$:     TYPE    ,MSG53          ;DISABLE WRITE LOCK
4058  017640  104401  037126                TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
4059  017644  004737  027452                JSR     PC,GETSP        ;GET SPACE
4060
4061  017650  004737  024652                JSR     PC,GSTAT
4062  017654  032737  004000  005426        BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK IS OFF
4063  017662  001402                        BEQ     2$
4064  017664  104110                        ERROR   110             ;WRITE LOCK NOT DISABLED
4065  017666  000762                        BR      1$
4066
4067  017670  005037  001362        2$:     CLR     TOCYL           ;SETUP
4068  017674  005037  001376                CLR     CALADD          ;FOR
4069  017700  005037  001504                CLR     HEAD            ;FILL HEADER TABLE
4070  017704  005037  001510                CLR     FORMAT          ;ROUTINE
4071
4072  017710  004737  025702        16$:    JSR     PC,FHDTAB       ;BUILD STD 22 SECTOR HEADER TABLE
4073
4074  017714  012765  001542  000004        MOV     #HDTAB,RKBA(R5)
4075  017722  012765  177676  000002        MOV     #-66.,RKWC(R5)
4076  017730  000337  001504                SWAB    HEAD
4077  017734  013765  001504  000006        MOV     HEAD,RKDA(R5)   ;HEAD ADDR
4078  017742  000337  001504                SWAB    HEAD
4079
4080
4081  017746  012765  000027  000000        MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
4082  017754  013737  001434  005442        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4083  017762  004737  024226                JSR     PC,FRDY         ;FIND RDY
4084  017766  104200                        ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4085  017770  004737  024652                JSR     PC,GSTAT        ;GET FRESH STATUS
4086  017774  032737  100000  005400        BIT     #CERR,HCS1
4087  020002  001401                        BEQ     64$
4088  020004  104201                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
4089  020006                        64$:
4090
4091
4092  020006  005237  001504                INC     HEAD
4093  020012  023727  001504  000003        CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
4094  020020  001333                        BNE     16$             ;BR IF NO
4095
4096  020022  004737  024714                JSR     PC,SUBCLR
4097  020026  104024                        ERROR   24              ;CERR AFTER SCLR
4098
4099  020030  012765  001516  000004        MOV     #DATA0,RKBA(R5) ;WRITE ALL 0'S
```

```
4100  020036  052765  000020  000010          BIS     #BAI,RKCS2(R5)
4101  020044  012765  137000  000002          MOV     #-66.*256.,RKWC(R5)     ;CYL 0, TRK 0,1,2
4102  020052  005065  000006                  CLR     RKDA(R5)                ;TRK/SEC 0
4103
4104  020056  012765  000023  000000          MOV     #<WRDATA>,RKCS1(R5)     ;WRITE DATA CMD
4105  020064  013737  001434  005442          MOV     T50000,TEMP1            ;SETUP TIMEOUT
4106  020072  004737  024226                  JSR     PC,FRDY                 ;FIND RDY
4107  020076  104011                          ERROR   11                      ;NO RDY AFTER WRITE DATA CMD
4108  020100  004737  024652                  JSR     PC,GSTAT                ;GET FRESH STATUS
4109  020104  032737  100000  005400          BIT     #CERR,HCS1
4110  020112  001401                          BEQ     65$
4111  020114  104012                          ERROR   12                      ;CERR AFTER WRITE DATA CMD
4112
4113  020116                          65$:
4114
4115  020116  004737  024714                  JSR     PC,SUBCLR
4116  020122  104024                          ERROR   24                      ;CERR AFTER SCLR
4117
4118  020124  012765  001516  000004          MOV     #DATA0,RKBA(R5)
4119  020132  052765  000020  000010          BIS     #BAI,RKCS2(R5)
4120  020140  012765  137000  000002          MOV     #-66.*256.,RKWC(R5)
4121  020146  005065  000006                  CLR     RKDA(R5)                ;TRK/SEC 0
4122  020152  013737  001516  001500          MOV     DATA0,WD2               ;EXPECTED WORD FOR TRUERR TYPEOUT
4123
4124  020160  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4125  020166  013737  001434  005442          MOV     T50000,TEMP1            ;SETUP TIMEOUT
4126  020174  004737  024226                  JSR     PC,FRDY                 ;FIND RDY
4127  020200  104015                          ERROR   15                      ;NO RDY AFTER WRITE CHECK CMD
4128  020202  004737  024652                  JSR     PC,GSTAT                ;GET FRESH STATUS
4129  020206  032737  100000  005400          BIT     #CERR,HCS1
4130  020214  001414                          BEQ     67$
4131
4132  020216  032737  040000  005402          BIT     #WCE,HCS2
4133  020224  001407                          BEQ     66$
4134  020226  016537  000024  001476          MOV     RKDB(R5),WD1    ;GET ACTUAL MISCOMPARED WORD FOR TYPEOUT
4135  020234  004737  026212                  JSR     PC,TRUERR       ;CHECK AGAINST BSE INFO
4136  020240  104016                          ERROR   16              ;WRITE CHECK ERROR
4137  020242  000401                          BR      67$
4138
4139  020244  104022                  66$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4140
4141  020246                          67$:
4142
4143
4144  020246  004737  024714          3$:     JSR     PC,SUBCLR
4145  020252  104024                          ERROR   24              ;CERR AFTER SCLR
4146
4147  020254  104401  041063                  TYPE    ,MSG54          ;ENABLE WRITE LOCK
4148  020260  104401  037126                  TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4149  020264  004737  027452                  JSR     PC,GETSP        ;GET SPACE
4150
4151  020270  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4152  020276  004737  024652                  JSR     PC,GSTAT
4153  020302  032737  004000  005426          BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK IS ON
4154  020310  001002                          BNE     4$
4155  020312  104115                          ERROR   115             ;WRITE LOCK NOT ENABLED
```

```
4156  020314  000754                          BR      3$
4157
4158  020316  012765  001522  000004  4$:     MOV     #DATA1,RKBA(R5) ;ATTEMPT TO WRITE ALL 1'S WITH WRITE LOCK SET
4159  020324  052765  000020  000010          BIS     #BAI,RKCS2(R5)
4160  020332  012765  165000  000002          MOV     #-22.*256.,RKWC(R5)  ; CYLINDER 0, HEAD 0
4161  020340  005065  000006                  CLR     RKDA(R5)
4162  020344  012765  000023  000000          MOV     #WRDATA,RKCS1(R5)
4163  020352  013737  001434  005442          MOV     T50000,TEMP1        ;SETUP TIMEOUT
4164  020360  004737  024226                  JSR     PC,FRDY             ;FIND RDY
4165  020364  104116                          ERROR   116                 ;CONTR NOT RDY
4166
4167  020366  004737  024652                  JSR     PC,GSTAT            ;GET FRESH STATUS
4168  020372  032737  100000  005400          BIT     #CERR,HCS1
4169  020400  001001                          BNE     17$
4170  020402  104207                          ERROR   207                 ;CERR NOT SET AFTER WRITE WITH WRL SET
4171  020404  032737  004000  005430  17$:    BIT     #D.WLE,HMR3         ;CHECK WRITE LOCK ERROR SET
4172  020412  001001                          BNE     5$                  ;BR IF SET
4173  020414  104121                          ERROR   121                 ;WRITE LOCK ERROR NOT SET
4174                                                                      ;AFTER WRITE DATA WITH WRITE LOCK SET
4175  020416  012737  054340  005436  5$:     MOV     #<D.DSC!D.SPIN!D.WRL!D.DRDY!D.VV!D.DRA>,SBMR2
4176  020424  004737  027126                  JSR     PC,CKMR2
4177  020430  104123                          ERROR   123                 ;MSG A0 ERROR AFTER WRITE WITH WRITE LOCK SET
4178  020432  012737  004200  005440          MOV     #<D.WLE!D.FLT>,SBMR3
4179  020440  004737  027210                  JSR     PC,CKMR3
4180  020444  104125                          ERROR   125                 ;MSG B0 ERROR
4181
4182  020446  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4183  020454  012765  000001  000026          MOV     #1,RKMR1(R5)        ;SELECT WORD 1
4184  020462  004737  024652                  JSR     PC,GSTAT
4185  020466  012737  001720  005436          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4186  020474  004737  027126                  JSR     PC,CKMR2
4187  020500  104126                          ERROR   126                 ;MSG A1 ERROR
4188  020502  005037  005440                  CLR     SBMR3
4189  020506  004737  027210                  JSR     PC,CKMR3
4190  020512  104127                          ERROR   127                 ;MSG B1 ERROR
4191
4192  020514  004737  024714                  JSR     PC,SUBCLR
4193  020520  104024                          ERROR   24                  ;CERR AFTER SCLR
4194
4195  020522  012765  001516  000004          MOV     #DATA0,RKBA(R5) ;CHECK THAT NONE OF ORIG DATA
4196  020530  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;HAS CHANGED
4197  020536  012765  165000  000002          MOV     #-22.*256.,RKWC(R5)  ;AS RESULT OF WRITE WITH WRITE LOCK SET
4198  020544  013737  001516  001500          MOV     DATA0,WD2           ;EXPECTED WORD FOR TRUERR TYPEOUT
4199
4200  020552  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4201  020560  013737  001434  005442          MOV     T50000, TEMP1       ;SETUP TIMEOUT
4202  020566  004737  024226                  JSR     PC,FRDY             ;FIND RDY
4203  020572  104015                          ERROR   15                  ;NO RDY AFTER WRITE CHECK CMD
4204  020574  004737  024652                  JSR     PC,GSTAT            ;GET FRESH STATUS
4205  020600  032737  100000  005400          BIT     #CERR,HCS1
4206  020606  001414                          BEQ     69$
4207
4208  020610  032737  040000  005402          BIT     #WCE,HCS2
4209  020616  001407                          BEQ     68$
4210  020620  016537  000024  001476          MOV     RKDB(R5),WD1        ;GET ACTUAL MISCOMPARED WORD FOR TYPEOUT
4211  020626  004737  026212                  JSR     PC,TRUERR           ;CHECK AGAINST BSE INFO
```

```
4212  020632  104016                        ERROR   16              ;WRITE CHECK ERROR
4213  020634  000401                        BR      69$
4214
4215  020636  104022                69$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4216
4217  020640                        69$:
4218
4219  020640  104401  042336                TYPE    ,MSG64          ;FOLLOWING TEST TEMPORARY BYPASSED
4220
4221                                ;THE CHANGE OF WRL LOGIC RESULTS IN A DSC AND ATTN TO THE CONTROLLER
4222                                ;WHICH RECOGNIZES IT AS AN AN ERROR
4223                                ;THEREFORE, IN CONTINUOUS WRITING RKDA WILL ALWAYS = 1 WHEN THE
4224                                ;WRL SWITCH IS ENABLED.
4225                                ;THE FOLLOWING CODE WILL BE VALID IF THE PROPOSED ECO IS APPROVED TO
4226                                ;ELIMINATE THE CHANGE OF WRL LOGIC
4227
4228  020644  000137  021650                JMP     12$             ;EXIT TEST
4229
4230  020650  004737  024714        6$:     JSR     PC,SUBCLR
4231  020654  104024                        ERROR   24              ;CERR AFTER SCLR
4232
4233  020656  104401  040772                TYPE    ,MSG53          ;DISABLE WRITE LOCK
4234  020662  104401  037126                TYPE    ,MSG37          ;SPACE BAR WHEN DONE
4235  020666  004737  027452                JSR     PC,GETSP        ;GET SPACE
4236  020672  004737  024652                JSR     PC,GSTAT
4237  020676  032737  004000  005426        BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK OFF
4238  020704  001361                        BNE     6$
4239                                                                ;TEST FOR WRITE LOCK AT SECTOR BOUNDRIES
4240                                                                ;FOR CONTINUOUS WRITING ON CYL 0, TRACK 0,1,2
4241  020706  104401  041063                TYPE    ,MSG54          ;SET WRITE LOCK
4242  020712  013737  001432  001160        MOV     T5000,$TMP0
4243  020720  005037  001162                CLR     $TMP1           ;BIT0=0:WRITE 0'S; BIT0=1;WRITE 1'S
4244
4245  020724  004737  024714        7$:     JSR     PC,SUBCLR
4246  020730  104024                        ERRCR   24              ;CERR AFTER SCLR
4247
4248  020732  032737  000001  001162        BIT     #BIT0,$TMP1
4249  020740  001004                        BNE     8$              ;BR IF WRITING 1'S
4250  020742  012765  001516  000004        MOV     #DATA0,RKBA(R5) ;SETUP ALL 0'S
4251  020750  000403                        BR      9$
4252
4253  020752  012765  001522  000004  8$:   MOV     #DATA1,RKBA(R5) ;SETUP ALL 1'S
4254  020760  052765  000020  000010  9$:   BIS     #BAI,RKCS2(R5)
4255  020766  012765  140400  000002        MOV     #-63.*256.,RKWC(R5)     ;WRITE TRACK 0,1,2  63 SECTORS
4256  020774  005065  000006                CLR     RKDA(R5)        ;BEGIN AT TRACK AND SEC 0
4257  021000  012765  000023  000000        MOV     #WRDATA,RKCS1(R5)
4258  021006  013737  001434  005442        MOV     T50000,TEMP1
4259  021014  004737  024226                JSR     PC,FRDY         ;FIND CONTR RDY
4260  021020  104011                        ERROR   11              ;CONTR NOT RDY
4261
4262  021022  032737  100000  005400        BIT     #CERR,HCS1
4263  021030  001017                        BNE     13$
4264  021032  005337  001160                DEC     $TMP0
4265  021036  001011                        BNE     10$
4266  021040  104204                        ERROR   204             ;CERR NOT SET BY TIMEOUT
4267  021042  104401  040772                TYPE    ,MSG53          ;DISABLE WRITE LOCK
```

```
4268  021046  104401  037126                  TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4269  021052  004737  027452                  JSR     PC,GETSP        ;INPUT SPACE
4270  021056  000137  021650                  JMP     12$             ;EXIT TEST
4271
4272  021062  005237  001162          10$:    INC     $TMP1
4273  021066  000716                  13$:    BR      7$              ;GO WRITE OPPOSITE DATA
4274
4275  021070  032737  004000  005430  13$:    BIT     #D.WLE,HMR3
4276  021076  001001                          BNE     11$
4277  021100  104205                          ERROR   205             ;NO WRL BY TIMEOUT
4278
4279  021102  104401  040772          11$:    TYPE    ,MSG53          ;DISABLE WRITE LOCK
4280  021106  104401  037126                  TYPE    ,MSG37          ;TYPE SPACE WHEN DONE
4281  021112  004737  027452                  JSR     PC,GETSP
4282
4283  021116  013737  005410  001164          MOV     HDA,$TMP2       ;STORE RKDA OF ERROR
4284  021124  013737  005410  001166          MOV     HDA,$TMP3       ;STORE FOR ERROR TYPEOUT
4285
4286  021132  023727  001164  000001          CMP     $TMP2,#1        ;SEE IF TRK 0, SEC 1
4287  021140  001643                          BEQ     6$              ;REPEAT,NO NEW DATA XFER TOOK PLACE
4288  021142  023727  001164  001024          CMP     $TMP2,#1024     ;SEE IF TRK 2, SEC 20
4289  021150  001637                          BEQ     6$              ;REPEAT,NO OLD DATA TO CHECK AGAINST
4290  021152  023727  001164  000400          CMP     $TMP2,#400      ;SEE IF WRL AT TRK 1, SEC 0
4291  021160  001004                          BNE     18$             ;BR IF NO
4292  021162  012765  000025  000006          MOV     #21.,RKDA(R5)   ;ELSE SECTOR AT WRL IS TRK 0, SEC 21
4293  021170  000415                          BR      20$
4294
4295  021172  023727  001164  001000  18$:    CMP     $TMP2,#1000     ;SEE IF WRL AT TRK 2, SEC 0
4296  021200  001004                          BNE     19$             ;BR IF NO
4297  021202  012765  000425  000006          MOV     #425,RKDA(R5)   ;ELSE SECTOR AT WRL IS TRK 1, SEC 21
4298  021210  000405                          BR      20$
4299
4300  021212  005337  001164          19$:    DEC     $TMP2           ;GET SECTOR AT WRL
4301  021216  013765  001164  000006          MOV     $TMP2,RKDA(R5)
4302
4303  021224  016537  000006  001166  20$:    MOV     RKDA(R5),$TMP3  ;FOR ERROR PRINTOUT
4304
4305
4306
4307  021232  004737  024714                  JSR     PC,SUBCLR
4308  021236  104024                          ERROR   24              ;CERR AFTER SCLR
4309
4310  021240  005737  001164                  TST     $TMP2           ;SEE IF TRK/SECTOR 0
4311  021244  001414                          BEQ     77$             ;REPEAT,NO NEW DATA XFER TOOK PLACE
4312  021246  023727  001164  001023          CMP     $TMP2,#1023     ;SEE IF TRK 2,SECTOR 19
4313  021254  001410                          BEQ     77$             ;REPEAT,NO OLD DATA TO CHECK AGAINST
4314  021256  032737  000001  001162          BIT     #BIT0,$TMP1
4315  021264  001006                          BNE     70$             ;BR IF WRITING 1'S WHEN WLE OCCURRED
4316  021266  012765  001522  000004          MOV     #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4317  021274  000405                          BR      71$
4318
4319  021276  000137  020650          77$:    JMP     6$
4320
4321  021302  012765  001516  000004  70$:    MOV     #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4322  021310  052765  000020  000010  71$:    BIS     #BAI,RKCS2(R5)
4323  021316  012765  177400  000002          MOV     #-256.,RKWC(R5)
```

```
4324  021324  013765  001166  000006          MOV     $TMP3,RKDA(R5)  ;REFRESH RKDA
4325  021332  017537  000004  001500          MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
4326
4327  021340  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4328  021346  013737  001434  005442          MOV     T50000, TEMP1   ;SETUP TIMEOUT
4329  021354  004737  024226                  JSR     PC,FRDY         ;FIND RDY
4330  021360  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4331  021362  004737  024652                  JSR     PC,GSTAT        ;GET FRESH STATUS
4332  021366  032737  100000  005400          BIT     #CERR,HCS1
4333  021374  001414                          BEQ     79$
4334
4335  021376  032737  040000  005402          BIT     #WCE,HCS2
4336  021404  001407                          BEQ     78$
4337  021406  016537  000024  001476          MOV     RKDB(R5),WD1    ;GET ACTUAL MISCOMPARED WORD FOR TYPEOUT
4338  021414  004737  026212                  JSR     PC,TRUERR       ;CHECK AGAINST BSE INFO
4339  021420  104134                          ERROR   134             ;WRITE CHECK ERROR
4340  021422  000401                          BR      79$
4341
4342  021424  104022                  78$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4343
4344  021426                          79$:
4345
4346  021426  000240                          NOP
4347  021430  000240                          NOP
4348
4349  021432  023727  001164  000400          CMP     $TMP2,#400      ;SEE IF WRL AT TRK 1, SECTOR 0
4350  021440  001004                          BNE     72$             ;BR IF NO
4351  021442  012765  000025  000006          MOV     #21.,RKDA(R5)   ;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4352  021450  000415                          BR      74$
4353  021452  023727  001164  001000  72$:    CMP     $TMP2,#1000     ;SEE IF WRL AT TRK 2,SECTOR 0
4354  021460  001004                          BNE     73$             ;BR IF NO
4355  021462  012765  000425  000006          MOV     #425,RKDA(R5)   ;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4356  021470  000405                          BR      74$
4357
4358  021472  005337  001164          73$:    DEC     $TMP2           ;GET SECTOR BEFORE WRL
4359  021476  013765  001164  000006          MOV     $TMP2,RKDA(R5)
4360  021504  016537  000006  001166  74$:    MOV     RKDA(R5),$TMP3  ;FOR ERROR PRINTOUT
4361  021512  032737  000001  001162          BIT     #BIT0,$TMP1
4362  021520  001004                          BNE     75$             ;BR IF WRITING 1'S WHEN WLE OCCURRED
4363  021522  012765  001516  000004          MOV     #DATA0,RKBA(R5) ;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4364  021530  000403                          BR      76$
4365
4366  021532  012765  001522  000004  75$:    MOV     #DATA1,RKBA(R5) ;WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4367  021540  052765  000020  000010  76$:    BIS     #BAI,RKCS2(R5)
4368  021546  012765  177400  000002          MOV     #-256.,RKWC(R5)
4369  021554  017537  000004  001500          MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
4370
4371  021562  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4372  021570  013737  001434  005442          MOV     T50000, TEMP1   ;SETUP TIMEOUT
4373  021576  004737  024226                  JSR     PC,FRDY         ;FIND RDY
4374  021602  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4375  021604  004737  024652                  JSR     PC,GSTAT        ;GET FRESH STATUS
4376  021610  032737  100000  005400          BIT     #CERR,HCS1
4377  021616  001414                          BEQ     81$
4378
4379  021620  032737  040000  005402          BIT     #WCE,HCS2
```

```
4380  021626  001407                          BEQ     80$
4381  021630  016537  000024  001476          MOV     RKDB(R5),WD1    ;GET ACTUAL MISCOMPARED WORD FOR TYPEOUT
4382  021636  004737  026212                  JSR     PC,TRUERR       ;CHECK AGAINST BSE INFO
4383  021642  104135                          ERROR   13$             ;WRITE CHECK ERROR
4384  021644  000401                          BR      81$
4385
4386  021646  104022                  80$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4387
4388  021650                          81$:
4389
4390
4391  021650                          12$:
4392
4393                                  ;;*****************************************************************
4394                                  ;*TEST 20       AC LOW DETECTION PART 2
4395                                  ;*
4396                                  ;*      THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
4397                                  ;*      AT SECTOR BOUNDRIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
4398                                  ;*      THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
4399                                  ;*      WHEN THE INTERFACE SHUTS DOWN.
4400                                  ;*
4401                                  ;;*****************************************************************
4402  021650  000004                  TST20:  SCOPE
4403  021652  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
4404  021660  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4405
4406  021664  004737  024714                  JSR     PC,SUBCLR
4407  021670  104024                          ERROR   24              ;CERR AFTER SCLR
4408
4409  021672  104401  036232                  TYPE    ,MSG26          ;AC LOW-PART 2
4410  021676  104401  040671                  TYPE    ,MSG52          ;CONTROL-C TO BYPASS TEST
4411                                                                  ;OR SPACE TO CONINUE
4412  021702  004737  027412                  JSR     PC,CCSP         ;INPUT CONT-C OR SPACE
4413  021706  000137  023226                  JMP     12$             ;RETURN HERE FOR CONT-C
4414                                                                  ;RETURN HERE FOR SPACE
4415  021712  005737  001534                  TST     BSERR           ;TEST FOR ALIGN CARTRIDGE
4416  021716  001401                          BEQ     1$              ;BR IF NOT ALIGN CART.
4417  021720  000536                          BR      2$
4418
4419  021722  004737  024652          1$:     JSR     PC,GSTAT
4420  021726  032737  004000  005426          BIT     #D.WRL,HMR2     ;SEE IF WRITE LOCK
4421  021734  001417                          BEQ     11$             ;BR IF NO
4422
4423  021736  104401  040772                  TYPE    ,MSG53          ;DISABLE WRITE LOCK
4424  021742  104401  037126                  TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4425  021746  004737  027452                  JSR     PC,GETSP        ;GET SPACE
4426
4427  021752  004737  024652                  JSR     PC,GSTAT
4428  021756  032737  004000  005426          BIT     #D.WRL,HMR2     ;SEE IF STILL WRITE LOCK
4429  021764  001403                          BEQ     11$             ;BR IF NO
4430  021766  104110                          ERROR   110             ;WRITE LOCK NOT DISABLED
4431  021770  000137  023226                  JMP     12$             ;EXIT TEST
4432
4433  021774  005037  001362          11$:    CLR     TOCYL           ;SETUP
4434  022000  005037  001376                  CLR     CALADD          ;FOR
4435  022004  005037  001504                  CLR     HEAD            ;FILL HEADER TABLE
```

```
4436  022010  005037  001510                CLR     FORMAT          ;ROUTINE
4437
4438  022014  004737  025702        13$:    JSR     PC,FHDTAB       ;BUILD STD 22 SECTOR HEADER TABLE
4439
4440  022020  012765  001542  000004        MOV     #HDTAB,RKBA(R5)
4441  022026  012765  177676  000002        MOV     #-66.,RKWC(R5)
4442  022034  000337  001504                SWAB    HEAD
4443  022040  013765  001504  000006        MOV     HEAD,RKDA(R5)   ;HEAD ADDR
4444  022046  000337  001504                SWAB    HEAD
4445
4446  022052  012765  000027  000000        MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
4447  022060  013737  001434  005442        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4448  022066  004737  024226                JSR     PC,FRDY         ;FIND RDY
4449  022072  104200                        ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4450  022074  004737  024652                JSR     PC,GSTAT        ;GET FRESH STATUS
4451  022100  032737  100000  005400        BIT     #CERR,HCS1
4452  022106  001401                        BEQ     64$
4453  022110  104201                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
4454  022112                        64$:
4455
4456  022112  005237  001504                INC     HEAD
4457  022116  023727  001504  000003        CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
4458  022124  001333                        BNE     13$             ;BR IF NO
4459
4460  022126  004737  024714                JSR     PC,SUBCLR
4461  022132  104024                        ERROR   24              ;CERR AFTER SCLR
4462
4463  022134  012765  001522  000004        MOV     #DATA1,RKBA(R5) ;RETURN HERE FOR SPACE
4464  022142  052765  000020  000010        BIS     #BAI,RKCS2(R5)  ;WRITE INITIAL BACKGROUND OF 1'S
4465  022150  012765  137000  000002        MOV     #-66.*256.,RKWC(R5)     ;TRACK 0,1,2  ALL SECTORS
4466
4467  022156  012765  000023  000000        MOV     #<WRDATA>, RKCS1(R5)    ;WRITE DATA CMD
4468  022164  013737  001434  005442        MOV     T50000, TEMP1   ;SETUP TIMEOUT
4469  022172  004737  024226                JSR     PC,FRDY         ;FIND RDY
4470  022176  104011                        ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4471  022200  004737  024652                JSR     PC,GSTAT        ;GET FRESH STATUS
4472  022204  032737  100000  005400        BIT     #CERR,HCS1
4473  022212  001401                        BEQ     65$
4474  022214  104012                        ERROR   12              ;CERR AFTER WRITE DATA CMD
4475
4476  022216                        65$:
4477
4478  022216  004737  024714        2$:     JSR     PC,SUBCLR
4479  022222  104024                        ERROR   24              ;CERR AFTER SCLR
4480
4481  022224  104401  040454                TYPE    ,MSG49          ;TURN OFF AC
4482  022230  104401  041436                TYPE    ,MSG57          ;VERIFY BATTERY RETRACT FUNCTIONAL
4483
4484  022234  005737  001534                TST     BSERR           ;SEE IF ALIGN CART USED
4485  022240  001405                        BEQ     15$             ;BR IF NO
4486  022242  104401  037126                TYPE    ,MSG37          ;PRESS SPACE WHEN DONE
4487  022246  004737  027452                JSR     PC,GETSP        ;GET SPACE
4488  022252  000504                        BR      8$              ;SKIP ALL WRITING
4489
4490  022254  013737  001432  001160  15$:  MOV     T5000,$TMP0
4491  022262  005037  001162                CLR     $TMP1           ;BIT0=0;WRITE 0'S; BIT0=1:WRITE 1'S
```

```
4492
4493  022266  004737  024714          4$:     JSR     PC,SUBCLR
4494  022272  104024                          ERROR   24                      ;CERR AFTER SCLR
4495
4496  022274  032737  000001  001162          BIT     #BIT0,$TMP1
4497  022302  001004                          BNE     5$                      ;BR IF WRITING 1'S
4498  022304  012765  001516  000004          MOV     #DATA0,RKBA(R5) ;SETUP ALL 0'S
4499  022312  000403                          BR      6$
4500
4501  022314  012765  001522  000004  5$:     MOV     #DATA1,RKBA(R5) ;SETUP ALL 1'S
4502  022322  052765  000020  000010  6$:     BIS     #BAI,RKCS2(R5)
4503  022330  012765  140400  000002          MOV     #-63.*256.,RKWC(R5)     ;TRACK 0,1,2  63 SECTORS
4504  022336  005065  000006                  CLR     RKDA(R5)                ;BEGIN AT TRK AND SECTOR 0
4505  022342  012765  000023  000000          MOV     #WRDATA,RKCS1(R5)
4506  022350  013737  001434  005442          MOV     T50000,TEMP1
4507  022356  004737  024226                  JSR     PC,FRDY                 ;FIND CONTR RDY
4508  022362  104011                          ERROR   11                      ;CONTR NOT RDY
4509
4510  022364  004737  024652                  JSR     PC,GSTAT
4511  022370  032737  100000  005400          BIT     #CERR,HCS1
4512  022376  001017                          BNE     3$
4513  022400  005337  001160                  DEC     $TMP0
4514  022404  001011                          BNE     7$
4515  022406  104146                          ERROR   146                     ;CERR NOT SET BY TIMEOUT
4516  022410  104401  040527                  TYPE    ,MSG50                  ;TURN AC BACK ON
4517  022414  104401  042775                  TYPE    ,MSG69                  ;DEPRESS SPACE AFTER 'READY' LIGHT ON
4518  022420  004737  027452                  JSR     PC,GETSP                ;GET SPACE
4519  022424  000137  023226                  JMP     12$                     ;EXIT TEST
4520
4521  022430  005237  001162          7$:     INC     $TMP1
4522  022434  000714                          BR      4$                      ;GO WRITE OPPOSITE DATA
4523
4524  022436  032737  000100  005430  3$:     BIT     #D.ACLO,HMR3
4525  022444  001001                          BNE     10$
4526  022446  104147                          ERROR   147                     ;AC LOW NOT SET
4527
4528  022450  012737  010300  005440  10$:    MOV     #<D.SPLS!D.ACLO!D.FLT>,SBMR3
4529  022456  004737  027210                  JSR     PC,CKMR3
4530  022462  104203                          ERROR   203                     ;MSG 80 ERROR AFTER AC SW OFF
4531
4532  022464  013737  005410  001164  8$:     MOV     HDA,$TMP2               ;SAVE RKDA
4533  022472  013737  005410  001166          MOV     HDA,$TMP3               ;SAVE FOR TYPEOUT
4534
4535  022500  104401  040527                  TYPE    ,MSG50                  ;TURN AC BACK ON
4536  022504  104401  042775                  TYPE    ,MSG69                  ;DEPRESS SPACE AFTER 'READY' LIGHT ON
4537  022510  004737  027452                  JSR     PC,GETSP                ;GET SPACE
4538
4539  022514  004737  024714                  JSR     PC,SUBCLR
4540  022520  104024                          ERROR   24                      ;CERR AFTER SCLR
4541
4542  022522  032737  000100  005430          BIT     #D.ACLO,HMR3
4543  022530  001401                          BEQ     9$
4544  022532  104152                          ERROR   152                     ;ACLO NOT RESET AFTER POWER UP
4545
4546  022534                          9$:
4547  022534  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
4548  022542  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;PACK CMD
4549  022550  013737  001422  005442          MOV     T10,TEMP1
4550  022556  004737  024226                  JSR     PC,FRDY         ;FIND CONTR RDY
4551  022562  104116                          ERROR   116             ;CONTR NOT RDY
4552
4553  022564  032737  000100  005426          BIT     #D.VV,HMR2
4554  022572  001001                          BNE     66$
4555  022574  104027                          ERROR   27              ;VOLUME VALID NOT SET AFTER PACK CMD
4556  022576                          66$:
4557  022576  005737  001534                  TST     BSERR           ;SEE IF ALIGN CART USED
4558  022602  001402                          BEQ     14$             ;BR IF NO
4559  022604  000137  023226                  JMP     12$             ;ELSE EXIT TEST
4560
4561  022610                          14$:
4562
4563  022610  004737  024714                  JSR     PC,SUBCLR
4564  022614  104024                          ERROR   24              ;CERR AFTER SCLR
4565
4566  022616  005737  001164                  TST     $TMP2           ;SEE IF TRK/SECTOR 0
4567  022622  001414                          BEQ     74$             ;REPEAT,NO NEW DATA XFER TOOK PLACE
4568  022624  023727  001164  001023          CMP     $TMP2,#1023     ;SEE IF TRK 2,SECTOR 19
4569  022632  001410                          BEQ     74$             ;REPEAT,NO OLD DATA TO CHECK AGAINST
4570  022634  032737  000001  001162          BIT     #BIT0,$TMP1
4571  022642  001006                          BNE     67$             ;BR IF WRITING 1'S WHEN WLE OCCURRED
4572  022644  012765  001522  000004          MOV     #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4573  022652  000405                          BR      68$
4574
4575  022654  000137  022216          74$:    JMP     2$
4576
4577  022660  012765  001516  000004  67$:    MOV     #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4578  022666  052765  000020  000010  68$:    BIS     #BAI,RKCS2(R5)
4579  022674  012765  177400  000002          MOV     #-256.,RKWC(R5)
4580  022702  013765  001166  000006          MOV     $TMP3,RKDA(R5)  ;REFRESH RKDA
4581  022710  017537  000004  001500          MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
4582
4583  022716  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4584  022724  013737  001434  005442          MOV     T50000, TEMP1   ;SETUP TIMEOUT
4585  022732  004737  024226                  JSR     PC,FRDY         ;FIND RDY
4586  022736  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4587  022740  004737  024652                  JSR     PC,GSTAT        ;GET FRESH STATUS
4588  022744  032737  100000  005400          BIT     #CERR,HCS1
4589  022752  001414                          BEQ     76$
4590
4591  022754  032737  040000  005402          BIT     #WCE,HCS2
4592  022762  001407                          BEQ     75$
4593  022764  016537  000024  001476          MOV     RKDB(R5),WD1    ;GET ACTUAL MISCOMPARED WORD FOR TYPEOUT
4594  022772  004737  026212                  JSR     PC,TRUERR       ;CHECK AGAINST BSE INFO
4595  022776  104136                          ERROR   136             ;WRITE CHECK ERROR
4596  023000  000401                          BR      76$
4597
4598  023002  104022                  75$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4599
4600  023004                          76$:
4601
4602  023004  000240                          NOP
4603  023006  000240                          NOP
```

```
4604
4605  023010  023727  001164  000400          CMP     $TMP2,#400      ;SEE IF WRL AT TRK 1, SECTOR 0
4606  023016  001004                          BNE     69$             ;BR IF NO
4607  023020  012765  000025  000006          MOV     #21.,RKDA(R5)   ;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4608  023026  000415                          BR      71$
4609  023030  023727  001164  001000  69$:    CMP     $TMP2,#1000     ;SEE IF WRL AT TRK 2,SECTOR 0
4610  023036  001004                          BNE     70$             ;BR IF NO
4611  023040  012765  000425  000006          MOV     #425,RKDA(R5)   ;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4612  023046  000405                          BR      71$
4613
4614  023050  005337  001164          70$:    DEC     $TMP2           ;GET SECTOR BEFORE WRL
4615  023054  013765  001164  000006          MOV     $TMP2,RKDA(R5)
4616  023062  016537  000006  001166  71$:    MOV     RKDA(R5),$TMP3  ;FOR ERROR PRINTOUT
4617  023070  032737  000001  001162          BIT     #BIT0,$TMP1
4618  023076  001004                          BNE     72$             ;BR IF WRITING 1'S WHEN WLE OCCURRED
4619  023100  012765  001516  000004          MOV     #DATA0,RKBA(R5) ;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4620  023106  000403                          BR      73$
4621
4622  023110  012765  001522  000004  72$:    MOV     #DATA1,RKBA(R5) ;WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4623  023116  052765  000020  000010  73$:    BIS     #BAI,RKCS2(R5)
4624  023124  012765  177400  000002          MOV     #-256.,RKWC(R5)
4625  023132  017537  000004  001500          MOV     @RKBA(R5),WD2   ;EXPECTED WORD FOR TRUERR TYPEOUT
4626
4627  023140  012765  000031  000000          MOV     #<WRTCHK>,RKCS1(R5)     ;WRITE CHECK CMD
4628  023146  013737  001434  005442          MOV     T50000, TEMP1   ;SETUP TIMEOUT
4629  023154  004737  024226                  JSR     PC,FRDY         ;FIND RDY
4630  023160  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
4631  023162  004737  024652                  JSR     PC,GSTAT        ;GET FRESH STATUS
4632  023166  032737  100000  005400          BIT     #CERR,HCS1
4633  023174  001414                          BEQ     78$
4634
4635  023176  032737  040000  005402          BIT     #WCE,HCS2
4636  023204  001407                          BEQ     77$
4637  023206  016537  000024  001476          MOV     RKDB(R5),WD1    ;GET ACTUAL MISCOMPARED WORD FOR TYPEOUT
4638  023214  004737  026212                  JSR     PC,TRUERR       ;CHECK AGAINST BSE INFO
4639  023220  104137                          ERROR   137             ;WRITE CHECK ERROR
4640  023222  000401                          BR      78$
4641
4642  023224  104022                  77$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
4643
4644  023226                          78$:
4645
4646
4647  023226                          12$:
4648
4649  023226                          ENDRV:
4650
4651                                  ;;*****************************************************************
4652                                  ;*TEST 21       END OF PROGRAM
4653                                  ;*
4654                                  ;*      THIS IS NOT A TEST BUT A LINKAGE TO PERFORM ALL THE
4655                                  ;*      ABOVE TESTS FOR THE NEXT DRIVE PRESENT.
4656                                  ;*      THE NEXT TEST IS ENTERED ONLY AFTER ALL DRIVES PRESENT
4657                                  ;*      HAVE BEEN TESTED.
4658                                  ;*      DO NOT LOOP ON THIS 'TEST'.
4659                                  ;*
```

```
                                ;;*****************************************************************
023226  000004                  TST21:  SCOPE
023230  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
023236  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

023242  005237  001214                  INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
023246  023737  005464  001214          CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
023254  001402                          BEQ     TST22           ;;GO TO NEXT TEST IF YES
023256  000137  013004                  JMP     NUDRV           ;IF NOT, GO BACK & TEST NEXT DRIVE PRESENT.

                                ;;*****************************************************************
                                ;*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 TEST
                                ;*                 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
                                ;*
                                ;*****************************************************************
023262  000004                  TST22:  SCOPE
023264  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
023272  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK PTR

023276  005737  001212                  TST     $PASS
023302  001402                          BEQ     1$              ;DO TEST ONLY IN 1ST PASS
023304  000137  023600                  JMP     11$             ;ELSE EXIT TEST

023310  023727  005464  000001  1$:     CMP     DRIVS,#1
023316  001004                          BNE     2$              ;BR IF MORE THAN 1 DRIVE PRESENT
023320  104401  042047                  TYPE    ,MSG62          ;BYPASS TEST, ONLY 1 DRIVE PRESENT
023324  000137  023600                  JMP     11$             ;ELSE EXIT TEST

023330  104401  036323          2$:     TYPE    ,MSG28          ;MULT DRV DETECTION TEST
023334  104401  040671                  TYPE    ,MSG52          ;PRESS CONT-C TO EXIT OR SPACE TO CONTINUE
023340  004737  027412                  JSR     PC,CCSP         ;INPUT CONTROL-C OR SPACE
023344  000137  023600                  JMP     11$             ;RETURN HERE FOR CONT-C

023350  104401  041504                  TYPE    ,MSG58          ;LOAD HEADS ON ALL DRIVES
023354  104401  037126                  TYPE    ,MSG37          ;PRESS SPACE WHEN SONE
023360  004737  027452                  JSR     PC,GETSP        ;GET SPACE

023364  004737  024714          3$:     JSR     PC,SUBCLR
023370  104024                          ERROR   24              ;CERR AFTER SCLR
```

```
4716  023372  104401  041566              TYPE    ,MSG59          ;INSERT SAME UNIT SEL PLUG # IN 2 DRIVES
4717  023376  104401  037126              TYPE    ,MSG37          ;DEPRESS SPACE BAR WHEN DONE
4718  023402  004737  027452              JSR     PC,GETSP        ;GET SPACE
4719
4720  023406  005000                      CLR     R0              ;DRIVE # COUNTER
4721
4722  023410  012765  100000  000000  6$: MOV     #CCLR,RKCS1(R5)
4723  023416  010065  000010              MOV     R0,RKCS2(R5)    ;DRIVE #
4724  023422  012765  000001  000000      MOV     #SELDRV,RKCS1(R5)  ;SELECT DRIVE CMD TO GET STATUS
4725  023430  013737  001422  005442      MOV     T10,TEMP1
4726  023436  004737  024226              JSR     PC,FRDY         ;FIND CONTR RDY
4727  023442  104117                      ERROR   117             ;NO CONTR RDY
4728
4729  023444  012737  011610  005442      MOV     #5000.,TEMP1
4730  023452  004737  024620              JSR     PC,DLY          ;REQ 2 MS DELAY BEFORE MDS DETECTED
4731
4732  023456  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
4733  023464  010065  000010              MOV     R0,RKCS2(R5)
4734  023470  012765  000001  000000      MOV     #SELDRV,RKCS1(R5)
4735  023476  013737  001422  005442      MOV     T10,TEMP1
4736  023504  004737  024226              JSR     PC,FRDY
4737  023510  104117                      ERROR   117             ;NO CONTR RDY
4738
4739  023512  032737  001000  005402      BIT     #MDS,HCS2       ;SEE IF THAT DRIVE HAS MDS
4740  023520  001006                      BNE     7$              ;BR IF YES
4741  023522  005200                      INC     R0              ;ELSE TRY ANOTHER DRIVE
4742  023524  020027  000010              CMP     R0,#8.          ;SEE IF ALL DRIVES TESTED
4743  023530  001327                      BNE     6$              ;BR IF NO
4744  023532  104176                      ERROR   176             ;CANNOT FIND MDS
4745  023534  000407                      BR      10$             ;TRY AGAIN
4746
4747  023536  104401  042001      7$:     TYPE    ,MSG61          ;MULT DRIVES FOUND ON DRIVE #
4748  023542  010046                      MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
4749                                                              ;;TYPE UNIT NO
4750  023544  104403                      TYPOS                   ;;GO TYPE--OCTAL ASCII
4751  023546     001                      .BYTE   1               ;;TYPE 1 DIGIT(S)
4752  023547     000                      .BYTE   0               ;;SUPPRESS LEADING ZEROS
4753  023550  104401  042735              TYPE    ,MSG68          ;VERIFY BOTH DRIVES UNLOADED
4754  023554  104401  041671      10$:    TYPE    ,MSG60          ;INSERT CORRECT UNIT SEL PLUG & LOAD HEADS
4755  023560  104401  042143              TYPE    ,MSG63          ;DEPRESS CONT-C OR SPACE BAR WHEN DONE
4756  023564  004737  027412              JSR     PC,CCSP         ;INPUT CONT-C OR SPACE
4757  023570  000137  023600              JMP     11$             ;RETURN HERE FOR CONTROL-C
4758  023574  000137  023364              JMP     3$              ;RETURN HERE FOR SPACE
4759
4760  023600  004737  024714      11$:    JSR     PC,SUBCLR
4761  023604  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

023606                          $EOP:
023606  000004                          SCOPE
023610  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
023614  005037  001170                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
023620  005237  001212                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
023624  042737  100000  001212          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
023632  005327                          DEC     (PC)+           ;;LOOP?
023634  000001                  $EOPCT: .WORD   1
023636  003022                          BGT     $DOAGN          ;;YES
023640  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
023642  000001                  $ENDCT: .WORD   1
023644  023634                          $EOPCT
023646  104401  023713                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
023652  013746  001212                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
023656  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
023660  104401  023710                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
023664  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
023670  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
023672  000005                          RESET                   ;;CLEAR THE WORLD
023674  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
023676  000240                          NOP                     ;;SAVE ROOM
023700  000240                          NOP                     ;;FOR
023702  000240                          NOP                     ;;ACT11
023704                          $DOAGN:
023704  000137                          JMP     @(PC)+          ;;RETURN
023706  011342                  $RTNAD: .WORD   ST5
023710     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
023713     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
023720  050040  051501  020123
023726  000043
```

```
                                .SBTTL  SUBROUTINES

                                ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                ;
023730  012700  005454          CLRFLG: MOV     #DDUMP,R0
023734  012701  177757                  MOV     #-17.,R1
023740  005020                  1$:     CLR     (R0)+
023742  005201                          INC     R1
023744  001375                          BNE     1$
023746  000207                          RTS     PC

                                ;
                                ;TYPE PROGRAM ID IF FTITLE=0
                                ;
023750  005737  001354          TITLE:  TST     FTITLE
023754  001004                          BNE     1$
023756  005237  001354                  INC     FTITLE
023762  104401  034062                  TYPE    ,MSG1                   ;PROGRAM ID
023766  000207                  1$:     RTS     PC

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

023770  104411                  GDRVS:  RDLIN
023772  012600                          MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
023774  012701  177770                  MOV     #-8.,R1                 ;SET UP COUNT
024000  112002                  1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
024002  042702  177400                  BIC     #177400,R2              ;MASK HI BYTE
024006  012703  005466                  MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
024012  012704  000060                  MOV     #60,R4

024016  020402                  2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
024020  001415                          BEQ     3$                      ;BRANCH IF YES
024022  005723                          TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
024024  005204                          INC     R4
024026  020427  000070                  CMP     R4,#70
024032  001371                          BNE     2$                      ;S/B 0-7 OR TERMINATOR
024034  005702                          TST     R2
024036  001022                          BNE     4$
024040  020127  177770                  CMP     R1,#-8.
024044  001426                          BEQ     6$                      ;DEFAULT ALL DRIVES
024046  005037  005514          7$:     CLR     SIZFLG                  ;BYPASS TEST 1 (SIZING)
024052  000207                          RTS     PC                      ;FOUND TERMINATOR, EXIT

024054  005213                  3$:     INC     @R3                     ;SET UP FLAG FOR THE DRIVE
024056  005237  005464                  INC     DRIVS                   ;INCREMENT TOTAL # DRIVES TO BE TESTED
024062  112002                          MOVB    (R0)+,R2                ;GET NEXT ASCII CHAR.
024064  042702  177400                  BIC     #177400,R2              ;MASK
024070  022702  000054                  CMP     #54,R2                  ;IS IT A COMMA?
024074  001407                          BEQ     5$                      ;YES, GO TO NEXT WORD.
024076  005702                          TST     R2                      ;NO, IS IT A TERMINATOR?
024100  001001                          BNE     4$                      ;IF NOT, SOMETHING WRONG.
```

```
4856  024102  000761                        BR      7$              ;FOUND TERMINATOR, EXIT
4857
4858  024104  104401  043163        4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
4859  024110  000137  010572                JMP     PRGSRT          ;START ALL OVER
4860
4861  024114  005201                5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
4862  024116  001330                        BNE     1$              ;DRIVES TYPED IN.
4863  024120  000771                        BR      4$              ;IF NORE, HAVE ERROR.
4864
4865  024122  005237  005514        6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
4866  024126  000207                        RTS     PC              ;EXIT.
4867
4868                                ;
4869                                ;ROUTINE TO INPUT RKBAS OR DEFAULT.
4870                                ;
4871
4872  024130  104412                GBA:    RDOCT
4873  024132  012600                        MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
4874  024134  005700                        TST     R0
4875  024136  001403                        BEQ     1$              ;BRANCH IF DEFAULT.
4876  024140  010037  001260                MOV     R0,$BASE
4877  024144  000207                        RTS     PC
4878  024146  012737  177440  001260  1$:  MOV     #177440,$BASE   ;DEFAULT VALUE
4879  024154  000207                        RTS     PC
4880
4881                                ;
4882                                ;ROUTINE TO INPUT RKVEC OR DEFAULT
4883                                ;
4884
4885  024156  104412                GINT:   RDOCT
4886  024160  012600                        MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
4887  024162  005700                        TST     R0
4888  024164  001405                        BEQ     1$              ;BRANCH IF DEFAULT
4889  024166  010037  001330                MOV     R0,RKVEC
4890  024172  004737  024210        2$:     JSR     PC,SETINT
4891  024176  000207                        RTS     PC
4892  024200  012737  000210  001330  1$:  MOV     #210,RKVEC      ;DEFAULT VALUE
4893  024206  000771                        BR      2$
4894
4895                                ;
4896                                ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
4897                                ;
4898
4899  024210  013700  001330        SETINT: MOV     RKVEC,R0
4900  024214  012720  027716                MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
4901  024220  013710  001332                MOV     RKPRI,(R0)      ;PR5 TO RKVEC+2
4902  024224  000207                        RTS     PC
4903
4904
4905                                ;
```

```
                                        ;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
                                        ;
024226  032765  000200  000000 FRDY:    BIT     #RDY,RKCS1(R5)
024234  001006                          BNE     1$
024236  005337  005442                  DEC     TEMP1
024242  001371                          BNE     FRDY
024244  004737  024264                  JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
024250  000207                          RTS     PC              ;NO RDY, EXIT
024252  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
024256  004737  024264                  JSR     PC,HOLD
024262  000207                          RTS     PC

                                        ;
                                        ;STORE ALL RK611 REGISTERS IN HOLDING REGS
                                        ;

024264  016537  000000  005400 HOLD:    MOV     RKCS1(R5),HCS1
024272  016537  000010  005402          MOV     RKCS2(R5),HCS2
024300  016537  000002  005404          MOV     RKWC(R5),HWC
024306  016537  000004  005406          MOV     RKBA(R5),HBA
024314  016537  000006  005410          MOV     RKDA(R5),HDA
024322  016537  000012  005412          MOV     RKDS(R5),HDS
024330  016537  000014  005414          MOV     RKER(R5),HER
024336  016537  000016  005416          MOV     RKASOF(R5),HASOF
024344  016537  000020  005420          MOV     RKDC(R5),HDC
024352  016537  000026  005424          MOV     RKMR1(R5),HMR1
024360  016537  000034  005426          MOV     RKMR2(R5),HMR2
024366  016537  000036  005430          MOV     RKMR3(R5),HMR3
024374  016537  000030  005432          MOV     RKECPS(R5),HPOS
024402  016537  000032  005434          MOV     RKECPT(R5),HPAT
024410  000207                          RTS     PC

                                        ;
                                        ;ROUTINE TO CHECK FOR CORRECT ATTN
                                        ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
                                        ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
                                        ;
024412  010446                  TSTATN: MOV     R4,-(SP)        ;SAV R4
024414  013704  001216                  MOV     $UNIT,R4
024420  136437  005370  005417          BITB    ATTN(R4),HASOF+1
024426  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
024430  012604                          MOV     (SP)+,R4        ;RESTOR R4
024432  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
024436  000207                          RTS     PC
024440  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
024442  000207                          RTS     PC
```

```
4956
4957                                    ;
4958                                    ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
4959                                    ;ENTER WITH TIME IN SECONDS IN TEMP2
4960                                    ;RETURN IF NO ATTN (ERROR CONDITION)
4961                                    ;RETURN +2 IF ATTN FOUND
4962                                    ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
4963                                    ;
4964
4965  024444  010446                    FATT1:  MOV     R4,-(SP)        ;SAV R4
4966  024446  012737  177777  005442    3$:     MOV     #-1,TEMP1
4967  024454  013704  001216            MOV     $UNIT,R4
4968  024460  136465  005370  000017    1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
4969  024466  001014                    BNE     2$
4970  024470  005337  005442            DEC     TEMP1
4971  024474  001371                    BNE     1$
4972  024476  005337  005444            DEC     TEMP2
4973  024502  001361                    BNE     3$
4974  024504  005065  000026            CLR     RKMR1(R5)       ;SELECT WORD 0
4975  024510  004737  024652            JSR     PC,GSTAT        ;GET LATEST STATUS
4976  024514  012604                    MOV     (SP)+,R4        ;RESTOR R4
4977  024516  000207                    RTS     PC
```

```
 4978  024520  005065  000026          2$:     CLR     RKMR1(R5)
 4979  024524  004737  024652                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
 4980  024530  012604                          MOV     (SP)+,R4        ;RESTOR R4
 4981  024532  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
 4982  024536  000207                          RTS     PC
 4983
 4984                                  ;
 4985                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
 4986                                  ;ENTER WITH COUNT IN TEMP1
 4987                                  ;RETURN IF NO ATTN (ERROR)
 4988                                  ;RETURN +2 IF ATTN FOUND
 4989                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 4990                                  ;
 4991
 4992  024540  010446                  FATT2:  MOV     R4,-(SP)        ;SAV R4
 4993  024542  013704  001216          2$:     MOV     $UNIT,R4
 4994  024546  136465  005370  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
 4995  024554  001011                          BNE     1$
 4996  024556  005337  005442                  DEC     TEMP1
 4997  024562  001367                          BNE     2$
 4998  024564  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
 4999  024570  004737  024652                  JSR     PC,GSTAT        ;GET LATEST STATUS.
 5000  024574  012604                          MOV     (SP)+,R4        ;RESTOR R4
 5001  024576  000207                          RTS     PC
 5002  024600  005065  000026          1$:     CLR     RKMR1(R5)
 5003  024604  004737  024652                  JSR     PC,GSTAT
 5004  024610  012604                          MOV     (SP)+,R4        ;RESTOR R4
 5005  024612  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
 5006  024616  000207                          RTS     PC
 5007                                  ;
 5008                                  ;ENTER WITH A COUNT IN TEMP1
 5009                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
 5010                                  ;WHEN COUNT IS 0. BASED ON AN 11/05
 5011                                  ;
 5012  024620  005737  005442          DLY:    TST     TEMP1           ;5.6 US
 5013  024624  001403                          BEQ     1$              ;2.5 US
 5014  024626  005337  005442                  DEC     TEMP1           ;6.8 US
 5015  024632  000772                          BR      DLY             ;2.5 US
 5016  024634  000207                  1$:     RTS     PC              ;3.8 US
 5017                                  ;
 5018                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
 5019                                  ;
 5020
 5021  024636  104401  035533          BYP:    TYPE    ,MSG14          ;BYPASS DRIVE
 5022  024642  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 5023                                                                  ;;TYPE DR#
 5024  024644  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 5025  024646     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
 5026  024647     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 5027  024650  000207                          RTS     PC
 5028
 5029                                  ;
 5030                                  ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
 5031                                  ;IT THEN WAITS FOR CONTROLLER READY
```

```
                                        ;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
                                        ;
024652  013746  005442          GSTAT:  MOV     TEMP1,-(SP)      ;SAVE TEMP1
024656  013765  001216  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
024664  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;GET STATUS WITH SELECT DRIVE CMD
024672  013737  001422  005442          MOV     T10,TEMP1
024700  004737  024226                  JSR     PC,FRDY          ;FIND RDY
024704  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRIVE CMD
024706  012637  005442                  MOV     (SP)+,TEMP1      ;RESTOR TEMP1.
024712  000207                          RTS     PC

                                        ;
                                        ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
                                        ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
                                        ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
                                        ;RETURN IF CERR SET
                                        ;RETURN +2 IF CERR CLEAR
                                        ;
024714  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
024722  013737  001422  005442          MOV     T10,TEMP1
024730  004737  024226                  JSR     PC,FRDY          ;FIND RDY
024734  104120                          ERROR   120              ;RDY NOT SET BY END OF SCLR
024736  013765  001216  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
024744  005065  000026                  CLR     RKMR1(R5)        ;SELECT WORD 0
024750  004737  024652                  JSR     PC,GSTAT         ;GET STATUS
024754  032737  100000  005400          BIT     #CERR,HCS1       ;CHECK FOR CONT ERROR
024762  001401                          BEQ     1$
024764  000207                          RTS     PC
024766  062716  000002          1$:     ADD     #2,(SP)          ;SKIP OVER ERROR
024772  000207                          RTS     PC

                                        ;
                                        ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                        ;
024774  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)     ;WORD 3
025002  004737  024652                  JSR     PC,GSTAT
025006  013737  005430  001416          MOV     HMR3,SECTOR
025014  042737  177017  001416          BIC     #^C<M.SECT>,SECTOR
025022  006237  001416                  ASR     SECTOR           ;RIGHT JUSTIFY
025026  006237  001416                  ASR     SECTOR           ;SECTOR
025032  006237  001416                  ASR     SECTOR           ;INFO
025036  006237  001416                  ASR     SECTOR
025042  000207                          RTS     PC

                                        ;
                                        ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
                                        ;
025044  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)     ;WORD 2
025052  004737  024652                  JSR     PC,GSTAT
025056  013737  005426  001372          MOV     HMR2,CYLDIF
```

```
5082  025064  042737  160017  001372          BIC     #^C<M.CDIF>,CYLDIF
5083  025072  006237  001372                  ASR     CYLDIF          ;RIGHT JUSTIFY
5084  025076  006237  001372                  ASR     CYLDIF          ;CYL DIFF/OFFSET
5085  025102  006237  001372                  ASR     CYLDIF          ;INFO
5086  025106  006237  001372                  ASR     CYLDIF
5087  025112  023727  001372  000777          CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
5088  025120  001002                          BNE     1$              ;BR IF NOT
5089  025122  005037  001372                  CLR     CYLDIF          ;CLR IF YES
5090  025126  000207                  1$:     RTS     PC
5091                                  ;
5092                                  ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
5093                                  ;
5094  025130  012765  000002  000026  RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
5095  025136  004737  024652                  JSR     PC,GSTAT
5096  025142  013737  005430  001374          MOV     HMR3,CYLADD
5097  025150  042737  160017  001374          BIC     #^C<M.CADD>,CYLADD
5098  025156  006237  001374                  ASR     CYLADD          ;RIGHT JUSTIFY
5099  025162  006237  001374                  ASR     CYLADD          ;CYL ADDR
5100  025166  006237  001374                  ASR     CYLADD          ;INFO
5101  025172  006237  001374                  ASR     CYLADD
5102  025176  000207                          RTS     PC
5103                                  ;
5104                                  ;FIND SECTOR 17
5105                                  ;RETURN IF NOT FOUND
5106                                  ;RETURN +4 IF FOUND
5107                                  ;
5108  025200  013737  001432  005442  FSEC17: MOV     T5000,TEMP1     ;SETUP TIMEOUT
5109  025206  004737  024774          1$:     JSR     PC,RDSEC        ;READ SECTOR
5110  025212  023727  001416  000021          CMP     SECTOR,#17.     ;TEST FOR SECTOR 17
5111  025220  001014                          BNE     2$              ;BR IF NOT 17
5112
5113  025222  004737  024774                  JSR     PC,RDSEC
5114  025226  023727  001416  000021          CMP     SECTOR,#17.
5115  025234  001412                          BEQ     3$              ;BR IF READ SAME TWICE
5116  025236  004737  024774                  JSR     PC,RDSEC        ;ELSE TRY 1 MORE TIME
5117  025242  023727  001416  000021          CMP     SECTOR,#17.
5118  025250  001404                          BEQ     3$              ;BR IF 17
5119
5120  025252  005337  005442          2$:     DEC     TEMP1
5121  025256  001353                          BNE     1$              ;TRY AGAIN
5122  025260  000207                          RTS     PC
5123
5124  025262  062716  000004          3$:     ADD     #4,(SP)         ;SKIP OVER ERROR
5125  025266  000207                          RTS     PC
5126                                  ;
5127                                  ;FIND DESIRED CYL DIFF
5128                                  ;RETURN IF NOT FOUND
5129                                  ;RETURN+6 IF FOUND
5130                                  ;
```

```
5131  025270  013737  001432  005442  FCYL:   MOV     T5000,TEMP1     ;SETUP TIMEOUT
5132  025276  004737  025044          1$:     JSR     PC,RDCYLD
5133  025302  023737  001372  005444          CMP     CYLDIF,TEMP2    ;TEST FOR CYL DIFF
5134  025310  001014                          BNE     2$              ;BR IF NOT FOUND
5135
5136  025312  004737  025044                  JSR     PC,RDCYLD
5137  025316  023737  001372  005444          CMP     CYLDIF,TEMP2
5138  025324  001412                          BEQ     3$              ;BR IF READ SAME TWICE
5139  025326  004737  025044                  JSR     PC,RDCYLD       ;ELSE TRY 1 MORE TIME
5140  025332  023737  001372  005444          CMP     CYLDIF,TEMP2
5141  025340  001404                          BEQ     3$
5142
5143  025342  005337  005442          2$:     DEC     TEMP1
5144  025346  001353                          BNE     1$              ;TRY AGAIN
5145  025350  000207                          RTS     PC
5146
5147  025352  062716  000006          3$:     ADD     #6,(SP)         ;SKIP OVER ERROR
5148  025356  000207                          RTS     PC
5149                                  ;
5150                                  ;FIND DESIRED CYL DIFF WHILE GOING THRU LOOP
5151                                  ;RETURN IF NOT FOUND
5152                                  ;RETURN+6 IF NOT FOUND
5153                                  ;
5154  025360  005037  001462          FCYLOP: CLR     LPCNT
5155  025364  006337  005444                  ASL     TEMP2
5156  025370  006337  005444                  ASL     TEMP2
5157  025374  006337  005444                  ASL     TEMP2
5158  025400  006337  005444                  ASL     TEMP2
5159
5160  025404  004737  025470          1$:     JSR     PC,QKCYLD       ;GET CYL DIFF
5161  025410  023737  001372  005444          CMP     CYLDIF,TEMP2
5162  025416  001014                          BNE     2$              ;BR IF NOT FFOUND
5163
5164  025420  004737  025470                  JSR     PC,QKCYLD
5165  025424  023737  001372  005444          CMP     CYLDIF,TEMP2
5166  025432  001413                          BEQ     3$              ;BR IF READ SAME TWICE
5167  025434  004737  025470                  JSR     PC,QKCYLD       ;ELSE TRY 1 MORE TIME
5168  025440  023737  001372  005444          CMP     CYLDIF,TEMP2
5169  025446  001405                          BEQ     3$
5170
5171  025450  004737  026740          2$:     JSR     PC,LOOP
5172  025454  005737  001462                  TST     LPCNT           ;SEE IF OVERFLOW
5173  025460  001351                          BNE     1$              ;BR IF NO
5174
5175  025462  062716  000006          3$:     ADD     #6,(SP)         ;SKIP OVER ERROR
5176  025466  000207                          RTS     PC
5177                                  ;
5178                                  ;QUICK SELECT DRIVE COMMAND TO OBTAIN CYL DIFF
5179                                  ;
5180  025470  013746  005442          QKCYLD: MOV     TEMP1,-(SP)     ;SAVE TEMP1
```

```
5181  025474  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
5182  025502  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)       ;SELECT DRIVE CMD
5183  025510  013737  001422  005442          MOV     T10,TEMP1
5184  025516  032765  000200  000000  1$:     BIT     #RDY,RKCS1(R5)  ;TEST FOR CONT RDY
5185  025524  001004                          BNE     2$              ;BR IF THERE
5186  025526  005337  005442                  DEC     TEMP1
5187  025532  001371                          BNE     1$
5188  025534  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5189
5190  025536  016537  000034  001372  2$:     MOV     RKMR2(R5),CYLDIF
5191  025544  042737  160017  001372          BIC     #^C<M.CDIF>,CYLDIF      ;GET CYL DIFF INFO (NO SHIFTING)
5192  025552  012637  005442                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
5193  025556  000207                          RTS     PC
5194                                  ;
5195                                  ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
5196                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
5197                                  ;RETURN IF NOT FOUND
5198                                  ;RETURN+2 IF FOUND - SKIP OVER ERROR
5199                                  ;
5200  025560  012737  177777  005442  FHDHM:  MOV     #-1,TEMP1       ;ALL 1'S
5201  025566  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
5202  025574  004737  024652          1$:     JSR     PC,GSTAT
5203  025600  032737  000040  005426          BIT     #D.HDHM,HMR2
5204  025606  001007                          BNE     2$
5205  025610  005337  005442                  DEC     TEMP1
5206  025614  001367                          BNE     1$
```

```
5207  025616  005337  005444          DEC     TEMP2
5208  025622  001356                  BNE     FHDHM
5209  025624  000207                  RTS     PC
5210  025626  062716  000002  2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
5211  025632  000207                  RTS     PC
5212                                  ;
5213                                  ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE 5MS
5214                                  ;RETURN IF NOT FOUND
5215                                  ;RETURN+2 IF FOUND: SKIP OVER ERROR
5216                                  ;
5217  025634  012737  000372  005442  FLOAD:  MOV     #250.,TEMP1
5218  025642  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
5219  025650  004737  024652          1$:     JSR     PC,GSTAT
5220  025654  032737  010000  005426          BIT     #D.LOAD,HMR2
5221  025662  001004                  BNE     2$
5222  025664  005337  005442          DEC     TEMP1
5223  025670  001367                  BNE     1$
5224  025672  000207                  RTS     PC
5225  025674  062716  000002  2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
5226  025700  000207                  RTS     PC
5227                                  ;
5228                                  ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
5229                                  ;ENTER WITH CYL # IN 'CALADD'
5230                                  ;ENTER WITH HEAD # IN 'HEAD'
5231                                  ;ENTER WITH FORMAT IN 'FORMAT'
5232                                  ;
5233  025702  010046          FHDTAB: MOV     R0,-(SP)        ;SAV R0
5234  025704  010146                  MOV     R1,-(SP)        ;SAV R1
5235  025706  012700  001542          MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
5236  025712  005001                  CLR     R1              ;SECTOR COUNTER
5237  025714  013737  001504  001506  MOV     HEAD,HD1
5238  025722  006337  001506          ASL     HD1
5239  025726  006337  001506          ASL     HD1
5240  025732  006337  001506          ASL     HD1
5241  025736  006337  001506          ASL     HD1
5242  025742  006337  001506          ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
5243  025746  013737  001510  001512  MOV     FORMAT,FMT1
5244  025754  000337  001512          SWAB    FMT1
5245  025760  006337  001512          ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
5246
5247  025764  013720  001376  1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
5248  025770  010110                  MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
5249  025772  053710  001506          BIS     HD1,(R0)        ;               -HEAD NO
5250  025776  053710  001512          BIS     FMT1,(R0)       ;               -FORMAT
5251  026002  052710  140000          BIS     #<BIT14!BIT15>,(R0)     ;       -GOOD SECTOR FLAGS
5252
5253  026006  013737  001376  005442          MOV     CALADD,TEMP1
5254  026014  011037  005444                  MOV     (R0),TEMP2
5255  026020  043737  001376  005444          BIC     CALADD,TEMP2
5256  026026  042037  005442                  BIC     (R0)+,TEMP1
5257  026032  053737  005442  005444          BIS     TEMP1,TEMP2
5258  026040  013720  005444                  MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
5259
5260  026044  005201                  INC     R1              ;SECTOR CTR
5261  026046  020127  000026          CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
5262  026052  001344                  BNE     1$              ;BR IF NO
```

```
5263
5264  026054  012601                        MOV     (SP)+,R1        ;RESTOR R1
5265  026056  012600                        MOV     (SP)+,R0        ;RESTOR R0
5266  026060  000207                        RTS     PC
5267
5268                                  ;
5269                                  ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
5270                                  ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
5271                                  ;
5272  026062  010046                SORT:   MOV     R0,-(SP)        ;SAVE R0
5273  026064  010146                        MOV     R1,-(SP)        ;SAVE R1
5274  026066  004737  024774                JSR     PC,RDSEC
5275  026072  062737  000001  001416        ADD     #1,SECTOR
5276  026100  004737  026170                JSR     PC,MULT6        ;MULT SECTOR BY 6
5277
5278  026104  012700  000204                MOV     #132.,R0
5279  026110  163700  001416                SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
5280  026114  010037  001416                MOV     R0,SECTOR
5281  026120  062737  001746  001416        ADD     #RHTAB,SECTOR   ;SAVE INDEX
5282
5283  026126  062700  001746                ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
5284  026132  012701  002152                MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
```

```
5285
5286  026136  012021                  1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
5287  026140  020027  002152                  CMP     R0,#RHTAB+132.
5288  026144  001374                         BNE     1$
5289
5290  026146  012700  001746                  MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
5291  026152  012021                  2$:     MOV     (R0)+,(R1)+
5292  026154  020037  001416                  CMP     R0,SECTOR
5293  026160  001374                         BNE     2$
5294
5295  026162  012601                         MOV     (SP)+,R1        ;RESTOR R1
5296  026164  012600                         MOV     (SP)+,R0        ;RESTOR R0
5297  026166  000207                         RTS     PC
5298
5299                                  ;
5300                                  ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
5301                                  ;
5302  026170  006337  001416          MULT6: ASL     SECTOR          ;2 X SECTOR
5303  026174  013746  001416                 MOV     SECTOR,-(SP)
5304  026200  006337  001416                 ASL     SECTOR          ;4 X SECTOR
5305  026204  062637  001416                 ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
5306  026210  000207                         RTS     PC
5307
5308                                  ;
5309                                  ;THIS ROUTINE IS ENTERED IF THERE IS A READ HEADER MISMATCH.
5310                                  ;IT COMPARES THE CYLINDER, SECTOR AND HEAD NUMBERS OF WHERE THE
5311                                  ;HEADS WERE AT THE TIME OF READING AGAINST THE BSE INFO
5312                                  ;OBTAINED IN A PREVIOUS TEST.
5313                                  ;RETURN IF NO COMPARE: ERROR CONDITION
5314                                  ;RETURN+2 IF COMPARE FOUND: SKIP OVER ERROR
5315                                  ;
5316  026212  010446                  TRUERR: MOV    R4,-(SP)        ;SAVE R4
5317  026214  013746  001160                  MOV    $TMP0,-(SP)     ;SAVE
5318  026220  013746  001162                  MOV    $TMP1,-(SP)     ;SAVE
5319
5320  026224  013737  005410  001160          MOV    HDA,$TMP0       ;READ TRK SECTOR INFO
5321  026232  013737  005410  001162          MOV    HDA,$TMP1
5322
5323  026240  042737  177740  001160          BIC    #^C<37>,$TMP0   ;TMP0 HAS SECTOR INFO
5324  026246  042737  174377  001162          BIC    #^C<3400>,$TMP1
5325  026254  000337  001162                  SWAB   $TMP1           ;TMP1 HAS HEAD INFO
5326
5327  026260  005737  001160                  TST    $TMP0           ;SEE IF SECTOR 0
5328  026264  001414                          BEQ    6$
5329  026266  005337  001160                  DEC    $TMP0           ;GET ACTUAL WLE SECTOR
5330  026272  013737  001160  001416          MOV    $TMP0,SECTOR    ;STORE SECTOR
5331  026300  013737  001162  001504          MOV    $TMP1,HEAD      ;STORE HEAD
5332  026306  013737  005420  001364          MOV    HDC,CCYL        ;STORE CYLINDER
5333  026314  000440                          BR     9$
5334
5335  026316  005737  001162          6$:     TST    $TMP1           ;SEE IF HEAD 0
5336  026322  001414                          BEQ    7$              ;BR IF YES
5337  026324  005337  001162                  DEC    $TMP1           ;GET ACTUAL WLE HEAD
5338  026330  013737  001162  001504          MOV    $TMP1,HEAD
5339  026336  012737  000025  001416          MOV    #21.,SECTOR
5340  026344  013737  005420  001364          MOV    HDC,CCYL
```

```
5341  026352  000421                          BR 9$
5342
5343  026354  005737  005420          7$:     TST     HDC             ;SEE IF CYL 0
5344  026360  001414                          BEQ     8$
5345  026362  005337  005420                  DEC     HDC             ;GET ACTUAL WLE CYL
5346  026366  013737  005420  001364          MOV     HDC,CCYL
5347  026374  012737  000002  001504          MOV     #2,HEAD
5348  026402  012737  000025  001416          MOV     #21.,SECTOR
5349  026410  000402                          BR      9$
5350
5351  026412  104206                  8$:     ERROR   206             ;RKDC & RKDA INDICATES WCE
5352  026414  000453                          BR      5$              ;OCCURRED AT CYL 411,HD 2 SECTOR 21
5353
5354  026416  012704  003366          9$:     MOV     #BSE22+8.,R4
5355  026422  012437  001530          1$:     MOV     (R4)+,WORD      ;GET CYL# OFF BSE TABLE
5356  026426  023727  001530  177777          CMP     WORD,#-1        ;SEE IF ALL 1'S
5357  026434  001406                          BEQ     2$              ;EXIT IF YES
5358  026436  023737  001530  001364          CMP     WORD,CCYL       ;COMPARE CYL #
5359  026444  001410                          BEQ     3$              ;BR IF CYL MATCH
5360  026446  005724                          TST     (R4)+           ;ADV TO NEXT CYL WORD
5361  026450  000764                          BR      1$
5362
5363  026452  012637  001162          2$:     MOV     (SP)+,$TMP1     ;RESTORE
5364  026456  012637  001160                  MOV     (SP)+,$TMP0     ;RESTORE
5365  026462  012604                          MOV     (SP)+,R4        ;RESTOR R4
5366  026464  000207                          RTS     PC
5367
5368  026466  011437  001530          3$:     MOV     (R4),WORD       ;GET HEAD & SECTOR FROM BSE TABLE
5369  026472  042737  177400  001530          BIC     #177400,WORD    ;KEEP SECTOR# ONLY
5370  026500  023737  001530  001416          CMP     WORD,SECTOR     ;SECTOR COMPARE?
5371  026506  001402                          BEQ     4$              ;BR IF YES
5372  026510  005724                          TST     (R4)+           ;ELSE GET NEXT CYL # OFF TABLE
5373  026512  000743                          BR      1$
5374
5375  026514  012437  001530          4$:     MOV     (R4)+,WORD      ;GET HEAD & SECTOR FROM BSE TABLE
5376  026520  000337  001530                  SWAB    WORD
5377  026524  042737  177400  001530          BIC     #177400,WORD    ;KEEP HEAD# ONLY
5378  026532  023737  001530  001504          CMP     WORD,HEAD       ;HEAD COMPARE?
5379  026540  001401                          BEQ     5$              ;BR IF YES
5380  026542  000727                          BR      1$
5381
5382  026544  012637  001162          5$:     MOV     (SP)+,$TMP1     ;RESTOR
5383  026550  012637  001160                  MOV     (SP)+,$TMP0     ;RESTOR
5384  026554  012604                          MOV     (SP)+,R4        ;RESTOR R4
5385  026556  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERR ON RETURN
5386  026562  000207                          RTS     PC
5387
5388                                  ;
5389                                  ;THIS ROUTINE CALIBRATES ANY PDP-11 AGAINST 2 CONSECUTIVE 16MS TICKS FROM AN L OR P CLOC
5390                                  ;
                                      ;THE 1ST TICK CLEARS A COUNTER 'LPCNT'. THE PROGRAM THEN GOES THRU A LOOP.
```

```
5391                    ;EACH TIME IT GOES THRU THE LOOP,LPCNT IS INCREMENTED BY 1.
5392                    ;THE 2ND TICK STOPS INCREMENTING LPCNT AND THE CLOCK INTERRUPT IS TURNED OFF.
5393                    ;NOW, FOR THIS PARTICULAR PDP-11, THE TIME TO GO THRU THE ABOVE LOOP ONE IS
5394                    ;16.6MS DIVIDED BY THE CONTENTS OF LPCNT.
5395                    ;
5396                    ;AN APPROX VALUE FOR A PDP11-05 MAY BE 0.4MS TO GO THRU THE LOOP ONCE
5397                    ;
5398                    ;FROM THIS POINT ON, WHENEVER ACCURACIES GREATER THAN WHAT THE L CLOCK
5399                    ;CAN PROVIDE ARE NECESSARY, THE EVENT TO BE TIMED IS COMPARED AGAINST
5400                    ;THE NUMBER OF TIMES THE LOOP IS PASSED THRU AND MULT. BY THE ABOVE
5401                    ;CALCULATED FIGURE
5402                    ;
```

```
5403
5404  026564  104413                  CALCLK: SAVREG
5405  026566  012737  000001  001402          MOV     #1,COUNT        ;GO THRU CLOCK HANDLER
5406  026574  012737  000001  001404          MOV     #1,SEC          ;ONLY ONCE
5407  026602  012737  177777  005442          MOV     #-1,TEMP1       ;ALL 1'S FOR TIMEOUT
5408  026610  004737  026764                  JSR     PC,CLKON
5409
5410
5411  026614  005737  001406          1$:     TST     TIMUP
5412  026620  001007                          BNE     2$              ;BR IF GOT 1'ST TICK
5413  026622  005337  005442                  DEC     TEMP1
5414  026626  001372                          BNE     1$              ;BR IF TIMEOUT NOT DONE
5415  026630  104401  036102                  TYPE    ,MSG23          ;NO CLOCK INTR PRESENT, ABORT TIMING TEST
5416  026634  000137  023226                  JMP     ENDRV           ;ABORT DRIVE
5417
5418  026640  012737  000001  001402  2$:     MOV     #1,COUNT        ;GOT 1'ST TICK
5419  026646  012737  000001  001404          MOV     #1,SEC
5420  026654  005037  001462                  CLR     LPCNT           ;LOOP COUNTER
5421  026660  005037  001406                  CLR     TIMUP           ;CLEAR BEFORE 2'ND TICK
5422
5423  026664  005737  001406          3$:     TST     TIMUP
5424  026670  001003                          BNE     4$              ;BR IF GOT 2'ND TICK
5425  026672  004737  026740                  JSR     PC,LOOP
5426  026676  000772                          BR      3$
5427
5428  026700  004737  027072          4$:     JSR     PC,CLKOF        ;GOT 2'ND TICK, TURN OFF CLOCK
5429
5430  026704  005000                          CLR     R0              ;CLEAR FOR DIV
5431  026706  005001                          CLR     R1
5432  026710  005002                          CLR     R2
5433  026712  005004                          CLR     R4
5434  026714  012703  040432                  MOV     #16666.,R3      ;LSB DIVIDEND (16666 US)
5435  026720  013705  001462                  MOV     LPCNT,R5        ;DIVISOR
5436  026724  004737  030040                  JSR     PC,M.DPID       ;DO DIVIDE
5437  026730  010337  001464                  MOV     R3,LPTIM        ;STORE QUOTIENT
5438                                                                  ;THIS EQUALS THE TIME IN USEC
5439                                                                  ;TO GO THRU THE LOOP ONCE
5440  026734  104414                          RESREG
5441  026736  000207                          RTS     PC
5442
5443
5444                                  ;
5445                                  ;WITH 50 IN R0, IT TAKES APPROX 400 US FOR AN 11/05 TO GO THRU THIS
5446                                  ;LOOP AND INCREMENT 'LPCNT' ONCE.
5447                                  ;THIS 400 US IS APPROX 2.5% OF 16MS GIVING A RESOLUTION OF 0.4MS
5448                                  ;PER COUNT IN 'LPCNT'.
5449                                  ;WHEN USED BY THE 'CALCLK' ROUTINE, LPCLK SHOULD BE APPROX 40(10)=50(8)
5450                                  ;WITH AN 11/05 TO 200(10)=264(8) FOR AN 11/70
5451                                  ;
5452  026740  010046                  LOOP:   MOV     R0,-(SP)        ;SAVE R0
5453  026742  012700  000062                  MOV     #50.,R0
5454  026746  005300                  1$:     DEC     R0
5455  026750  001401                          BEQ     2$
5456  026752  000775                          BR      1$
5457  026754  005237  001462          2$:     INC     LPCNT
5458  026760  012600                          MOV     (SP)+,R0        ;RESTORE R0
```

```
                        RTS     PC
                ;
                ;
                ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
                ;
CLKON:  MOV     #PR0,-(SP)      ;PSW LOADED TO BE
        MOV     #2$,-(SP)       ;LSI-11 COMPATABLE
        RTI                     ;ENABLE CLOCK INTERRUPTS
2$:     CLR     TIMUP
        TST     PCLKF
        BNE     1$              ;BRANCH IF P-CLOCK PRESENT
        MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
        RTS     PC
1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
        MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
        RTS     PC              ;LINE FREQ & RUN
                ;
                ;KW11-L & KW11-P INTERRUPT HANDLER
                ;
CLOCK:  CLR     TIMUP
        DEC     COUNT
        BNE     1$
        MOV     HZ,COUNT
        DEC     SEC
        BNE     1$
        INC     TIMUP           ;SORRY, TIME IS UP
1$:     RTI
                ;
                ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
                ;
CLKOF:  MOV     #PR7,-(SP)      ;PSW LOADED TO BE
        MOV     #2$,-(SP)       ;LSI-11 COMPATABLE
        RTI                     ;LOCK OUT ALL INTERRUPTS
2$:     TST     PCLKF
        BNE     1$              ;BRACH IF P-CLOCK PRESENT
        CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
        RTS     PC
1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
        RTS     PC
                ;
                ;THIS ROUTINE CHECKS RKMR2 (MSGA)-WORDS 0 & 1
                ;ENTER WITH SHOULD BE VALUE IN SBMR2.
                ;RETURN IF NO COMPARE, RETURN +2 IF COMPARE
                ;
CKMR2:  MOV     TEMP1,-(SP)     ;SAV TEMP1
        BIS     $UNIT,SBMR2     ;INSERT DRIVE #
        MOV     SBMR2,TEMP1
        JSR     PC,SBPAR        ;GET PARITY FOR SBMR2
        MOV     TEMP1,SBMR2     ;NOW HAS PARITY
        CMP     SBMR2,HMR2      ;SHOULD BE SAME
        BNE     1$
        MOV     (SP)+,TEMP1     ;RESTOR TEMP1
        ADD     #2,(SP)         ;COMPARE OK, SKIP OVER ERROR.
        RTS     PC
1$:     MOV     (SP)+,TEMP1     ;RESTOR TEMP1
        RTS     PC
```

```
                                        ;
                                        ;THIS ROUTINE CHECKS RKMR3 (MSGB)-WORDS 0 & 1
                                        ;ENTER WITH SHOULD BE VALUE IN SBMR3
                                        ;RETURN IF NO COMPARE, RETURN +2 IF COMPARE
                                        ;
027210  013746  005442        CKMR3:  MOV     TEMP1,-(SP)     ;SAV TEMP1
027214  013737  005424  005442        MOV     HMR1,TEMP1
027222  042737  177774  005442        BIC     #^C<M.ID>,TEMP1
027230  053737  005442  005440        BIS     TEMP1,SBMR3     ;INSERT WORD #
027236  013737  005440  005442        MOV     SBMR3,TEMP1
027244  004737  027306                JSR     PC,SBPAR        ;GET PARITY FOR SBMR3
027250  013737  005442  005440        MOV     TEMP1,SBMR3     ;NOW HAS PARITY
027256  023737  005440  005430        CMP     SBMR3,HMR3      ;SHOULD BE SAME
027264  001005                        BNE     1$
027266  012637  005442                MOV     (SP)+,TEMP1     ;RESTOR TEMP1
027272  062716  000002                ADD     #2,(SP)         ;COMPARE OK, SKIP OVER ERROR.
027276  000207                        RTS     PC
027300  012637  005442        1$:     MOV     (SP)+,TEMP1     ;RESTOR TEMP1
027304  000207                        RTS     PC
                                        ;
                                        ;THIS ROUTINE GENERATES PARITY FOR SBMR2 AND SBMR3
                                        ;ENTER WITH SBMR2 / SBMR3 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
                                        ;
027306  010046                SBPAR:  MOV     R0,-(SP)        ;SAVE R0
027310  010146                        MOV     R1,-(SP)        ;SAVE R1
027312  012700  000021                MOV     #17.,R0         ;SHIFT COUNTER
027316  005001                        CLR     R1              ;COUNT # OF 1'S IN TEMP1
027320  000241                        CLC                     ;CLEAR CARRY

027322  006137  005442        1$:     ROL     TEMP1
027326  103001                        BCC     2$              ;BR IF CARRY CLEAR
027330  005201                        INC     R1              ;COUNT # OF 1'S
027332  005300                2$:     DEC     R0              ;SHIFT COUNTER
027334  001372                        BNE     1$

027336  032701  000001                BIT     #BIT0,R1
027342  001003                        BNE     3$              ;BR IF ODD # IN R0
027344  052737  100000  005442        BIS     #M.PAR,TEMP1    ;SET PARITY BIT
027352  012601                3$:     MOV     (SP)+,R1        ;RESTORE R1
027354  012600                        MOV     (SP)+,R0        ;RESTORE R0
027356  000207                        RTS     PC

                                        ;
                                        ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                        ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                        ; IE: MY LOOP MACRO
                                        ;
```

```
5571  027360  032777  001000  151552  SCOP1$:  BIT     #SW9,@SWR       ;LOOP ON ERROR?
5572  027366  001406                           BEQ     1$              ;BR IF NO
5573  027370  105737  001103                   TSTB    $ERFLG          ;HAD ERROR?
5574  027374  001403                           BEQ     1$              ;BR IF NO
5575  027376  013716  001110                   MOV     $LPERR,(SP)
5576  027402  000002                           RTI
5577
5578  027404  011637  001110          1$:      MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
5579  027410  000002                           RTI
5580
5581
5582                                   ;
5583                                   ;ROUTINE TO INPUT A 'SPACE' OR 'CONTROL-C' FROM TTY
5584                                   ;RETURN IF CONTROL-C
5585                                   ;RETURN +4 IF SPACE
5586                                   ;
5587
5588  027412  005777  151530          CCSP:    TST     @$TKB           ;CLEAR DONE FLAG
5589  027416  104410                           RDCHR                   ;READ CHAR FROM TTY
5590  027420  012600                           MOV     (SP)+,R0        ;GET CHAR OFF STACK
5591  027422  020027  000040                   CMP     R0,#SPBAR       ;SEE IF SPACE
5592  027426  001406                           BEQ     1$              ;BR IF YES.
5593
5594  027430  020027  000003                   CMP     R0,#CONTC       ;SEE IF CONTROL-C
5595  027434  001405                           BEQ     2$              ;BR IF YES
5596  027436  104401  036513                   TYPE    ,MSG31          ;"?"
5597  027442  000763                           BR      CCSP            ;TRY AGAIN
5598
5599  027444  062716  000004          1$:      ADD     #4,(SP)
5600  027450  000207                  2$:      RTS     PC
5601
5602                                   ;
5603                                   ;ROUTINE TO INPUT A 'SPACE' FROM TTY
5604                                   ;
5605  027452  005777  151470          GETSP:   TST     @$TKB           ;CLEAR DONE FLAG
5606  027456  104410                           RDCHR                   ;READ CHAR OFF TTY
5607  027460  012600                           MOV     (SP)+,R0        ;GET CHAR OFF STACK
5608  027462  020027  000040                   CMP     R0,#SPBAR       ;SEE IF SPACE
5609  027466  001403                           BEQ     1$              ;EXIT IF YES
5610  027470  104401  036513                   TYPE    ,MSG31          ;?
5611  027474  000766                           BR      GETSP           ;TRY AGAIN
5612  027476  000207                  1$:      RTS     PC
5613
5614                                   .SBTTL  UNEXPECTED TIMEOUT HANDLER
5615
5616                                   ;
5617                                   ;THIS ROUTINE IS ENTERED IF THERE IS
5618                                   ;       A.  NON EXISTANT MEMORY (NO SSYN)
5619                                   ;       B.  BOUNDRY ERROR
5620                                   ;       C.  STACK OVERFLOW
5621                                   ;
5622
5623  027500  011600                   BADTMO:  MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
5624  027502  005740                           TST     -(R0)           ;GET PC BEFORE UPDATE
5625  027504  032777  020000  151426           BIT     #SW13,@SWR      ;INHIBIT ERR TYPOUT?
5626  027512  001005                           BNE     1$              ;YES, DON'T TYPE
```

```
5627  027514  104401  043327          TYPE    EM3             ;ABORT TESTS,UNEXP T.O. @ PC=
5628  027520  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
5629                                                          ;;TYPE PC
5630  027522  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
5631  027524     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
5632  027525     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
5633  027526  032777  001000  151404  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
5634  027534  001403                  BEQ     2$              ;NO, BRANCH
5635  027536  022626                  CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
5636  027540  000177  151342          JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
5637                                                          ;THAT GAVE BAD TIMEOUT
5638  027544  032777  040000  151366  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
5639  027552  001401                  BEQ     3$              ;NO BRANCH
5640  027554  000002                  RTI                     ;YES
5641
5642  027556  000000                  3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
5643                                                          ;AS INDICATED. YOU CAN LOOP ON
5644                                                          ;ERROR, LOOP ON TEST OR INHIBIT
5645                                                          ;ERROR TYPEOUT BY SETTING THOSE
5646                                                          ;SWITCHES.
5647
5648  027560  022626                  CMP     (SP)+,(SP)+     ;RESTORE STACK
5649  027562  000137  023606          JMP     $EOP            ;ABORT TESTS
5650
5651                                  .SBTTL  UNEXPECTED INTERRUPT HANDLER
5652
5653                                  ;
5654                                  ;THIS ROUTINE CHECKS SW13 (INH ERR TYPOUT), SW9 (LOOP ON ERR)
5655                                  ;& SW14 (LOOP ON TEST).
5656                                  ;
5657
5658  027566  011600                  BADINT: MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED
5659  027570  005740                  TST     -(R0)           ;GET PC BEFORE UPDATE
5660  027572  032777  020000  151340          BIT     #SW13,@SWR      ;INHIBIT ERR TYPEOUT?
5661  027600  001005                  BNE     1$              ;YES, DONT TYPE
5662  027602  104401  043402          TYPE    EM4             ;ABORT TESTS, UNEXP INT @ PC=
5663  027606  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
5664                                                          ;;TYPE PC
5665  027610  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
5666  027612     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
5667  027613     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
5668
5669  027614  032777  001000  151316  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
5670  027622  001403                  BEQ     2$              ;NO, BRANCH
5671  027624  022626                  CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
5672  027626  000177  151254          JMP     @$LPADR         ;GO TO THE STARTING ADDR OF
5673                                                          ;TEST THAT GAVE UNEXP. INT.
5674  027632  032777  040000  151300  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
5675  027640  001401                  BEQ     3$              ;NO, BRANCH
5676  027642  000002                  RTI                     ;YES.
5677
5678  027644  000000                  3$:     HALT                    ;UNEXPECTED INTERRUPT OCCURRED AS
5679                                                          ;INDICATED. YOU CAN LOOP ON ERROR,
5680                                                          ;LOOP ON TEST OR INHIBIT
5681                                                          ;ERROR TYPEOUT BY SETTING THOSE
5682                                                          ;SWITCHES
```

```
                                                        ;
027646  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
027650  000137  023606                  JMP     $EOP            ;ABORT TESTS

                        .SBTTL  MEMORY CHECK ENABLE TRAP

027654  012737  027670  001172  MEMERR: MOV     #1$,$ESCAPE
027662  011637  001350                  MOV     (SP),TRAPPC     ;STORE PC
027666  104202                          ERROR   202             ;UNEXP MEM PARITY ERROR

027670  005037  001172          1$:     CLR     $ESCAPE
027674  032777  001000  151236          BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
027702  001001                          BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
027704  000002                          RTI                     ;ELSE RETURN

027706  012706  001100          2$:     MOV     #STACK,SP       ;INIT STACK
027712  000177  151172                  JMP     @$LPERR

                        .SBTTL  RK06 INTERRUPT HANDLER

027716  000240          INTER:  NOP
027720  000240                  NOP
027722  000240                  NOP
027724  011600                  MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
027726  005740                  TST     -(R0)           ;GET PC BEFORE UPDATE.
027730  104401  035202          TYPE    ,MSG6           ;INT AT PC=
027734  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                        ;;TYPE PC
027736  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
027740     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
027741     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
027742  000000                  HALT
027744  000240                  NOP
027746  000240                  NOP
027750  000002                  RTI

                        .SBTTL  POWER DOWN AND UP ROUTINES

                        ;POWER DOWN ROUTINE

027752  012737  027764  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
027760  000000                  HALT
027762  000776                  BR      .-2             ;HANG UP.

                        ;POWER UP ROUTINE

027764  005037  030036          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
027770  005237  030036          1$:     INC     $PWRCT          ;WAIT FOR THE INCR
027774  001375                  BNE     1$              ;OF WORD
027776  012737  027752  000024          MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
030004  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;PRIORITY 7
030012  012737  000340  000036          MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
030020  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
030024  104401  035376                  TYPE    ,MSG11          ;REPORT POWER FAIL
030030  000005                          RESET
030032  000137  013136                  JMP     PFSRT
```

```
                                ;
030036  000000                  $PWRCT: 0                       ;WAIT COUNT FOR TTY
                                ;
                                ;DIVISION UTILITY ROUTINE
                                ;
                                ;R0-R1-R2-R3=DIVIDEND
                                ;R4-R5=DIVISOR
                                ;R0-R1=REMAINDER AFTER DIVISION
                                ;R2-R3=QUOTIENT AFTER DIVISION
                                ;ENTER WITH JSR PC,M.DPID
                                ;
030040  012746  000040          M.DPID: MOV     #40,-(SP)       ;COUNTER FOR DIVISION CYCLES
030044  010446                          MOV     R4,-(SP)        ;HI ORDER
030046  010546                          MOV     R5,-(SP)        ;LO ORDER TO THE STACK
030050  005466  000002                  NEG     2(SP)           ;FORM NEGATIVE
030054  005416                          NEG     @SP             ;VERSION OF DIVISOR
030056  005666  000002                  SBC     2(SP)
030062  061601                          ADD     @SP,R1
030064  005500                          ADC     R0              ;PERFORM INIT SUBT.
030066  066600  000002                  ADD     2(SP),R0
030072  103445                          BCS     M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
030074  005046                          CLR     -(SP)           ;THIS IS A LONGER LASTING CARRY BIT
030076  006103                  M.DP40: ROL     R3
030100  006102                          ROL     R2
030102  006101                          ROL     R1
030104  006100                          ROL     R0
030106  005716                          TST     @SP             ;TEST CARRY INDICATOR
030110  001410                          BEQ     M.DP41          ;IF TO CARRY THEN ADD, ELSE SUBT.
030112  005016                          CLR     @SP             ;CLEAR UP FOR NEXT TIME
030114  066601  000002                  ADD     2(SP),R1
030120  005500                          ADC     R0              ;ADD -(DIVISOR)
030122  005516                          ADC     @SP             ;SET CARRY
030124  066600  000004                  ADD     4(SP),R0
030130  000404                          BR      M.DP42
                                ;
030132  060501                  M.DP41: ADD     R5,R1
030134  005500                          ADC     R0              ;ADD +(DIVISOR)
030136  005516                          ADC     @SP             ;SET CARRY
030140  060400                          ADD     R4,R0
030142  005516                  M.DP42: ADC     @SP             ;SET CARRY
030144  005716                          TST     @SP             ;TEST THE UPDATE INDICATOR
030146  001401                          BEQ     .+4             ;IF 0,FORGET IT
030150  005203                          INC     R3              ;NO CARRY POSSIBLE HERE
030152  005366  000006                  DEC     6(SP)           ;DECREMENT CTR
030156  003347                          BGT     M.DP40          ;BR IF MORE TO DO
030160  006003                          ROR     R3
030162  103404                          BCS     M.DP44
030164  060501                          ADD     R5,R1
030166  005500                          ADC     R0
030170  060400                          ADD     R4,R0
030172  000241                          CLC
                                ;
030174  006103                  M.DP44: ROL     R3
030176  062706  000010                  ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
030202  000242                          CLV
```

```
 5795  030204  000207                        RTS     PC
 5796
 5797  030206  062706  000006        M.DP50: ADD     #6,SP
 5798  030212  000262                        SEV
 5799  030214  000207                        RTS     PC
 5800
```

```
                        .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

030216                  $SCOPE:
030216  104407                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
030220  032777  040000  150712  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
030226  001114                 BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
030230  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
030232  013746  000004         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
030236  012737  030256  000004 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
030244  005737  177060         TST     @#177060        ;;TIME OUT ON XOR?
030250  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
030254  000463                 BR      $SVLAD          ;;GO TO THE NEXT TEST
030256  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
030260  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
030264  000423                 BR      7$              ;;LOOP ON THE PRESENT TEST
030266                  6$:;#####END OF CODE FOR THE XOR TESTER#####
030266  032777  000400  150644         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
030274  001404                 BEQ     2$              ;;BR IF NO
030276  127737  150636  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
030304  001465                 BEQ     $OVER           ;;BR IF YES
030306  105737  001103  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
030312  001421                 BEQ     3$              ;;BR IF NO
030314  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
030322  101015                 BHI     3$              ;;BR IF NO
030324  032777  001000  150606         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
030332  001404                 BEQ     4$              ;;BR IF NO
030334  013737  001110  001106 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
030342  000446                 BR      $OVER
030344  105037  001103  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
030350  005037  001170         CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
030354  000415                 BR      1$              ;;ESCAPE TO THE NEXT TEST
030356  032777  004000  150554 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
030364  001011                 BNE     1$              ;;BR IF YES
030366  005737  001212         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
030372  001406                 BEQ     1$              ;;      INHIBIT ITERATIONS
030374  005237  001104         INC     $ICNT           ;;INCREMENT ITERATION COUNT
030400  023737  001170  001104 CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
030406  002024                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
030410  012737  000001  001104 1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
030416  013737  030474  001170         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
030424  105237  001102  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
030430  113737  001102  001210         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 5857  030436  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 5858  030442  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 5859  030446  005037  001172                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 5860  030452  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 5861  030460  013777  001102  150454  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 5862  030466  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 5863  030472  000002                          RTI                     ;;FIXES PS
 5864  030474  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 5865                                  .SBTTL  ERROR HANDLER ROUTINE
 5866
 5867                                  ;;**************************************************************
 5868                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 5869                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 5870                                  ;*AND GO TO TYPERR ON ERROR
 5871                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 5872                                  ;*SW15=1         HALT ON ERROR
 5873                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
 5874                                  ;*SW10=1         BELL ON ERROR
 5875                                  ;*SW09=1         LOOP ON ERROR
 5876                                  ;*CALL
 5877                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 5878
 5879  030476                          $ERROR:
 5880  030476  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 5881  030500  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 5882  030504  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 5883  030506  013777  001102  150426          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 5884  030514  032777  002000  150416          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 5885  030522  001402                          BEQ     1$              ;;NO - SKIP
 5886  030524  104401  001174                  TYPE    ,$BELL          ;;RING BELL
 5887  030530  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 5888  030534  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 5889  030540  162737  000002  001116          SUB     #2,$ERRPC
 5890  030546  117737  150344  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 5891  030554  032777  020000  150356          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 5892  030562  001004                          BNE     20$             ;;SKIP TYPEOUTS
 5893  030564  004737  055200                  JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 5894  030570  104401  001201                  TYPE    ,$CRLF
 5895  030574                          20$:
 5896  030574  122737  000001  001224          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 5897  030602  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
 5898  030604  113737  001114  030616          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 5899  030612  004737  031422                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 5900  030616     000                  21$:    .BYTE   0
 5901  030617     000                          .BYTE   0
 5902  030620  000777                  22$:    BR      22$             ;;APT ERROR LOOP
 5903  030622  005777  150312          2$:     TST     @SWR            ;;HALT ON ERROR
 5904  030626  100002                          BPL     3$              ;;SKIP IF CONTINUE
 5905  030630  000000                          HALT                    ;;HALT ON ERROR!
 5906  030632  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 5907  030634  032777  001000  150276  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 5908  030642  001402                          BEQ     4$              ;;BR IF NO
 5909  030644  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 5910  030650  005737  001172          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 5911  030654  001402                          BEQ     5$              ;;BR IF NONE
 5912  030656  013716  001172                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
5913  030662
5914  030662  022737  023674  000042  5$:     CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
5915  030670  001001                          BNE     6$              ;;BRANCH IF NO
5916  030672  000000                          HALT                    ;;YES
5917  030674                              6$:
5918  030674  000002                          RTI                     ;;RETURN
5919                                    .SBTTL  TYPE ROUTINE
5920
5921                                    ;;**************************************************************
5922                                    ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
5923                                    ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
5924                                    ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
5925                                    ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
5926                                    ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
5927                                    ;*
5928                                    ;*CALL:
5929                                    ;*1) USING A TRAP INSTRUCTION
5930                                    ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
5931                                    ;*OR
5932                                    ;*      TYPE
5933                                    ;*      MESADR
5934                                    ;*
5935
5936  030676  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
5937  030702  100002                          BPL     1$              ;;BR IF YES
5938  030704  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
5939  030706  000430                          BR      3$              ;;LEAVE
5940  030710  010046                      1$: MOV     R0,-(SP)        ;;SAVE R0
5941  030712  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
5942  030716  122737  000001  001224        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
5943  030724  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
5944  030726  132737  000100  001225        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
5945  030734  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
5946  030736  010037  030746                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
5947  030742  004737  031412                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
5948  030746  000000                      61$:.WORD   0               ;;MESSAGE ADDRESS
5949  030750  132737  000040  001225  62$:  BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
5950  030756  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
5951  030760  112046                      2$: MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
5952  030762  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
5953  030764  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
5954  030766  012600                      60$:MOV     (SP)+,R0        ;;RESTORE R0
5955  030770  062716  000002                3$: ADD     #2,(SP)         ;;ADJUST RETURN PC
5956  030774  000002                          RTI                     ;;RETURN
5957  030776  122716  000011                4$: CMPB    #HT,(SP)        ;;BRANCH IF <HT>
5958  031002  001430                          BEQ     8$
5959  031004  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
5960  031010  001006                          BNE     5$
5961  031012  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
5962  031014  104401                          TYPE                    ;;TYPE A CR AND LF
5963  031016  001201                          $CRLF
5964  031020  105037  031154                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
5965  031024  000755                          BR      2$              ;;GET NEXT CHARACTER
5966  031026  004737  031110                5$: JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
5967  031032  123726  001156                6$: CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
5968  031036  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
 5969  031040  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 5970                                                                  ;;AND THE NULL CHAR.
 5971  031044  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 5972  031050  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 5973  031052  004737  031110                JSR     PC,$TYPEC       ;;GO TYPE A NULL
 5974  031056  105337  031154                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 5975  031062  000770                        BR      7$              ;;LOOP
 5976
 5977                                  ;HORIZONTAL TAB PROCESSOR
 5978
 5979  031064  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 5980  031070  004737  031110        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 5981  031074  132737  000007  031154        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 5982  031102  001372                        BNE     9$              ;;TAB STOP
 5983  031104  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 5984  031106  000724                        BR      2$              ;;GET NEXT CHARACTER
 5985  031110  105777  150034        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 5986  031114  100375                        BPL     $TYPEC
 5987  031116  116677  000002  150026        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 5988  031124  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 5989  031132  001003                        BNE     1$              ;;BRANCH IF NO
 5990  031134  105037  031154                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 5991  031140  000406                        BR      $TYPEX          ;;EXIT
 5992  031142  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 5993  031150  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 5994  031152  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 5995  031154  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
 5996  031156  000207        $TYPEX: RTS     PC
 5997
 5998                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 5999
 6000                                  ;;**************************************************************
 6001                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 6002                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 6003                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 6004                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 6005                                  ;*REPLACED WITH SPACES.
 6006                                  ;*CALL:
 6007                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 6008                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
 6009
 6010  031160                          $TYPDS:
 6011  031160  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6012  031162  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 6013  031164  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 6014  031166  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 6015  031170  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 6016  031172  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 6017  031176  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 6018  031202  100004                        BPL     1$              ;;BR IF INPUT IS POS.
 6019  031204  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 6020  031206  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 6021  031214  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 6022  031216  012703  031374                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 6023  031222  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 6024  031226  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
```

```
 6025  031230  016001  031364            MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 6026  031234  160105            3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 6027  031236  002402                    BLT     4$              ;;BR IF DONE
 6028  031240  005202                    INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 6029  031242  000774                    BR      3$
 6030  031244  060105            4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 6031  031246  005702                    TST     R2              ;;CHECK IF BCD DIGIT=0
 6032  031250  001002                    BNE     5$              ;;FALL THROUGH IF 0
 6033  031252  105716                    TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 6034  031254  100407                    BMI     7$              ;;BR IF YES
 6035  031256  106316            5$:     ASLB    (SP)            ;;MSD?
 6036  031260  103003                    BCC     6$              ;;BR IF NO
 6037  031262  116663  000001  177777    MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 6038  031270  052702  000060      6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 6039  031274  052702  000040      7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 6040  031300  110223                    MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 6041  031302  005720                    TST     (R0)+           ;;JUST INCREMENTING
 6042  031304  020027  000010            CMP     R0,#10          ;;CHECK THE TABLE INDEX
 6043  031310  002746                    BLT     2$              ;;GO DO THE NEXT DIGIT
 6044  031312  003002                    BGT     8$              ;;GO TO EXIT
 6045  031314  010502                    MOV     R5,R2           ;;GET THE LSD
 6046  031316  000764                    BR      6$              ;;GO CHANGE TO ASCII
 6047  031320  105726            8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 6048  031322  100003                    BPL     9$              ;;BR IF NO
 6049  031324  116663  177777  177776    MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 6050  031332  105013            9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 6051  031334  012605                    MOV     (SP)+,R5        ;;POP STACK INTO R5
 6052  031336  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
 6053  031340  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
 6054  031342  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
 6055  031344  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
 6056  031346  104401  031374            TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 6057  031352  016666  000002  000004    MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 6058  031360  012616                    MOV     (SP)+,(SP)
 6059  031362  000002                    RTI                     ;;RETURN TO USER
 6060  031364  023420            $DTBL:  10000.
 6061  031366  001750                    1000.
 6062  031370  000144                    100.
 6063  031372  000012                    10.
 6064  031374  000004            $DBLK:  .BLKW   4
 6065                              .SBTTL  APT COMMUNICATIONS ROUTINE
 6066
 6067                              ;;**************************************************************
 6068  031404  112737  000001  031650  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 6069  031412  112737  000001  031646  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 6070  031420  000403                    BR      $ATYC
 6071  031422  112737  000001  031650  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 6072  031430                            $ATYC:
 6073  031430  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6074  031432  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 6075  031434  105737  031646            TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 6076  031440  001450                    BEQ     5$              ;;IF NOT:  BR
 6077  031442  122737  000001  001224    CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 6078  031450  001031                    BNE     3$              ;;IF NOT:  BR
 6079  031452  132737  000100  001225    BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 6080  031460  001425                    BEQ     3$              ;;IF NOT:  BR
```

```
6081  031462  017600  000004              MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
6082  031466  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
6083  031474  005737  001204          1$: TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
6084  031500  001375                      BNE     1$              ;;IF NOT:  WAIT
6085  031502  010037  001220              MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
6086  031506  105720                  2$: TSTB    (R0)+           ;;FIND END OF MESSAGE
6087  031510  001376                      BNE     2$
6088  031512  163700  001220              SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
6089  031516  006200                      ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
6090  031520  010037  001222              MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
6091  031524  012737  000004  001204      MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
6092  031532  000413                      BR      5$
6093  031534  017637  000004  031560  3$: MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
6094  031542  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
6095  031550  013746  177776              MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
6096  031554  004737  030676              JSR     PC,$TYPE        ;;CALL TYPE MACRO
6097  031560  000000                  4$: .WORD   0
6098  031562                          5$:
6099  031562  105737  031650          10$: TSTB   $FFLG           ;;SHOULD REPORT FATAL ERROR?
6100  031566  001416                      BEQ     12$             ;;IF NOT:  BR
6101  031570  005737  001224              TST     $ENV            ;;RUNNING UNDER APT?
6102  031574  001413                      BEQ     12$             ;;IF NOT:  BR
6103  031576  005737  001204          11$: TST    $MSGTYPE        ;;FINISHED LAST MESSAGE?
6104  031602  001375                      BNE     11$             ;;IF NOT:  WAIT
6105  031604  017637  000004  001206      MOV     @4(SP),$FATAL   ;;GET ERROR #
6106  031612  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
6107  031620  005237  001204              INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
6108  031624  105037  031650          12$: CLRB   $FFLG           ;;CLEAR FATAL FLAG
6109  031630  105037  031647              CLRB    $LFLG           ;;CLEAR LOG FLAG
6110  031634  105037  031646              CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
6111  031640  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
6112  031642  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
6113  031644  000207                      RTS     PC              ;;RETURN
6114  031646     000              $MFLG:  .BYTE   0               ;;MESSG. FLAG
6115  031647     000              $LFLG:  .BYTE   0               ;;LOG FLAG
6116  031650     000              $FFLG:  .BYTE   0               ;;FATAL FLAG
6117          031652                      .EVEN
6118          000200              APTSIZE=200
6119          000001              APTENV=001
6120          000100              APTSPOOL=100
6121          000040              APTCSUP=040
6122                              .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
6123
6124                              ;;**************************************************************
6125                              ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
6126                              ;*OCTAL (ASCII) NUMBER AND TYPE IT.
6127                              ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
6128                              ;*CALL:
6129                              ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
6130                              ;*      TYPOS                   ;;CALL FOR TYPEOUT
6131                              ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
6132                              ;*      .BYTE   M               ;;M=1 OR 0
6133                              ;*                                      ;;1=TYPE LEADING ZEROS
6134                              ;*                                      ;;0=SUPPRESS LEADING ZEROS
6135                              ;*
6136                              ;*$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

031652  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
031656  116637  000001  032075          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
031664  112637  032077                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
031670  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
031674  000406                          BR      $TYPON
031676  112737  000001  032075  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
031704  112737  000006  032077          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
031712  112737  000005  032074  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
031720  010346                          MOV     R3,-(SP)        ;;SAVE R3
031722  010446                          MOV     R4,-(SP)        ;;SAVE R4
031724  010546                          MOV     R5,-(SP)        ;;SAVE R5
031726  113704  032077                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
031732  005404                          NEG     R4
031734  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
031740  110437  032076                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
031744  113704  032075                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
031750  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
031754  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
031756  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
031760  000404                          BR      3$              ;;GO DO MSB
031762  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
031764  006105                          ROL     R5
031766  006105                          ROL     R5
031770  010503                          MOV     R5,R3
031772  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
031774  105337  032076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
032000  100016                          BPL     7$              ;;BR IF NO
032002  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
032006  001002                          BNE     4$              ;;TEST FOR 0
032010  005704                          TST     R4              ;;SUPPRESS THIS 0?
032012  001403                          BEQ     5$              ;;BR IF YES
032014  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
032016  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
032022  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
032026  110337  032072                  MOVB    R3,8$           ;;SAVE FOR TYPING
032032  104401  032072                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
032036  105337  032074          7$:     DECB    $OCNT           ;;COUNT BY 1
032042  003347                          BGT     2$              ;;BR IF MORE TO DO
032044  002402                          BLT     6$              ;;BR IF DONE
032046  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
032050  000744                          BR      2$              ;;GO DO THE LAST DIGIT
032052  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
032054  012604                          MOV     (SP)+,R4        ;;RESTORE R4
032056  012603                          MOV     (SP)+,R3        ;;RESTORE R3
032060  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
032066  012616                          MOV     (SP)+,(SP)
```

```
                                        RTI                             ;;RETURN
032070  000002                  B$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
032072  000                             .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
032073  000                     $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
032074  000                     $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
032075  000                     $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
032076  000000                  .SBTTL  TTY INPUT ROUTINE

                                ;;**********************************************************************
                                .ENABL  LSB

                                ;;**********************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                ;*WHEN OPERATING IN TTY FLAG MODE.
032100  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR              ;;IS THE SOFT-SWR SELECTED?
032106  001074                         BNE     15$                     ;;BRANCH IF NO
032110  105777  147030                 TSTB    @$TKS                   ;;CHAR THERE?
032114  100071                         BPL     15$                     ;;IF NO, DON'T WAIT AROUND
032116  117746  147024                 MOVB    @$TKB,-(SP)             ;;SAVE THE CHAR
032122  042716  177600                 BIC     #^C177,(SP)             ;;STRIP-OFF THE ASCII
032126  022726  000007                 CMP     #7,(SP)+                ;;IS IT A CONTROL G?
032132  001062                         BNE     15$                     ;;NO, RETURN TO USER
032134  123727  001134  000001         CMPB    $AUTOB,#1               ;;ARE WE RUNNING IN AUTO-MODE?
032142  001456                         BEQ     15$                     ;;BRANCH IF YES

032144  104401  032765                 TYPE    ,$CNTLG                 ;;ECHO THE CONTROL-G (^G)
032150  104401  032772         $GTSWR: TYPE    ,$MSWR                  ;;TYPE CURRENT CONTENTS
032154  013746  000176                 MOV     SWREG,-(SP)             ;;SAVE SWREG FOR TYPEOUT
032160  104402                         TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
032162  104401  033003                 TYPE    ,$MNEW                  ;;PROMPT FOR NEW SWR
032166  005046                 19$:    CLR     -(SP)                   ;;CLEAR COUNTER
032170  005046                         CLR     -(SP)                   ;;THE NEW SWR
032172  105777  146746         7$:     TSTB    @$TKS                   ;;CHAR THERE?
032176  100375                         BPL     7$                      ;;IF NOT TRY AGAIN

032200  117746  146742                 MOVB    @$TKB,-(SP)             ;;PICK UP CHAR
032204  042716  177600                 BIC     #^C177,(SP)             ;;MAKE IT 7-BIT ASCII


032210  021627  000025         9$:     CMP     (SP),#25                ;;IS IT A CONTROL-U?
032214  001005                         BNE     10$                     ;;BRANCH IF NOT
032216  104401  032760                 TYPE    ,$CNTLU                 ;;YES, ECHO CONTROL-U (^U)
032222  062706  000006         20$:    ADD     #6,SP                   ;;IGNORE PREVIOUS INPUT
032226  000757                         BR      19$                     ;;LET'S TRY IT AGAIN

032230  021627  000015         10$:    CMP     (SP),#15                ;;IS IT A <CR>?
032234  001022                         BNE     16$                     ;;BRANCH IF NO
032236  005766  000004                 TST     4(SP)                   ;;YES, IS IT THE FIRST CHAR?
032242  001403                         BEQ     11$                     ;;BRANCH IF YES
032244  016677  000002  146666         MOV     2(SP),@SWR              ;;SAVE NEW SWR
032252  062706  000006         11$:    ADD     #6,SP                   ;;CLEAR UP STACK
032256  104401  001201         14$:    TYPE    ,$CRLF                  ;;ECHO <CR> AND <LF>
```

```
                                        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
                                        BNE     15$             ;;BRANCH IF NOT
                                        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
                                15$:    RTI                     ;;RETURN
                                16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
                                        CMP     (SP),#60        ;;CHAR < 0?
                                        BLT     18$             ;;BRANCH IF YES
                                        CMP     (SP),#67        ;;CHAR > 7?
                                        BGT     18$             ;;BRANCH IF YES
                                        BIC     #60,(SP)+       ;;STRIP-OFF ASCII
                                        TST     2(SP)           ;;IS THIS THE FIRST CHAR
                                        BEQ     17$             ;;BRANCH IF YES
                                        ASL     (SP)            ;;NO, SHIFT PRESENT
                                        ASL     (SP)            ;;   CHAR OVER TO MAKE
                                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
                                17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
                                        BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
                                        BR      7$              ;;GET THE NEXT ONE
                                18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
                                        BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB

                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
                                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
                                        MOV     4(SP),2(SP)     ;;SAVE THE PS
                                1$:     TSTB    @$TKS           ;;WAIT FOR
                                        BPL     1$              ;;A CHARACTER
                                        MOVB    @$TKB,4(SP)     ;;READ THE TTY
                                        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
                                        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
                                        BNE     3$              ;;BRANCH IF NO
                                2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
                                        BPL     2$              ;;LOOP UNTIL ITS THERE
                                        MOVB    @$TKB,-(SP)     ;;GET CHARACTER
                                        BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
                                        BNE     2$              ;;IF NOT DISCARD IT
                                        BR      1$              ;;YES, RESUME
                                3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
                                        BLT     4$              ;;BRANCH IF YES
                                        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
                                        BGT     4$              ;;BRANCH IF YES
                                        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
                                4$:     RTI                     ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*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

032502  010346                  $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
032504  005046                          CLR     -(SP)                   ;;CLEAR THE RUBOUT KEY
032506  012703  032736            1$:   MOV     #$TTYIN,R3              ;;GET ADDRESS
032512  022703  032760            2$:   CMP     #$TTYIN+22,R3           ;;BUFFER FULL?
032516  101456                          BLOS    4$                      ;;BR IF YES
032520  104410                          RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
032522  112613                          MOVB    (SP)+,(R3)              ;;GET CHARACTER
032524  122713  000177           10$:   CMPB    #177,(R3)               ;;IS IT A RUBOUT
032530  001022                          BNE     5$                      ;;BR IF NO
032532  005716                          TST     (SP)                    ;;IS THIS THE FIRST RUBOUT?
032534  001007                          BNE     6$                      ;;BR IF NO
032536  112737  000134  032734          MOVB    #'\,9$                  ;;TYPE A BACK SLASH
032544  104401  032734                  TYPE    ,9$
032550  012716  177777                  MOV     #-1,(SP)                ;;SET THE RUBOUT KEY
032554  005303                  6$:     DEC     R3                      ;;BACKUP BY ONE
032556  020327  032736                  CMP     R3,#$TTYIN              ;;STACK EMPTY?
032562  103434                          BLO     4$                      ;;BR IF YES
032564  111337  032734                  MOVB    (R3),9$                 ;;SETUP TO TYPEOUT THE DELETED CHAR.
032570  104401  032734                  TYPE    ,9$                     ;;GO TYPE
032574  000746                          BR      2$                      ;;GO READ ANOTHER CHAR.
032576  005716                  5$:     TST     (SP)                    ;;RUBOUT KEY SET?
032600  001406                          BEQ     7$                      ;;BR IF NO
032602  112737  000134  032734          MOVB    #'\,9$                  ;;TYPE A BACK SLASH
032610  104401  032734                  TYPE    ,9$
032614  005016                          CLR     (SP)                    ;;CLEAR THE RUBOUT KEY
032616  122713  000025            7$:   CMPB    #25,(R3)                ;;IS CHARACTER A CTRL U?
032622  001003                          BNE     8$                      ;;BR IF NO
032624  104401  032760                  TYPE    ,$CNTLU                 ;;TYPE A CONTROL "U"
032630  000726                          BR      1$                      ;;GO START OVER
032632  122713  000022            8$:   CMPB    #22,(R3)                ;;IS CHARACTER A "↑R"?
032636  001011                          BNE     3$                      ;;BRANCH IF NO
032640  105013                          CLRB    (R3)                    ;;CLEAR THE CHARACTER
032642  104401  001201                  TYPE    ,$CRLF                  ;;TYPE A "CR" & "LF"
032646  104401  032736                  TYPE    ,$TTYIN                 ;;TYPE THE INPUT STRING
032652  000717                          BR      2$                      ;;GO PICKUP ANOTHER CHACTER
032654  104401  001200            4$:   TYPE    ,$QUES                  ;;TYPE A '?'
032660  000712                          BR      1$                      ;;CLEAR THE BUFFER AND LOOP
032662  111337  032734            3$:   MOVB    (R3),9$                 ;;ECHO THE CHARACTER
032666  104401  032734                  TYPE    ,9$
032672  122723  000015                  CMPB    #15,(R3)+               ;;CHECK FOR RETURN
032676  001305                          BNE     2$                      ;;LOOP IF NOT RETURN
032700  105063  177777                  CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
032704  104401  001202                  TYPE    ,$LF                    ;;TYPE A LINE FEED
032710  005726                          TST     (SP)+                   ;;CLEAN RUBOUT KEY FROM THE STACK
032712  012603                          MOV     (SP)+,R3                ;;RESTORE R3
032714  011646                          MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
032716  016666  000004  000002          MOV     4(SP),2(SP)             ;;      FIRST ASCII CHARACTER ON IT
032724  012766  032736  000004          MOV     #$TTYIN,4(SP)
032732  000002                          RTI                             ;;RETURN
032734     000                  9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
032735     000                          .BYTE   0                       ;;TERMINATOR
032736  000022              $TTYIN:     .BLKB   22                      ;;RESERVE 22 BYTES FOR TTY INPUT
032760  052536  005015  000 $CNTLU:     .ASCIZ  /↑U/<15><12>            ;;CONTROL "U"
```

```
6361  032765  136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
6362  032772  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
6363  033000  020075  000
6364  033003  040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
6365  033010  036440  000040
6366                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
6367
6368                                  ;;*****************************************************************
6369                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
6370                                  ;*CHANGE IT TO BINARY.
6371                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
6372                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
6373                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
6374                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
6375                                  ;*CALL:
6376                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
6377                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
6378                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
6379
6380  033014  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
6381  033016  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
6382  033024  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6383  033026  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6384  033030  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6385  033032  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
6386  033034  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
6387  033036  010037  033142          MOV     R0,5$           ;;AND SAVE IT
6388  033042  005001                  CLR     R1              ;;CLEAR DATA WORD
6389  033044  005002                  CLR     R2
6390  033046  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
6391  033050  001420                  BEQ     3$              ;;IF ZERO GET OUT
6392  033052  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
6393  033056  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
6394  033060  122716  000067          CMPB    #'7,(SP)
6395  033064  002423                  BLT     4$
6396  033066  006301                  ASL     R1              ;;*2
6397  033070  006102                  ROL     R2
6398  033072  006301                  ASL     R1              ;;*4
6399  033074  006102                  ROL     R2
6400  033076  006301                  ASL     R1              ;;*8
6401  033100  006102                  ROL     R2
6402  033102  042716  177770          BIC     #↑C7,(SP)       ;;STRIP THE ASCII JUNK
6403  033106  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
6404  033110  000756                  BR      2$              ;;LOOP
6405  033112  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
6406  033114  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
6407  033120  010237  033152          MOV     R2,$HIOCT
6408  033124  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
6409  033126  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
6410  033130  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
6411  033132  000002                  RTI                     ;;RETURN
6412  033134  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
6413  033136  105010                  CLRB    (R0)            ;;SET A TERMINATOR
6414  033140  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
6415  033142  000000          5$:     .WORD   0
6416  033144  104401  001200          TYPE    ,$QUES          ;;"?" "CR" & "LF"
```

```
                        BR      1$                      ;;TRY AGAIN
$HIOCT: .WORD   0                       ;;HIGH ORDER BITS GO HERE
.SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE


;;**********************************************************************
;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
;*UNSIGNED OCTAL ASCIZ NUMBER.
;*CALL
;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
;*      JSR     PC,@#$DB2O      ;;CALL THE ROUTINE
;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK


$DB2O:  SAVREG                  ;;SAVE ALL REGISTERS
        MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
        MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
        MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
        MOV     #^C7,R3         ;;MASK
        MOV     (R1)+,R0        ;;LOWER WORD
        MOV     (R1)+,R1        ;;HIGH WORD
        CLR     R2              ;;TERMINATOR
1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
        MOV     R0,R2           ;;GET THIS DIGIT
        DEC     R4              ;;COUNT THIS CHARACTER
        BGT     3$              ;;BR IF NOT THE LAST DIGIT
        BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
        INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
        MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
        RESREG                  ;;RESTORE ALL REGISTERS
        RTS     PC              ;;RETURN TO USER
2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
        ROR     R0              ;;      THE NEXT OCTAL DIGIT
        ROR     R1
        ROR     R0
        ROR     R1
        ROR     R0
        BIC     R3,R2           ;;MASK OUT ALL JUNK
        ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
        BR      1$              ;;GO PUT IT IN THE DATA TABLE
$OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
.SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE


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


$DB2D:  SAVREG                  ;;SAVE REGISTERS
        MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
```

```
                                        MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
                                        MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
                                        MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
                                        MOV     (R2)+,R2
                                        MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
                                        MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
                                        MOV     #$TNPWR+2,R5
                                1$:     CLR     R3              ;;CLEAR PARTIAL
                                2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
                                        SBC     R2
                                        SUB     (R5),R2
                                        BLT     3$              ;;BR IF TEN POWER TO LARGE
                                        INC     R3              ;;ADD 1 TO PARTIAL
                                        BR      2$              ;;LOOP
                                3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
                                        ADC     R2
                                        ADD     (R4)+,R2
                                        CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
                                        BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
                                        MOVB    R3,(R0)+        ;;SAVE IT
                                        DEC     (PC)+           ;;DONE?
                                4$:     .WORD   0
                                        BNE     1$              ;;BR IF NO
                                        CLRB    (R0)+           ;;TERMINATOR
                                        RESREG                  ;;RESTORE REGISTERS
                                        RTS     PC              ;;RETURN
                                $TNPWR: 145000                  ;;1.0E09
                                        35632
                                        160400                  ;;1.0E08
                                        2765
                                        113200                  ;;1.0E07
                                        230
                                        041100                  ;;1.0E06
                                        17
                                        103240                  ;;1.0E05
                                        1
                                        23420                   ;;1.0E04
                                        0
                                        1750                    ;;1.0E03
                                        0
                                        144                     ;;1.0E02
                                        0
                                        12                      ;;1.0E01
                                        0
                                        1                       ;;1.0E00
                                        0
                                $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
                                .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE

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

```
                                        $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
                                                MOV     #1$,-(SP)       ;;SET POINTER
                                                JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
                                                ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
                                                MOV     (SP)+,2(SP)     ;;PICKUP POINTER
                                                RTS     PC              ;;RETURN
                                        1$:     .WORD   0,0
                                        .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS

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

                                        $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
                                                MOV     4(SP),R0        ;;PICKUP THE POINTER
                                        1$:     TSTB    (R0)            ;;TERMINATEOR?
                                                BEQ     2$              ;;BR IF YES
                                                CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
                                                BEQ     1$              ;;BR IF YES
                                        2$:     DEC     R0              ;;BACKUP BY "1"
                                                MOV     R0,3$           ;;SAVE FOR TYPING
                                                TYPE                    ;;GO TYPE
                                        3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
                                                MOV     (SP)+,R0        ;;RESTORE R0
                                                MOV     (SP)+,(SP)      ;;RESTORE THE STACK
                                                RTS     PC              ;;RETURN
                                        .SBTTL  INTEGER MULTIPLY ROUTINE

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

                                        $MULT:
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                CLR     -(SP)           ;;CLEAR THE SIGN KEY
                                                MOV     12(SP),R1       ;;GET THE MULTIPLICAND
                                                BPL     1$              ;;BR IF PLUS
                                                INC     (SP)            ;;SET THE SIGN KEY
                                                NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
                                        1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
```

```
                        BPL     2$              ;;BR IF PLUS
                        DEC     (SP)            ;;UPDATE THE SIGN KEY
                        NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
033620  012746  000021  2$:     MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
033624  005000          CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
033626  103001  3$:     BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
033630  060200          ADD     R2,R0
033632  006000  4$:     ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
033634  006001          ROR     R1              ;;THE MULTIPLICAND
033636  005316          DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
033640  001372          BNE     3$              ;;BR IF NO
033642  022616          CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
033644  001403          BEQ     5$              ;;GO TO EXIT IF NO
033646  005400          NEG     R0              ;;YES--SO MAKE IT SO
033650  005401          NEG     R1
033652  005600          SBC     R0
033654  005726  5$:     TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
033656  010066  000012  MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
033662  010166  000010  MOV     R1,10(SP)       ;;LSB'S
033666  012602          MOV     (SP)+,R2        ;;POP STACK INTO R2
033670  012601          MOV     (SP)+,R1        ;;POP STACK INTO R1
033672  012600          MOV     (SP)+,R0        ;;POP STACK INTO R0
033674  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

033676          $SAVREG:
033676  010046          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
033700  010146          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
033702  010246          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
033704  010346          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
033706  010446          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
033710  010546          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
033712  016646  000022  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
033716  016646  000022  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
033722  016646  000022  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
033726  016646  000022  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
033732  000002          RTI

                ;*RESTORE R0-R5
                ;*CALL:
                ;*      RESREG
```

```
                                ;RESREG:
                                        MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
                                        MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
                                        MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
                                        MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
                                        MOV     (SP)+,R5        ;;POP STACK INTO R5
                                        MOV     (SP)+,R4        ;;POP STACK INTO R4
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        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.

                                ;TRAP:   MOV     R0,-(SP)        ;;SAVE R0
                                        MOV     2(SP),R0        ;;GET TRAP ADDRESS
                                        TST     -(R0)           ;;BACKUP BY 2
                                        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
                                        ASL     R0              ;;POSITION FOR INDEXING
                                        MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
                                        RTS     R0              ;;GO TO ROUTINE


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

                                ;TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
                                        MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
                                        RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
                                ;TRPAD: .WORD   $TRAP2
                                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
                                        $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
                                        $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
                                        $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
                                        $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

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

                                        $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
                                        $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
                                        $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
                                        $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
                                        $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
```

```
  6697  034056  033734                  $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
  6698  034060  027360                  SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
  6699
```

```
                                .SBTTL  SERVICE MESSAGES

034062  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
034070  051525  051040  030113
034076  020066  051104  053111
034104  020105  044504  043501
034112  047516  052123  041511
034120  005015  050011  051101          .ASCII  <CR><LF>/       PART 3/
034126  020124     063
034131     015  046412  044501          .ASCII  <CR><LF>/MAINDEC-11-DZR6J-B-PB/<CR><LF>
034136  042116  041505  030455
034144  026461  055104  033122
034152  026512  026502  041120
034160  005015
034162  005015  025011  025052          .ASCII  <CR><LF>/       ***** CAUTION *****/<CR><LF>
034170  025052  041440  052501
034176  044524  047117  025040
034204  025052  025052  005015
034212  005015  044124  051511          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY AT THE END/
034220  050040  047522  051107
034226  046501  051440  047510
034234  046125  020104  041040
034242  020105  040510  052114
034250  042105  047440  046116
034256  020131  052101  052040
034264  042510  042440  042116
034272  005015  043117  040440          .ASCII  <CR><LF>/OF A PASS, OTHERWISE HEADERS WRITTEN ON THE/
034300  050040  051501  026123
034306  047440  044124  051105
034314  044527  042523  044040
034322  040505  042504  051522
034330  053440  044522  052124
034336  047105  047440  020116
034344  044124     105
034347     015  042012  051511          .ASCII  <CR><LF>/DISK CARTRIDGE MAY BE LEFT IN AN UNFORMATTED STATE/<CR><LF>
034354  020113  040503  052122
034362  044522  043504  020105
034370  040515  020131  042502
034376  046040  043105  020124
034404  047111  040440  020116
034412  047125  047506  046522
034420  052101  042524  020104
034426  052123  052101  006505
034434     012
034435     015  040412  051514          .ASCII  <CR><LF>/ALSO, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
034442  026117  042040  044522
034450  042526  020123  047524
034456  041040  020105  042524
034464  052123  042105  051440
034472  047510  046125  020104
034500  040510  042526  006472
034506     012
034507     015  040412  020056          .ASCII  <CR><LF>/A. HEADS MANUALLY LOADED/
034514  042510  042101  020123
```

```
6756  034522  040515  052516  046101
6757  034530  054514  046040  040517
6758  034536  042504     104
6759  034541     015  041012  020056          .ASCII  <CR><LF>/B. CORRECT PORT SELECTED/
6760  034546  047503  051122  041505
6761  034554  020124  047520  052122
6762  034562  051440  046105  041505
6763  034570  042524     104
6764  034573     015  041412  020056          .ASCII  <CR><LF>/C. WRITE LOCK DISABLED/
6765  034600  051127  052111  020105
6766  034606  047514  045503  042040
6767  034614  051511  041101  042514
6768  034622     104
6769  034623     015  042012  020056          .ASCII  <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
6770  034630  051104  053111  020105
6771  034636  042522  042'01  020131
6772  034644  047111  044504  040503
6773  034652  047524  020122  047117
6774  034660  005015
6775  034662  005015  051104  053111          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
6776  034670  051505  047040  052117
6777  034676  052040  020117  042502
6778  034704  052040  051505  042524
6779  034712  020104  052515  052123
6780  034720  044040  053101     105
6781  034725     015  041012  052117          .ASCIZ  <CR><LF>/BOTH PORTS DESELECTED/
6782  034732  020110  047520  052122
6783  034740  020123  042504  042523
6784  034746  042514  052103  042105
6785  034754     000
6786
6787  034755     015  041012  020105  MSG2:   .ASCIZ  <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
6788  034762  052523  042522  052040
6789  034770  020117  052520  020124
6790  034776  041523  040522  041524
6791  035004  020110  040520  045503
6792  035012  044440  020116  051104
6793  035020  053111  020105  000060
6794  035026  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
6795  035034  024105  024523  052040
6796  035042  020117  042502  052040
6797  035050  051505  042524  035104
6798  035056  000040
6799  035060  005015  054524  042520  MSG4:   .ASCIZ  <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440 /
6800  035066  041040  051525  020123
6801  035074  042101  051104  051505
6802  035102  020123  043111  047040
6803  035110  052117  030440  033467
6804  035116  032064  020060     000
6805  035123     015  052012  050131  MSG5:   .ASCIZ  <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210 /
6806  035130  020105  047503  052116
6807  035136  047522  046114  051105
6808  035144  044440  052116  051105
6809  035152  052522  052120  053040
6810  035160  041505  047524  020122
6811  035166  043111  047040  052117
```

```
6812  035174  031040  030061  000040
6813  035202  005015  047111  042524  MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
6814  035210  051122  050125  020124
6815  035216  041517  052503  051122
6816  035224  042105  040440  020124
6817  035232  041520  000075
6818  035236  005015  051104  053111  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
6819  035244  020105  020060  044527
6820  035252  046114  047040  052117
6821  035260  041040  020105  042524
6822  035266  052123  042105     000
6823  035273     015  044412  052116  MSG8:   .ASCIZ  <CR><LF>/INTERLOCKS TEST/<CR><LF>
6824  035300  051105  047514  045503
6825  035306  020123  042524  052123
6826  035314  005015     000
6827  035317     015  051012  046505  MSG9:   .ASCIZ  <CR><LF>/REMOVE UNIT SELECT PLUG/
6828  035324  053117  020105  047125
6829  035332  052111  051440  046105
6830  035340  041505  020124  046120
6831  035346  043525     000
6832  035351     015  005012  044527  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
6833  035356  046114  052040  051505
6834  035364  020124  051104  053111
6835  035372  051505  000072
6836  035376  005015  050012  053517  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
6837  035404  051105  052440  020120
6838  035412  042522  052123  051101
6839  035420  020124  047524  052040
6840  035426  051505  020124  006461
6841  035434  000012
6842  035436  047516  042516  005015  MSG12:  .ASCIZ  /NONE/<CR><LF>
6843  035444     000
6844  035445     015  047012  020117  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
6845  035452  020114  051117  050040
6846  035460  041440  047514  045503
6847  035466  020123  051120  051505
6848  035474  047105     124
6849  035477     015  040412  046114          .ASCIZ  <CR><LF>/ALL TIMING TESTS BYPASSED/
6850  035504  052040  046511  047111
6851  035512  020107  042524  052123
6852  035520  020123  054502  040520
6853  035526  051523  042105     000
6854  035533     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
6855  035540  051501  044523  043516
6856  035546  042040  044522  042526
6857  035554  000040
6858  035556  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
6859  035564  042526  000040
6860  035570  005015  051104  053111  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL NO. /
6861  035576  020105  042523  044522
6862  035604  046101  047040  027117
6863  035612  000040
6864  035614  005015  040503  052122  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
6865  035622  044522  043504  020105
6866  035630  042523  044522  046101
6867  035636  047040  027117  000040
```

```
 6868  035644  005015  047125  052111  MSG18:  .ASCIZ  <CR><LF>/UNIT SELECT PLUG TEST/<CR><LF>
 6869  035652  051440  046105  041505
 6870  035660  020124  046120  043525
 6871  035666  052040  051505  006524
 6872  035674  000012
 6873  035676  005015  047520  052122  MSG19:  .ASCIZ  <CR><LF>/PORT SELECTION TESTS/<CR><LF>
 6874  035704  051440  046105  041505
 6875  035712  044524  047117  052040
 6876  035720  051505  051524  005015
 6877  035726     000
 6878  035727     015  051012  047125  MSG20:  .ASCIZ  <CR><LF>/RUN-STOP SWITCH TEST/<CR><LF>
 6879  035734  051455  047524  020120
 6880  035742  053523  052111  044103
 6881  035750  052040  051505  006524
 6882  035756  000012
 6883  035760  005015  041501  046040  MSG21:  .ASCIZ  <CR><LF>/AC LOW DETECTION TEST-PART 1/<CR><LF>
 6884  035766  053517  042040  052105
 6885  035774  041505  044524  047117
 6886  036002  052040  051505  026524
 6887  036010  040520  052122  030440
 6888  036016  005015     000
 6889  036021     015  047012  047117  MSG22:  .ASCIZ  <CR><LF>/NON-EXECUTABLE FUNCTION (NXF) DETECTION TEST/<CR><LF>
 6890  036026  042455  042530  052503
 6891  036034  040524  046102  020105
 6892  036042  052506  041516  044524
 6893  036050  047117  024040  054116
 6894  036056  024506  042040  052105
 6895  036064  041505  044524  047117
 6896  036072  052040  051505  006524
 6897  036100  000012
 6898  036102  005015  047516  041440  MSG23:  .ASCIZ  <CR><LF>/NO CLOCK INTERRUPTS PRESENT, ABORTING TIMING TESTS/
 6899  036110  047514  045503  044440
 6900  036116  052116  051105  052522
 6901  036124  052120  020123  051120
 6902  036132  051505  047105  026124
 6903  036140  040440  047502  052122
 6904  036146  047111  020107  044524
 6905  036154  044515  043516  052040
 6906  036162  051505  051524     000
 6907  036167     015  053412  044522  MSG24:  .ASCIZ  <CR><LF>/WRITE LOCK TEST/<CR><LF>
 6908  036174  042524  046040  041517
 6909  036202  020113  042524  052123
 6910  036210  005015     000
 6911  036213     040  044515  051103  MSG25:  .ASCIZ  / MICRO SECONDS/
 6912  036220  020117  042523  047503
 6913  036226  042116  000123
 6914  036232  005015  041501  046040  MSG26:  .ASCIZ  <CR><LF>/AC LOW DETECTION TEST-PART 2/<CR><LF>
 6915  036240  053517  042040  052105
 6916  036246  041505  044524  047117
 6917  036254  052040  051505  026524
 6918  036262  040520  052122  031040
 6919  036270  005015     000
 6920  036273     015  005012  046101  MSG27:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
 6921  036300  020114  051104  053111
 6922  036306  051505  052040  051505
 6923  036314  042524  006504  005012
```

```
 6924  036322    000
 6925  036323    015  046412  046125  MSG28:  .ASCIZ  <CR><LF>/MULTIPLE DRIVE DETECTION TEST/<CR><LF>
 6926  036330  044524  046120  020105
 6927  036336  051104  053111  020105
 6928  036344  042504  042524  052103
 6929  036352  047511  020116  042524
 6930  036360  052123  005015    000
 6931  036365    015  050012  042514  MSG29:  .ASCIZ  <CR><LF>/PLEASE WAIT, HEADS BEING LOADED/<CR><LF>
 6932  036372  051501  020105  040527
 6933  036400  052111  020054  042510
 6934  036406  042101  020123  042502
 6935  036414  047111  020107  047514
 6936  036422  042101  042105  005015
 6937  036430    000
 6938  036431    015  053012  051105  MSG30:  .ASCIZ  <CR><LF>/VERIFY DOOR CAN NOW BE OPENED & LEAVE IT OPEN/<CR><LF>
 6939  036436  043111  020131  047504
 6940  036444  051117  041440  047101
 6941  036452  047040  053517  041040
 6942  036460  020105  050117  047105
 6943  036466  042105  023040  046040
 6944  036474  040505  042526  044440
 6945  036502  020124  050117  047105
 6946  036510  005015    000
 6947  036513    015  037412  005015  MSG31:  .ASCIZ  <CR><LF>/?/<CR><LF>
 6948  036520    000
 6949  036521    015  051012  046505  MSG32:  .ASCIZ  <CR><LF>/REMOVE DISK PACK & CLOSE DOOR/
 6950  036526  053117  020105  044504
 6951  036534  045523  050040  041501
 6952  036542  020113  020046  046103
 6953  036550  051517  020105  047504
 6954  036556  051117    000
 6955  036561    015  042012  050105  MSG33:  .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE DOOR OPEN/
 6956  036566  042522  051523  023440
 6957  036574  052522  026516  052123
 6958  036602  050117  020047  053523
 6959  036610  052111  044103  052040
 6960  036616  020117  051047  047125
 6961  036624  020047  044127  046111
 6962  036632  020105  047504  051117
 6963  036640  047440  042520  000116
 6964  036646  005015  042526  044522  MSG34:  .ASCIZ  <CR><LF>/VERIFY SPINDLE DOES NOT START & HEADS DO NOT LOAD/<CR><LF>
 6965  036654  054506  051440  044520
 6966  036662  042116  042514  042040
 6967  036670  042517  020123  047516
 6968  036676  020124  052123  051101
 6969  036704  020124  020046  042510
 6970  036712  042101  020123  047504
 6971  036720  047040  052117  046040
 6972  036726  040517  006504  000012
 6973  036734  005015  042504  051120  MSG35:  .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE CARTRIDGE REMOVED/
 6974  036742  051505  020123  051047
 6975  036750  047125  051455  047524
 6976  036756  023520  051440  044527
 6977  036764  041524  020110  047524
 6978  036772  023440  052522  023516
 6979  037000  053440  044510  042514
```

```
 6980  037006  041440  051101  051124
 6981  037014  042111  042507  051040
 6982  037022  046505  053117  042105
 6983  037030     000
 6984  037031     015  044412  051516  MSG36:  .ASCIZ  <CR><LF>/INSERT DISK PACK, DEPRESS 'RUN-STOP' TO 'RUN' & CLOSE DOOR/
 6985  037036  051105  020124  044504
 6986  037044  045523  050040  041501
 6987  037052  026113  042040  050105
 6988  037060  042522  051523  023440
 6989  037066  052522  026516  052123
 6990  037074  050117  020047  047524
 6991  037102  023440  052522  023516
 6992  037110  023040  041440  047514
 6993  037116  042523  042040  047517
 6994  037124  000122
 6995  037126  005015  042504  051120  MSG37:  .ASCIZ  <CR><LF>/DEPRESS SPACE BAR WHEN FINISHED/<CR><LF>
 6996  037134  051505  020123  050123
 6997  037142  041501  020105  040502
 6998  037150  020122  044127  047105
 6999  037156  043040  047111  051511
 7000  037164  042510  006504  000012
 7001  037172  005015  047111  042523  MSG38:  .ASCII  <CR><LF>/INSERT UNIT SELECT PLUGS, IN ANY ORDER/
 7002  037200  052122  052440  044516
 7003  037206  020124  042523  042514
 7004  037214  052103  050040  052514
 7005  037222  051507  020054  047111
 7006  037230  040440  054516  047440
 7007  037236  042122  051105
 7008  037242  005015  044124  020105          .ASCIZ  <CR><LF>/THE PROGRAM WILL ECHO THE UNIT SELECT PLUG NUMBER/<CR><LF>
 7009  037250  051120  043517  040522
 7010  037256  020115  044527  046114
 7011  037264  042440  044103  020117
 7012  037272  044124  020105  047125
 7013  037300  052111  051440  046105
 7014  037306  041505  020124  046120
 7015  037314  043525  047040  046525
 7016  037322  042502  006522  000012
 7017  037330  005015  042504  051120  MSG39:  .ASCIZ  <CR><LF>/DEPRESS CONTROL-C TO EXIT TEST/<CR><LF>
 7018  037336  051505  020123  047503
 7019  037344  052116  047522  026514
 7020  037352  020103  047524  042440
 7021  037360  044530  020124  042524
 7022  037366  052123  005015     000
 7023  037373     015  053012  046117  MSG40:  .ASCII  <CR><LF>/VOLUME VALID NOT SET/
 7024  037400  046525  020105  040526
 7025  037406  044514  020104  047516
 7026  037414  020124  042523     124
 7027  037421     015  046412  045501          .ASCIZ  <CR><LF>/MAKE SURE ORIGINAL UNIT SELECT PLUG IS INSERTED/<CR><LF>
 7028  037426  020105  052523  042522
 7029  037434  047440  044522  044507
 7030  037442  040516  020114  047125
 7031  037450  052111  051440  046105
 7032  037456  041505  020124  046120
 7033  037464  043525  044440  020123
 7034  037472  047111  042523  052122
 7035  037500  042105  005015     000
```

```
7036  037505  015     042012  051505  MSG41:  .ASCIZ  <CR><LF>/DESELECT PORT IN USE & SELECT OPPOSITE PORT/<CR><LF>
7037  037512  046105  041505  020124
7038  037520  047520  052122  044440
7039  037526  020116  051525  020105
7040  037534  020046  042523  042514
7041  037542  052103  047440  050120
7042  037550  051517  052111  020105
7043  037556  047520  052122  005015
7044  037564  000
7045  037565  015     042012  051505  MSG42:  .ASCIZ  <CR><LF>/DESELECT WRONG PORT & SELECT CORRECT PORT/<CR><LF>
7046  037572  046105  041505  020124
7047  037600  051127  047117  020107
7048  037606  047520  052122  023040
7049  037614  051440  046105  041505
7050  037622  020124  047503  051122
7051  037630  041505  020124  047520
7052  037636  052122  005015  000
7053  037643  015     042012  051505  MSG43:  .ASCIZ  <CR><LF>/DESELECT BOTH PORTS/<CR><LF>
7054  037650  046105  041505  020124
7055  037656  047502  044124  050040
7056  037664  051117  051524  005015
7057  037672  000
7058  037673  015     051412  046105  MSG44:  .ASCIZ  <CR><LF>/SELECT CORRECT PORT/<CR><LF>
7059  037700  041505  020124  047503
7060  037706  051122  041505  020124
7061  037714  047520  052122  005015
7062  037722  000
7063  037723  015     041412  051117  MSG45:  .ASCIZ  <CR><LF>/CORRECT PORT NOT SELECTED, TRY AGAIN/<CR><LF>
7064  037730  042522  052103  050040
7065  037736  051117  020124  047516
7066  037744  020124  042523  042514
7067  037752  052103  042105  020054
7068  037760  051124  020131  043501
7069  037766  044501  006516  000012
7070  037774  005015  047504  040440  MSG46:  .ASCII  <CR><LF>/DO A MANUAL DRIVE LOAD/
7071  040002  046440  047101  040525
7072  040010  020114  051104  053111
7073  040016  020105  047514  042101
7074  040024  005015  044124  020105          .ASCII  <CR><LF>/THE SPINDLE SHOULD START AND THE 'STOP' INDICATOR SHOULD GO OFF
7075  040032  050123  047111  046104
7076  040040  020105  044123  052517
7077  040046  042114  051440  040524
7078  040054  052122  040440  042116
7079  040062  052040  042510  023440
7080  040070  052123  050117  020047
7081  040076  047111  044504  040503
7082  040104  047524  020122  044123
7083  040112  052517  042114  043440
7084  040120  020117  043117  106
7085  040125  015     023412  042522          .ASCIZ  <CR><LF>/'READY' SHOULD GO ON IN APPROX. 1 MIN/<CR><LF>
7086  040132  042101  023531  051440
7087  040140  047510  046125  020104
7088  040146  047507  047440  020116
7089  040154  047111  040440  050120
7090  040162  047522  027130  030440
7091  040170  046440  047111  005015
```

```
7092  040176     000
7093  040177     015  053012  051105  MSG47:  .ASCIZ  <CR><LF>/VERIFY DOOR CANNOT BE OPENED (DO NOT FORCE)/<CR><LF>
7094  040204  043111  020131  047504
7095  040212  051117  041440  047101
7096  040220  047516  020124  042502
7097  040226  047440  042520  042516
7098  040234  020104  042050  020117
7099  040242  047516  020124  047506
7100  040250  041522  024505  005015
7101  040256     000
7102  040257     015  042012  050105  MSG48:  .ASCII  <CR><LF>/DEPRESS THE 'RUN-STOP' SWITCH TO 'STOP'/
7103  040264  042522  051523  052040
7104  040272  042510  023440  052522
7105  040300  026516  052123  050117
7106  040306  020047  053523  052111
7107  040314  044103  052040  020117
7108  040322  051447  047524  023520
7109  040330  005015  042526  044522          .ASCII  <CR><LF>/VERIFY THE HEADS UNLOAD, THE 'READY' LIGHT GOES OFF/
7110  040336  054506  052040  042510
7111  040344  044040  040505  051504
7112  040352  052440  046116  040517
7113  040360  026104  052040  042510
7114  040366  023440  042522  042101
7115  040374  023531  046040  043511
7116  040402  052110  043440  042517
7117  040410  020123  043117     106
7118  040415     015  023012  052040          .ASCIZ  <CR><LF>/& THE 'STOP' LIGHT GOES ON/<CR><LF>
7119  040422  042510  023440  052123
7120  040430  050117  020047  044514
7121  040436  044107  020124  047507
7122  040444  051505  047440  006516
7123  040452  000012
7124  040454  005015  052524  047122  MSG49:  .ASCIZ  <CR><LF>/TURN OFF AC POWER FROM BEHIND THE RK06/<CR><LF>
7125  040462  047440  043106  040440
7126  040470  020103  047520  042527
7127  040476  020122  051106  046517
7128  040504  041040  044105  047111
7129  040512  020104  044124  020105
7130  040520  045522  033060  005015
7131  040526     000
7132  040527     015  051412  044527  MSG50:  .ASCIZ  <CR><LF>/SWITCH AC POWER BACK ON/
7133  040534  041524  020110  041501
7134  040542  050040  053517  051105
7135  040550  041040  041501  020113
7136  040556  047117     000
7137  040561     015  046412  046517  MSG51:  .ASCII  <CR><LF>/MOMENTARILY REMOVE & INSERT UNIT SELECT PLUG/
7138  040566  047105  040524  044522
7139  040574  054514  051040  046505
7140  040602  053117  020105  020046
7141  040610  047111  042523  052122
7142  040616  052440  044516  020124
7143  040624  042523  042514  052103
7144  040632  050040  052514     107
7145  040637     015  052012  020117          .ASCIZ  <CR><LF>/TO RESET VOLUME VALID/<CR><LF>
7146  040644  042522  042523  020124
7147  040652  047526  052514  042515
```

```
 7148  040660  053040  046101  042111
 7149  040666  005015     000
 7150  040671     015  042012  050105  MSG52:  .ASCII  <CR><LF>/DEPRESS SPACE TO DO TEST/
 7151  040676  042522  051523  051440
 7152  040704  040520  042503  052040
 7153  040712  020117  047504  052040
 7154  040720  051505     124
 7155  040723     015  047412  020122          .ASCIZ  <CR><LF>/OR CONTROL-C TO BYPASS ENTIRE TEST/<CR><LF>
 7156  040730  047503  052116  047522
 7157  040736  026514  020103  047524
 7158  040744  041040  050131  051501
 7159  040752  020123  047105  044524
 7160  040760  042522  052040  051505
 7161  040766  006524  000012
 7162  040772  005015  044504  040523  MSG53:  .ASCIZ  <CR><LF>/DISABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES OFF/<CR><LF>
 7163  041000  046102  020105  044124
 7164  041006  020105  051127  052111
 7165  041014  020105  047514  045503
 7166  041022  051440  044527  041524
 7167  041030  026110  053040  051105
 7168  041036  043111  020131  044514
 7169  041044  044107  020124  047507
 7170  041052  051505  047440  043106
 7171  041060  005015     000
 7172  041063     015  042412  040516  MSG54:  .ASCIZ  <CR><LF>/ENABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES ON/<CR><LF>
 7173  041070  046102  020105  044124
 7174  041076  020105  051127  052111
 7175  041104  020105  047514  045503
 7176  041112  051440  044527  041524
 7177  041120  026110  053040  051105
 7178  041126  043111  020131  044514
 7179  041134  044107  020124  047507
 7180  041142  051505  047440  006516
 7181  041150  000012
 7182  041152  005015  054105  052111  MSG55:  .ASCIZ  <CR><LF>/EXIT TEST WITH ORIGINAL UNIT SELECT PLUG NO. /
 7183  041160  052040  051505  020124
 7184  041166  044527  044124  047440
 7185  041174  044522  044507  040516
 7186  041202  020114  047125  052111
 7187  041210  051440  046105  041505
 7188  041216  020124  046120  043525
 7189  041224  047040  027117  000040
 7190  041232  005015  046101  043511  MSG56:  .ASCII  <CR><LF>/ALIGNMENT CARTRIDGE USED/
 7191  041240  046516  047105  020124
 7192  041246  040503  052122  044522
 7193  041254  043504  020105  051525
 7194  041262  042105
 7195  041264  005015  051120  043517          .ASCII  <CR><LF>/PROGRAM WILL BYPASS THE WRITE LOCK TEST/
 7196  041272  040522  020115  044527
 7197  041300  046114  041040  050131
 7198  041306  051501  020123  044124
 7199  041314  020105  051127  052111
 7200  041322  020105  047514  045503
 7201  041330  052040  051505     124
 7202  041335     015  040412  042116          .ASCIZ  <CR><LF>/AND READ-WRITE DATA PORTION OF AC LOW DETECTION TEST-PART 2/<CR
 7203  041342  051040  040505  026504
```

```
7204  041350  051127  052111  020105
7205  041356  040504  040524  050040
7206  041364  051117  044524  047117
7207  041372  047440  020106  041501
7208  041400  046040  053517  042040
7209  041406  052105  041505  044524
7210  041414  047117  052040  051505
7211  041422  026524  040520  052122
7212  041430  031040  005015  000012
7213  041436  005015  042526  044522  MSG57:  .ASCIZ  <CR><LF>/VERIFY BATTERY RETRACT FUNCTIONAL/<CR><LF>
7214  041444  054506  041040  052101
7215  041452  042524  054522  051040
7216  041460  052105  040522  052103
7217  041466  043040  047125  052103
7218  041474  047511  040516  006514
7219  041502  000012
7220  041504  005015  047514  042101  MSG58:  .ASCIZ  <CR><LF>/LOAD HEADS ON ALL DRIVES TO BE TESTED FOR MDS/<CR><LF>
7221  041512  044040  040505  051504
7222  041520  047440  020116  046101
7223  041526  020114  051104  053111
7224  041534  051505  052040  020117
7225  041542  042502  052040  051505
7226  041550  042524  020104  047506
7227  041556  020122  042115  006523
7228  041564  000012
7229  041566  005015  047111  042523  MSG59:  .ASCIZ  <CR><LF>/INSERT SAME UNIT SELECT PLUG NUMBER IN ANY 2 DRIVES TO BE TESTE
7230  041574  052122  051440  046501
7231  041602  020105  047125  052111
7232  041610  051440  046105  041505
7233  041616  020124  046120  043525
7234  041624  047040  046525  042502
7235  041632  020122  047111  040440
7236  041640  054516  031040  042040
7237  041646  044522  042526  020123
7238  041654  047524  041040  020105
7239  041662  042524  052123  042105
7240  041670     000
7241  041671     015  044412  051516  MSG60:  .ASCII  <CR><LF>/INSERT CORRECT UNIT SELECT PLUGS & LOAD HEADS/
7242  041676  051105  020124  047503
7243  041704  051122  041505  020124
7244  041712  047125  052111  051440
7245  041720  046105  041505  020124
7246  041726  046120  043525  020123
7247  041734  020046  047514  042101
7248  041742  044040  040505  051504
7249  041750  005015  047117  050040          .ASCIZ  <CR><LF>/ON PREVIOUS 2 DRIVES/<CR><LF>
7250  041756  042522  044526  052517
7251  041764  020123  020062  051104
7252  041772  053111  051505  005015
7253  042000     000
7254  042001     015  046412  046125  MSG61:  .ASCIZ  <CR><LF>/MULTIPLE DRIVES FOUND ON DRIVE NO. /
7255  042006  044524  046120  020105
7256  042014  051104  053111  051505
7257  042022  043040  052517  042116
7258  042030  047440  020116  051104
7259  042036  053111  020105  047516
```

```
7260  042044  020056     000
7261  042047     015  041012  050131  MSG62:  .ASCII  <CR><LF>/BYPASSING MULT. DRIVE SELECT TEST/
7262  042054  051501  044523  043516
7263  042062  046440  046125  027124
7264  042070  042040  044522  042526
7265  042076  051440  046105  041505
7266  042104  020124  042524  052123
7267  042112  005015  047117  054514          .ASCIZ  <CR><LF>/ONLY 1 DRIVE PRESENT/<CR><LF>
7268  042120  030440  042040  044522
7269  042126  042526  050040  042522
7270  042134  042523  052116  005015
7271  042142     000
7272  042143     015  042012  050105  MSG63:  .ASCII  <CR><LF>/DEPRESS SPACE BAR TO DO ANOTHER 2 DRIVES/
7273  042150  042522  051523  051440
7274  042156  040520  042503  041040
7275  042164  051101  052040  020117
7276  042172  047504  040440  047516
7277  042200  044124  051105  031040
7278  042206  042040  044522  042526
7279  042214     123
7280  042215     015  047412  020122          .ASCII  <CR><LF>/OR CONRTOL-C TO EXIT TEST/
7281  042222  047503  051116  047524
7282  042230  026514  020103  047524
7283  042236  042440  044530  020124
7284  042244  042524  052123
7285  042250  005015  044450  051516          .ASCIZ  <CR><LF>/(INSERT CORRECT UNIT SELECT PLUGS BEFORE EXITING)/<CR><LF>
7286  042256  051105  020124  047503
7287  042264  051122  041505  020124
7288  042272  047125  052111  051440
7289  042300  046105  041505  020124
7290  042306  046120  043525  020123
7291  042314  042502  047506  042522
7292  042322  042440  044530  044524
7293  042330  043516  006451  000012
7294  042336  005015  044124  020105  MSG64:  .ASCII  <CR><LF>/THE SECOND PORTION OF THIS TEST IS TEMPORARILY BYPASSED/
7295  042344  042523  047503  042116
7296  042352  050040  051117  044524
7297  042360  047117  047440  020106
7298  042366  044124  051511  052040
7299  042374  051505  020124  051511
7300  042402  052040  046505  047520
7301  042410  040522  044522  054514
7302  042416  041040  050131  051501
7303  042424  042523     104
7304  042427     015  051412  042505          .ASCIZ  <CR><LF>/SEE COMMENTS IN THE LISTING/<CR><LF>
7305  042434  041440  046517  042515
7306  042442  052116  020123  047111
7307  042450  052040  042510  046040
7308  042456  051511  044524  043516
7309  042464  005015     000
7310  042467     015  042012  050105  MSG65:  .ASCIZ  <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'STOP'/
7311  042474  042522  051523  023440
7312  042502  052522  026516  052123
7313  042510  050117  020047  053523
7314  042516  052111  044103  052040
7315  042524  020117  051447  047524
```

```
 7316  042532  023520     000
 7317  042535     015  042012  050105  MSG66:  .ASCII  <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN'/
 7318  042542  042522  051523  023440
 7319  042550  052522  026516  052123
 7320  042556  050117  020047  053523
 7321  042564  052111  044103  052040
 7322  042572  020117  051047  047125
 7323  042600     047
 7324  042601     015  023012  042040          .ASCIZ  <CR><LF>/& DEPRESS SPACE WHEN 'READY' LIGHT GOES ON/<CR><LF>
 7325  042606  050105  042522  051523
 7326  042614  051440  040520  042503
 7327  042622  053440  042510  020116
 7328  042630  051047  040505  054504
 7329  042636  020047  044514  044107
 7330  042644  020124  047507  051505
 7331  042652  047440  006516  000012
 7332  042660  005015  042504  042523  MSG67:  .ASCIZ  <CR><LF>/DESELECT PORT SWITCH ON ALL OTHER DRIVES/<CR><LF>
 7333  042666  042514  052103  050040
 7334  042674  051117  020124  053523
 7335  042702  052111  044103  047440
 7336  042710  020116  046101  020114
 7337  042716  052117  042510  020122
 7338  042724  051104  053111  051505
 7339  042732  005015     000
 7340  042735     015  053012  051105  MSG68:  .ASCIZ  <CR><LF>/VERIFY BOTH DRIVES UNLOADED/<CR><LF>
 7341  042742  043111  020131  047502
 7342  042750  044124  042040  044522
 7343  042756  042526  020123  047125
 7344  042764  047514  042101  042105
 7345  042772  005015     000
 7346  042775     015  042012  050105  MSG69:  .ASCIZ  <CR><LF>/DEPRESS SPACE WHEN 'READY' LIGHT GOES ON/<CR><LF>
 7347  043002  042522  051523  051440
 7348  043010  040520  042503  053440
 7349  043016  042510  020116  051047
 7350  043024  040505  054504  020047
 7351  043032  044514  044107  020124
 7352  043040  047507  051505  047440
 7353  043046  006516  000012
 7354  043052  005015  042526  044522  MSG70:  .ASCIZ <CR><LF>/VERIFY HEADS LOAD/<CR><LF>
 7355  043060  054506  044040  040505
 7356  043066  051504  046040  040517
 7357  043074  006504  000012
 7358  043100  005015  042523  042514  MSG71:  .ASCIZ  <CR><LF>/SELECT CORRECT PORT SWITCH ON ALL OTHER DRIVES/<CR><LF>
 7359  043106  052103  041440  051117
 7360  043114  042522  052103  050040
 7361  043122  051117  020124  053523
 7362  043130  052111  044103  047440
 7363  043136  020116  046101  020114
 7364  043144  052117  042510  020122
 7365  043152  051104  053111  051505
 7366  043160  005015     000
 7367
 7368
 7369                                  .SBTTL  ERROR MESSAGES
 7370
 7371  043163     015  042412  051122  EM1:    .ASCIZ  <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
```

```
7372  043170  051117  020054  047117
7373  043176  054514  030040  052040
7374  043204  051110  020125  020067
7375  043212  046101  047514  042527
7376  043220  026104  052040  054522
7377  043226  040440  040507  047111
7378  043234  005015     000
7379  043237     104  044522  042526  EM2:    .ASCIZ  /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
7380  043244  021440  044440  020116
7381  043252  045522  051503  020062
7382  043260  040503  047116  052117
7383  043266  041040  020105  042522
7384  043274  042101  041040  041501
7385  043302  020113  047503  051122
7386  043310  041505  046124  020131
7387  043316  047111  051040  046513
7388  043324  031122     000
7389  043327     015  040412  047502  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
7390  043334  052122  052040  051505
7391  043342  051524  027056  052456
7392  043350  042516  050130  041505
7393  043356  042524  020104  044524
7394  043364  042515  047440  052125
7395  043372  040440  020124  041520
7396  043400  000075
7397  043402  005015  041101  051117  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
7398  043410  020124  042524  052123
7399  043416  027123  027056  047125
7400  043424  054105  042520  052103
7401  043432  042105  044440  052116
7402  043440  051105  052522  052120
7403  043446  040440  020124  041520
7404  043454  000075
7405  043456  042115  020123  042523  EM5:    .ASCIZ  /MDS SET IN RKCS2/
7406  043464  020124  047111  051040
7407  043472  041513  031123     000
7408  043477     125  042506  051440  EM6:    .ASCIZ  /UFE SET IN RKCS2/
7409  043504  052105  044440  020116
7410  043512  045522  051503  000062
7411  043520  051104  020101  047111  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
7412  043526  051040  042113  020123
7413  043534  020046  042516  020104
7414  043542  047111  051040  041513
7415  043550  031123  051040  051505
7416  043556  052105  020073  051127
7417  043564  047117  020107  047520
7418  043572  052122  051440  046105
7419  043600  041505  042524  037504
7420  043606     000
7421  043607     104  044522  042526  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
7422  043614  050040  042522  042523
7423  043622  052116  041040  052125
7424  043630  047040  052117  051440
7425  043636  042520  044503  044506
7426  043644  042105  041040  020131
7427  043652  050117  051105  052101
```

```
7428  043660  051117     000
7429  043663     104  044522  042526  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
7430  043670  047040  052117  050040
7431  043676  042522  042523  052116
7432  043704  041040  052125  051440
7433  043712  042520  044503  044506
7434  043720  042105  041040  020131
7435  043726  050117  051105  052101
7436  043734  051117     000
7437  043737     101  047502  052122  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
7438  043744  052040  051505  051524
7439  043752  027056  041456  047101
7440  043760  047516  020124  042522
7441  043766  042506  042522  041516
7442  043774  020105  047503  052116
7443  044002  047522  046114  051105
7444  044010  051040  043505  051511
7445  044016  042524  000122
7446  044022  051104  020101  047111  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
7447  044030  051040  042113  020123
7448  044036  020046  042516  020104
7449  044044  047111  051040  041513
7450  044052  031123  041040  052117
7451  044060  020110  042523  000124
7452  044066  047503  052116  047522  EM12:   .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
7453  044074  046114  051105  047040
7454  044102  052117  051040  040505
7455  044110  054504  044440  020116
7456  044116  045522  051503  000061
7457  044124  047516  040440  052124  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
7458  044132  020116  047111  051040
7459  044140  040513  047523  000106
7460  044146  051127  047117  020107  EM14:   .ASCIZ  /WRONG ATTN IN RKASOF/
7461  044154  052101  047124  044440
7462  044162  020116  045522  051501
7463  044170  043117     000
7464  044173     104  042122  020131  EM15:   .ASCIZ  /DRDY NOT CLEARED IN RKMR2/
7465  044200  047516  020124  046103
7466  044206  040505  042522  020104
7467  044214  047111  051040  046513
7468  044222  031122     000
7469  044225     104  041523  047040  EM16:   .ASCIZ  /DSC NOT SET IN RKMR2/
7470  044232  052117  051440  052105
7471  044240  044440  020116  045522
7472  044246  051115  000062
7473  044252  042515  051523  043501  EM17:   .ASCIZ  /MESSAGE A0 ERROR/
7474  044260  020105  030101  042440
7475  044266  051122  051117     000
7476  044273     115  051505  040523  EM18:   .ASCIZ  /MESSAGE B0 ERROR/
7477  044300  042507  041040  020060
7478  044306  051105  047522  000122
7479  044314  042515  051523  043501  EM19:   .ASCIZ  /MESSAGE A1 ERROR/
7480  044322  020105  030501  042440
7481  044330  051122  051117     000
7482  044335     115  051505  040523  EM20:   .ASCIZ  /MESSAGE B1 ERROR/
7483  044342  042507  041040  020061
```

```
7484  044350  051105  047522  000122
7485  044356  042503  051122  051440  EM21:   .ASCIZ  /CERR SET IN RKCS1/
7486  044364  052105  044440  020116
7487  044372  045522  051503  000061
7488  044400  047504  051117  051440  EM22:   .ASCIZ  /DOOR STATUS IN RKMR2 NOT CLEARED/
7489  044406  040524  052524  020123
7490  044414  047111  051040  046513
7491  044422  031122  047040  052117
7492  044430  041440  042514  051101
7493  044436  042105     000
7494  044441     123  044520  042116  EM23:   .ASCIZ  /SPINDLE ON SET IN RKMR2/
7495  044446  042514  047440  020116
7496  044454  042523  020124  047111
7497  044462  051040  046513  031122
7498  044470     000
7499  044471     126  046117  046525  EM24:   .ASCIZ  /VOLUME VALID NOT SET IN RKMR2/
7500  044476  020105  040526  044514
7501  044504  020104  047516  020124
7502  044512  042523  020124  047111
7503  044520  051040  046513  031122
7504  044526     000
7505  044527     103  051101  051124  EM25:   .ASCIZ  /CARTRIDGE STATUS IN RKMR2 NOT CLEARED/
7506  044534  042111  042507  051440
7507  044542  040524  052524  020123
7508  044550  047111  051040  046513
7509  044556  031122  047040  052117
7510  044564  041440  042514  051101
7511  044572  042105     000
7512  044575     126  046117  046525  EM26:   .ASCIZ  /VOLUME VALID NOT CLEARED IN RKMR2/
7513  044602  020105  040526  044514
7514  044610  020104  047516  020124
7515  044616  046103  040505  042522
7516  044624  020104  047111  051040
7517  044632  046513  031122     000
7518  044637     116  042105  047040  EM27:   .ASCIZ  /NED NOT SET IN RKCS2/
7519  044644  052117  051440  052105
7520  044652  044440  020116  045522
7521  044660  051503  000062
7522  044664  047526  052514  042515  EM28:   .ASCIZ  /VOLUME VALID SET IN RKMR2/
7523  044672  053040  046101  042111
7524  044700  051440  052105  044440
7525  044706  020116  045522  051115
7526  044714  000062
7527  044716  051504  020103  047516  EM29:   .ASCIZ  /DSC NOT SET IN RKMR2/
7528  044724  020124  042523  020124
7529  044732  047111  051040  046513
7530  044740  031122     000
7531  044743     101  052124  020116  EM30:   .ASCIZ  /ATTN NOT RESET IN RKASOF/
7532  044750  047516  020124  042522
7533  044756  042523  020124  047111
7534  044764  051040  040513  047523
7535  044772  000106
7536  044774  042516  020104  047516  EM31:   .ASCIZ  /NED NOT CLEARED IN RKCS2/
7537  045002  020124  046103  040505
7538  045010  042522  020104  047111
7539  045016  051040  041513  031123
```

```
7540  045024     000
7541  045025     123  044520  042116  EM32:   .ASCIZ  /SPINDLE ON NOT SET IN RKMR2/
7542  045032  042514  047440  020116
7543  045040  047516  020124  042523
7544  045046  020124  047111  051040
7545  045054  046513  031122     000
7546  045061     104  044522  042526  EM33:   .ASCIZ  /DRIVE NOT READY IN RKMR2/
7547  045066  047040  052117  051040
7548  045074  040505  054504  044440
7549  045102  020116  045522  051115
7550  045110  000062
7551  045112  047504  051117  051440  EM34:   .ASCIZ  /DOOR STATUS BIT NOT SET IN RKMR2/
7552  045120  040524  052524  020123
7553  045126  044502  020124  047516
7554  045134  020124  042523  020124
7555  045142  047111  051040  046513
7556  045150  031122     000
7557  045153     110  040505  051504  EM35:   .ASCIZ  /HEADS HOME NOT SET IN RKMR2/
7558  045160  044040  046517  020105
7559  045166  047516  020124  042523
7560  045174  020124  047111  051040
7561  045202  046513  031122     000
7562  045207     103  046131  040440  EM36:   .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
7563  045214  042104  020122  047111
7564  045222  051040  046513  031522
7565  045230  047040  052117  051440
7566  045236  046501  020105  051501
7567  045244  051040  042113  000103
7568  045252  041501  046040  053517  EM37:   .ASCIZ  /AC LOW NOT SET IN RKMR3/
7569  045260  047040  052117  051440
7570  045266  052105  044440  020116
7571  045274  045522  051115  000063
7572  045302  041501  046040  053517  EM38:   .ASCIZ  /AC LOW DID NOT SET FAULT IN RKMR3/
7573  045310  042040  042111  047040
7574  045316  052117  051440  052105
7575  045324  043040  052501  052114
7576  045332  044440  020116  045522
7577  045340  051115  000063
7578  045344  054503  020114  044504  EM39:   .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
7579  045352  043106  023040  047440
7580  045360  043106  042523  020124
7581  045366  047111  051040  046513
7582  045374  031122  047040  052117
7583  045402  041440  042514  051101
7584  045410  042105     000
7585  045413     103  046131  040440  EM40:   .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
7586  045420  042104  020122  047111
7587  045426  051040  046513  031522
7588  045434  047040  052117  041440
7589  045442  042514  051101  042105
7590  045450     000
7591  045451     103  046131  040440  EM41:   .ASCIZ  /CYL ADDR IN RKMR3 DID NOT REMAIN CLEARED/
7592  045456  042104  020122  047111
7593  045464  051040  046513  031522
7594  045472  042040  042111  047040
7595  045500  052117  051040  046505
```

```
7596  045506  044501  020116  046103
7597  045514  040505  042522  000104
7598  045522  042516  020104  047516  EM42:   .ASCIZ  /NED NOT SET IN RKCS2/
7599  045530  020124  042523  020124
7600  045536  047111  051040  041513
7601  045544  031123     000
7602  045547     101  046103  020117  EM43:   .ASCIZ  /ACLO NOT CLEARED IN RKMR3/
7603  045554  047516  020124  046103
7604  045562  040505  042522  020104
7605  045570  047111  051040  046513
7606  045576  031522     000
7607  045601     126  046117  046525  EM44:   .ASCIZ  /VOLUME VALID NOT CLEARED IN RKMR2/
7608  045606  020105  040526  044514
7609  045614  020104  047516  020124
7610  045622  046103  040505  042522
7611  045630  020104  047111  051040
7612  045636  046513  031122     000
7613  045643     126  046117  046525  EM45:   .ASCIZ  /VOLUME VALID SET IN RKMR2 AFTER HEADS LOADED/
7614  045650  020105  040526  044514
7615  045656  020104  042523  020124
7616  045664  047111  051040  046513
7617  045672  031122  040440  052106
7618  045700  051105  044040  040505
7619  045706  051504  046040  040517
7620  045714  042504  000104
7621  045720  047516  026516  054105  EM46:   .ASCIZ  /NON-EXECUTABLE FUNCTION (NXF) NOT SET IN RKMR3/
7622  045726  041505  052125  041101
7623  045734  042514  043040  047125
7624  045742  052103  047511  020116
7625  045750  047050  043130  020051
7626  045756  047516  020124  042523
7627  045764  020124  047111  051040
7628  045772  046513  031522     000
7629  045777     103  046131  047111  EM47:   .ASCIZ  /CYLINDER ADDRESS CHANGED FROM 0/
7630  046004  042504  020122  042101
7631  046012  051104  051505  020123
7632  046020  044103  047101  042507
7633  046026  020104  051106  046517
7634  046034  030040     000
7635  046037     127  044522  042524  EM48:   .ASCIZ  /WRITE LOCK IN RKMR2 NOT CLEARED/
7636  046044  046040  041517  020113
7637  046052  047111  051040  046513
7638  046060  031122  047040  052117
7639  046066  041440  042514  051101
7640  046074  042105     000
7641  046077     127  044522  042524  EM49:   .ASCIZ  /WRITE LOCK IN RKMR2 NOT SET/
7642  046104  046040  041517  020113
7643  046112  047111  051040  046513
7644  046120  031122  047040  052117
7645  046126  051440  052105     000
7646  046133     127  044522  042524  EM50:   .ASCIZ  /WRITE LOCK ERROR IN RKMR3 NOT SET/
7647  046140  046040  041517  020113
7648  046146  051105  047522  020122
7649  046154  047111  051040  046513
7650  046162  031522  047040  052117
7651  046170  051440  052105     000
```

```
7652  046175  127     044522  042524  EM51:   .ASCIZ  /WRITE LOCK DID NOT OCCUR AT SECTOR BOUNDRY/
7653  046202  046040  041517  020113
7654  046210  044504  020104  047516
7655  046216  020124  041517  052503
7656  046224  020122  052101  051440
7657  046232  041505  047524  020122
7658  046240  047502  047125  051104
7659  046246  000131
7660  046250  047125  020123  047516  EM52:   .ASCIZ  /UNS NOT SET IN RKMR3/
7661  046256  020124  042523  020124
7662  046264  047111  051040  046513
7663  046272  031522  000
7664
7665  046275  125     046116  040517  EM53:   .ASCIZ  /UNLOAD NOT SET IN RKMR2/
7666  046302  020104  047516  020124
7667  046310  042523  020124  047111
7668  046316  051040  046513  031122
7669  046324  000
7670  046325  103     047101  047516  EM54:   .ASCIZ  /CANNOT FIND MULT. DRIVE SELECT IN RKCS2/
7671  046332  020124  044506  042116
7672  046340  046440  046125  027124
7673  046346  042040  044522  042526
7674  046354  051440  046105  041505
7675  046362  020124  047111  051040
7676  046370  041513  031123  000
7677  046375  101     052124  020116  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
7678  046402  047516  020124  046103
7679  046410  040505  042522  020104
7680  046416  047111  051040  040513
7681  046424  047523  000106
7682  046430  047125  054105  042520  EM56:   .ASCIZ  /UNEXPECTED MEMORY PARITY ERROR TRAP/
7683  046436  052103  042105  046440
7684  046444  046505  051117  020131
7685  046452  040520  044522  054524
7686  046460  042440  051122  051117
7687  046466  052040  040522  000120
7688  045474  042503  051122  044440  EM57:   .ASCIZ  /CERR IN RKCS1 NOT SET/
7689  046502  020116  045522  051503
7690  046510  020061  047516  020124
7691  046516  042523  000124
7692  046522  042510  042101  020123  EM60:   .ASCIZ  /HEADS HOME NOT FOUND IN RKMR2/
7693  046530  047510  042515  047040
7694  046536  052117  043040  052517
7695  046544  042116  044440  020116
7696  046552  045522  051115  000062
7697  046560  054116  020106  044504  EM61:   .ASCIZ  /NXF DID NOT SET FAULT/
7698  046566  020104  047516  020124
7699  046574  042523  020124  040506
7700  046602  046125  000124
7701  046606  052104  020105  042523  EM62:   .ASCIZ  /DTE SET IN RKER/
7702  046614  020124  047111  051040
7703  046622  042513  000122
7704  046626  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
7705  046634  020124  047111  051040
7706  046642  041513  031123  000
7707  046647  122     042113  020103  EM64:   .ASCII  /RKDC & RKDA INDICATE THAT WRITE CHECK ERROR/
```

```
7708  046654  020046  045522  040504
7709  046662  044440  042116  041511
7710  046670  052101  020105  044124
7711  046676  052101  053440  044522
7712  046704  042524  041440  042510
7713  046712  045503  042440  051122
7714  046720  051117
7715  046722  005015  041517  052503          .ASCIZ  <CR><LF>/OCCURRED AT CYL 411, TRACK 2, SECTOR 21/
7716  046730  051122  042105  040440
7717  046736  020124  054503  020114
7718  046744  030464  026061  052040
7719  046752  040522  045503  031040
7720  046760  020054  042523  052103
7721  046766  051117  031040  000061
7722  046774  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERROR/
7723  047002  040505  042504  020122
7724  047010  051105  047522  000122
7725  047016  054503  020114  042101  EM66:   .ASCIZ  /CYL ADDR IN RKMR3 INCORRECT/
7726  047024  051104  044440  020116
7727  047032  045522  051115  020063
7728  047040  047111  047503  051122
7729  047046  041505  000124
7730  047052  040503  047116  052117  EM68:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
7731  047060  051040  040505  020104
7732  047066  040502  020104  042523
7733  047074  052103  051117  044440
7734  047102  043116  051117  040515
7735  047110  044524  047117     000
7736  047115     122  055124  047040  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
7737  047122  052117  051440  052105
7738  047130  044440  020116  045522
7739  047136  051115  000062
7740  047142  051127  052111  020105  EM80:   .ASCIZ  /WRITE CHECK ERROR SET IN RKCS2/
7741  047150  044103  041505  020113
7742  047156  051105  047522  020122
7743  047164  042523  020124  047111
7744  047172  051040  041513  031123
7745  047200     000
7746  047201     127  044522  042524  EM81:   .ASCIZ  /WRITE CHECK COMMAND NOT FUNCTIONING/
7747  047206  041440  042510  045503
7748  047214  041440  046517  040515
7749  047222  042116  047040  052117
7750  047230  043040  047125  052103
7751  047236  047511  044516  043516
7752  047244     000
7753  047245     122  040505  020104  EM82:   .ASCIZ  /READ DATA DID NOT COMPARE WITH WRITE DATA/
7754  047252  040504  040524  042040
7755  047260  042111  047040  052117
7756  047266  041440  046517  040520
7757  047274  042522  053440  052111
7758  047302  020110  051127  052111
7759  047310  020105  040504  040524
7760  047316     000
7761  047317     104  052101  020101  EM83:   .ASCIZ  /DATA CHECK ERROR SET IN RKER/
7762  047324  044103  041505  020113
7763  047332  051105  047522  020122
```

```
7764  047340  042523  020124  047111
7765  047346  051040  042513  000122
7766  047354  043117  051506  052105  EM84:   .ASCIZ  /OFFSET REG IN RKMR2  NOT  177/
7767  047362  051040  043505  044440
7768  047370  020116  045522  051115
7769  047376  020062  047040  052117
7770  047404  020040  033461  000067
7771  047412  043117  051506  052105  EM85:   .ASCIZ  /OFFSET STATUS BIT IN RKMR2 CLEARED/
7772  047420  051440  040524  052524
7773  047426  020123  044502  020124
7774  047434  047111  051040  046513
7775  047442  031122  041440  042514
7776  047450  051101  042105     000
7777  047455     117  043106  042523  EM86:   .ASCIZ  /OFFSET REG IN RKMR2 NOT 77/
7778  047462  020124  042522  020107
7779  047470  047111  051040  046513
7780  047476  031122  047040  052117
7781  047504  033440  000067
7782  047510  051127  052111  020105  EM87:   .ASCIZ  /WRITE CHECK FAILURE AT OFFSET IN RKASOF/
7783  047516  044103  041505  020113
7784  047524  040506  046111  051125
7785  047532  020105  052101  047440
7786  047540  043106  042523  020124
7787  047546  047111  051040  040513
7788  047554  047523  000106
7789  047560  047516  053440  044522  EM88:   .ASCIZ  /NO WRITE CHECK ERROR/
7790  047566  042524  041440  042510
7791  047574  045503  042440  051122
7792  047602  051117     000
7793  047605     110  040505  051504  EM89:   .ASCIZ  /HEADS HOME NOT CLEARED IN RKMR2/
7794  047612  044040  046517  020105
7795  047620  047516  020124  046103
7796  047626  040505  042522  020104
7797  047634  047111  051040  046513
7798  047642  031122     000
7799  047645     124  040522  045503  EM90:   .ASCIZ  /TRACK FOLL OK NOT SET IN RKMR2/
7800  047652  043040  046117  020114
7801  047660  045517  047040  052117
7802  047666  051440  052105  044440
7803  047674  020116  045522  051115
7804  047702  000062
7805  047704  042522  020126  047516  EM91:   .ASCIZ  /REV NOT SET IN RKMR2/
7806  047712  020124  042523  020124
7807  047720  047111  051040  046513
7808  047726  031122     000
7809  047731     122  053105  047040  EM92:   .ASCIZ  /REV NOT CLEARED IN RKMR2/
7810  047736  052117  041440  042514
7811  047744  051101  042105  044440
7812  047752  020116  045522  051115
7813  047760  000062
7814  047762  042522  042101  047111  EM93:   .ASCIZ  /READING WRONG CYLINDER # IN HEADER/
7815  047770  020107  051127  047117
7816  047776  020107  054503  044514
7817  050004  042116  051105  021440
7818  050012  044440  020116  042510
7819  050020  042101  051105     000
```

```
7820  050025     117  043106  042523  EM94:   .ASCIZ  /OFFSET STATUS BIT IN RKMR2 NOT CLEARED/
7821  050032  020124  052123  052101
7822  050040  051525  041040  052111
7823  050046  044440  020116  045522
7824  050054  051115  020062  047516
7825  050062  020124  046103  040505
7826  050070  042522  000104
7827  050074  047506  046522  052101  EM95:   .ASCIZ  /FORMAT BIT NOT SET IN RKMR2/
7828  050102  041040  052111  047040
7829  050110  052117  051440  052105
7830  050116  044440  020116  045522
7831  050124  051115  000062
7832  050130  040503  047116  052117  EM96:   .ASCIZ  /CANNOT FIND SECTOR 17/
7833  050136  043040  047111  020104
7834  050144  042523  052103  051117
7835  050152  030440  000067
7836  050156  042510  042101  051440  EM97:   .ASCIZ  /HEAD SWITCHING LONGER THAN 16 MS/
7837  050164  044527  041524  044510
7838  050172  043516  046040  047117
7839  050200  042507  020122  044124
7840  050206  047101  030440  020066
7841  050214  051515     000
7842  050217     103  047101  047516  EM98:   .ASCIZ  /CANNOT FIND CYLINDER 128/
7843  050224  020124  044506  042116
7844  050232  041440  046131  047111
7845  050240  042504  020122  031061
7846  050246  000070
7847  050250  040503  047116  052117  EM99:   .ASCIZ  /CANNOT FIND CYLINDER 256/
7848  050256  043040  047111  020104
7849  050264  054503  044514  042116
7850  050272  051105  031040  033065
7851  050300     000
7852  050301     104  044522  042526  EM100:  .ASCIZ  /DRIVE OFF TRACK SET IN RKMR3/
7853  050306  047440  043106  052040
7854  050314  040522  045503  051440
7855  050322  052105  044440  020116
7856  050330  045522  051115  000063
7857  050336  044504  020104  047516  EM101:  .ASCIZ  /DID NOT GO TO CYLINDER 0/
7858  050344  020124  047507  052040
7859  050352  020117  054503  044514
7860  050360  042116  051105  030040
7861  050366     000
7862
7863                                  .SBTTL  DATA HEADERS
7864
7865  050367     124  051505  020124  DH1:    .ASCIZ  /TEST NO.  PC/
7866  050374  047516  020056  050040
7867  050402  000103
7868  050404  045522  051115  004462  DH2:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKCS1   RKCS2/
7869  050412  045522  051115  004463
7870  050420  045522  051105  051011
7871  050426  042113  004523  045522
7872  050434  051503  004461  045522
7873  050442  051503  000062
7874  050446  045522  051501  043117  DH3:    .ASCIZ  /RKASOF/
7875  050454     000
```

```
7876  050455     123  047510  046125  DH4:    .ASCIZ  /SHOULD BE/
7877  050462  020104  042502     000
7878  050467     122  046513  031122  DH5:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDC    RKCS1   RKCS2/
7879  050474  051011  046513  031522
7880  050502  051011  042513  004522
7881  050510  045522  051504  051011
7882  050516  042113  004503  045522
7883  050524  051503  004461  045522
7884  050532  051503  000062
7885  050536  045522  051115  020062  DH6:    .ASCIZ  /RKMR2   RKMR3   RKDC   FROM CYL  TO CYL  CYL DIFF/
7886  050544  020040  045522  051115
7887  050552  020063  020040  045522
7888  050560  041504  020040  043040
7889  050566  047522  020115  054503
7890  050574  020114  052040  020117
7891  050602  054503  020114  041440
7892  050610  046131  042040  043111
7893  050616  000106
7894  050620  045522  051115  004462  DH7:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDA    RKCS1   RKCS2/
7895  050626  045522  051115  004463
7896  050634  045522  051105  051011
7897  050642  042113  004523  045522
7898  050650  040504  051011  041513
7899  050656  030523  051011  041513
7900  050664  031123     000
7901  050667     101  052106  051105  DH8:    .ASCIZ  /AFTER DRIVE UNLOADED & DOOR OPENED/
7902  050674  042040  044522  042526
7903  050702  052440  046116  040517
7904  050710  042504  020104  020046
7905  050716  047504  051117  047440
7906  050724  042520  042516  000104
7907  050732  043101  042524  020122  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
7908  050740  052123  051101  020124
7909  050746  050123  047111  046104
7910  050754  020105  047503  046515
7911  050762  047101  020104  042522
7912  050770  023503  020104  054502
7913  050776  042040  044522  042526
7914  051004     000
7915  051005     101  020124  047105  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
7916  051012  020104  043117  044040
7917  051020  040505  020104  047514
7918  051026  042101  047111  000107
7919  051034  043101  042524  020122  DH11:   .ASCIZ  /AFTER MANUALLY LOADING HEADS WITH DOOR OPEN/
7920  051042  040515  052516  046101
7921  051050  054514  046040  040517
7922  051056  044504  043516  044040
7923  051064  040505  051504  053440
7924  051072  052111  020110  047504
7925  051100  051117  047440  042520
7926  051106  000116
7927  051110  043101  042524  020122  DH12:   .ASCIZ  /AFTER DISK PACK REMOVED/
7928  051116  044504  045523  050040
7929  051124  041501  020113  042522
7930  051132  047515  042526  000104
7931  051140  043101  042524  020122  DH13:   .ASCIZ  /AFTER MANUALLY LOADING HEADS WITH DISK PACK REMOVED/
```

```
7932  051146  040515  052516  046101
7933  051154  054514  046040  040517
7934  051162  044504  043516  044040
7935  051170  040505  051504  053440
7936  051176  052111  020110  044504
7937  051204  045523  050040  041501
7938  051212  020113  042522  047515
7939  051220  042526  000104
7940  051224  043101  042524  020122  DH14:   .ASCIZ  /AFTER VOLUME VALID RESET/
7941  051232  047526  052514  042515
7942  051240  053040  046101  042111
7943  051246  051040  051505  052105
7944  051254     000
7945  051255     127  052111  047510  DH15:   .ASCIZ  /WITHOUT PACK COMMAND/
7946  051262  052125  050040  041501
7947  051270  020113  047503  046515
7948  051276  047101  000104
7949  051302  043101  042524  020122  DH16:   .ASCIZ  /AFTER UNIT SELECT PLUG REMOVED/
7950  051310  047125  052111  051440
7951  051316  046105  041505  020124
7952  051324  046120  043525  051040
7953  051332  046505  053117  042105
7954  051340     000
7955  051341     101  052106  051105  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
7956  051346  051040  041505  046101
7957  051354  041440  046517  040515
7958  051362  042116     000
7959  051365     101  052106  051105  DH18:   .ASCIZ  /AFTER UNLOAD COMMAND/
7960  051372  052440  046116  040517
7961  051400  020104  047503  046515
7962  051406  047101  000104
7963  051412  043101  042524  020122  DH19:   .ASCIZ  /AFTER PACK COMMAND/
7964  051420  040520  045503  041440
7965  051426  046517  040515  042116
7966  051434     000
7967  051435     101  052106  051105  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
7968  051442  051440  046105  041505
7969  051450  020124  051104  053111
7970  051456  020105  047503  046515
7971  051464  047101  000104
7972  051470  043101  042524  020122  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
7973  051476  052523  051502  051531
7974  051504  042524  020115  046103
7975  051512  040505  000122
7976  051516  043101  042524  020122  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
7977  051524  051104  053111  020105
7978  051532  046103  040505  020122
7979  051540  047503  046515  047101
7980  051546  000104
7981  051550  043101  042524  020122  DH23:   .ASCIZ  /AFTER WRONG PORT SELECTED/
7982  051556  051127  047117  020107
7983  051564  047520  052122  051440
7984  051572  046105  041505  042524
7985  051600  000104
7986  051602  043101  042524  020122  DH24:   .ASCIZ  /AFTER OFFSET COMMAND/
7987  051610  043117  051506  052105
```

```
7988  051616  041440  046517  040515
7989  051624  042116     000
7990  051627     101  052106  051105  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
7991  051634  051440  042505  020113
7992  051642  047503  046515  047101
7993  051650  000104
7994  051652  043101  042524  020122  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
7995  051660  042522  042101  042040
7996  051666  052101  020101  047503
7997  051674  046515  047101  000104
7998  051702  043101  042524  020122  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
7999  051710  051127  052111  020105
8000  051716  040504  040524  041440
8001  051724  046517  040515  042116
8002  051732     000
8003  051733     101  052106  051105  DH28:   .ASCIZ  /AFTER BOTH PORTS DESELECTED/
8004  051740  041040  052117  020110
8005  051746  047520  052122  020123
8006  051754  042504  042523  042514
8007  051762  052103  042105     000
8008  051767     101  052106  051105  DH29:   .ASCIZ  /AFTER CORRECT PORT SELECTED/
8009  051774  041440  051117  042522
8010  052002  052103  050040  051117
8011  052010  020124  042523  042514
8012  052016  052103  042105     000
8013  052023     101  052106  051105  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
8014  052030  051040  040505  020104
8015  052036  042510  042101  051105
8016  052044  041440  046517  040515
8017  052052  042116     000
8018  052055     101  052106  051105  DH31:   .ASCIZ  /AFTER DRIVE MANUALLY LOADED/
8019  052062  042040  044522  042526
8020  052070  046440  047101  040525
8021  052076  046114  020131  047514
8022  052104  042101  042105     000
8023  052111     101  052106  051105  DH32:   .ASCIZ  /AFTER WRITE CHECK COMMAND/
8024  052116  053440  044522  042524
8025  052124  041440  042510  045503
8026  052132  041440  046517  040515
8027  052140  042116     000
8028  052143     104  051125  047111  DH33:   .ASCIZ  /DURING SEEK COMMAND/
8029  052150  020107  042523  045505
8030  052156  041440  046517  040515
8031  052164  042116     000
8032  052167     101  052106  051105  DH34:   .ASCIZ  /AFTER START SPINDLE COMMAND/
8033  052174  051440  040524  052122
8034  052202  051440  044520  042116
8035  052210  042514  041440  046517
8036  052216  040515  042116     000
8037  052223     101  052106  051105  DH35:   .ASCIZ  /AFTER MANUALLY UNLOADING/
8038  052230  046440  047101  040525
8039  052236  046114  020131  047125
8040  052244  047514  042101  047111
8041  052252  000107
8042  052254  045522  051115  020062  DH36:   .ASCIZ  /RKMR2   RKMR3   RKCS1   RKCS2  FROM SECT  TO SECT/
8043  052262  020040  045522  051115
```

```
8044  052270  020063  020040  045522
8045  052276  051503  020061  020040
8046  052304  045522  051503  020062
8047  052312  043040  047522  020115
8048  052320  042523  052103  020040
8049  052326  047524  051440  041505
8050  052334  000124
8051  052336  043101  042524  020122  DH37:   .ASCIZ  /AFTER TIMEOUT TO POWER DOWN/
8052  052344  044524  042515  052517
8053  052352  020124  047524  050040
8054  052360  053517  051105  042040
8055  052366  053517  000116
8056  052372  043101  042524  020122  DH38:   .ASCIZ  /AFTER AC POWERED UP/
8057  052400  041501  050040  053517
8058  052406  051105  042105  052440
8059  052414  000120
8060  052416  043101  042524  020122  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
8061  052424  051127  052111  020105
8062  052432  042510  042101  051105
8063  052440  041440  046517  040515
8064  052446  042116     000
8065  052451     122  046513  031122  DH40:   .ASCIZ  /RKMR2  RKMR3   RKDA    WORD#   HEADER WAS  SHOULD BE/
8066  052456  051011  046513  031522
8067  052464  051011  042113  004501
8068  052472  047527  042122  004443
8069  052500  042510  042101  051105
8070  052506  053440  051501  020040
8071  052514  044123  052517  042114
8072  052522  041040  000105
8073  052526  052504  044522  043516  DH41:   .ASCIZ  /DURING RECAL COMMAND/
8074  052534  051040  041505  046101
8075  052542  041440  046517  040515
8076  052550  042116     000
8077  052553     117  020116  042523  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
8078  052560  052103  051117  020123
8079  052566  026060  026062  026064
8080  052574  020066  051117  034040
8081  052602  020040  054503  020114
8082  052610  030464  020060  051124
8083  052616  041501  020113  000062
8084  052624  047117  051440  041505  DH43:   .ASCIZ  /ON SECTORS 1,3,5,7 OR 9  CYL 410 TRACK2/
8085  052632  047524  051522  030440
8086  052640  031454  032454  033454
8087  052646  047440  020122  020071
8088  052654  041440  046131  032040
8089  052662  030061  052040  040522
8090  052670  045503  000062
8091  052674  047506  046522  052101  DH44:   .ASCIZ  /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8092  052702  023040  040440  046114
8093  052710  051040  040505  026504
8094  052716  051127  052111  020105
8095  052724  042524  052123  020123
8096  052732  044527  046114  041040
8097  052740  020105  054502  040520
8098  052746  051523  042105     000
8099  052753     101  052106  051105  DH45:   .ASCIZ  /AFTER SEEK WITH VOLUME VALID=0/
```

```
8100  052760  051440  042505  020113
8101  052766  044527  044124  053040
8102  052774  046117  046525  020105
8103  053002  040526  044514  036504
8104  053010  000060
8105  053012  043101  042524  020122  DH46:   .ASCIZ  /AFTER WRITE DATA WITH VOLUME VALID=0/
8106  053020  051127  052111  020105
8107  053026  040504  040524  053440
8108  053034  052111  020110  047526
8109  053042  052514  042515  053040
8110  053050  046101  042111  030075
8111  053056     000
8112  053057     101  052106  051105  DH47:   .ASCIZ  /AFTER SEEK COMMAND WITH MOVEMENT/
8113  053064  051440  042505  020113
8114  053072  047503  046515  047101
8115  053100  020104  044527  044124
8116  053106  046440  053117  046505
8117  053114  047105  000124
8118  053120  043101  042524  020122  DH48:   .ASCIZ  /AFTER WRITE LOCK SWITCH DISABLED/
8119  053126  051127  052111  020105
8120  053134  047514  045503  051440
8121  053142  044527  041524  020110
8122  053150  044504  040523  046102
8123  053156  042105     000
8124  053161     101  052106  051105  DH49:   .ASCIZ  /AFTER WRITE LOCK SWITCH ENABLED/
8125  053166  053440  044522  042524
8126  053174  046040  041517  020113
8127  053202  053523  052111  044103
8128  053210  042440  040516  046102
8129  053216  042105     000
8130  053221     101  052106  051105  DH50:   .ASCIZ  /AFTER WRITING WITH WRITE LOCK ENABLED/
8131  053226  053440  044522  044524
8132  053234  043516  053440  052111
8133  053242  020110  051127  052111
8134  053250  020105  047514  045503
8135  053256  042440  040516  046102
8136  053264  042105     000
8137  053267     101  052106  051105  DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
8138  053274  051440  042505  020113
8139  053302  047524  051440  046105
8140  053310  020106  047503  046515
8141  053316  047101  000104
8142  053322  044527  044124  044440  DH52:   .ASCIZ  /WITH INTENTIONAL MISCOMPARE/
8143  053330  052116  047105  044524
8144  053336  047117  046101  046440
8145  053344  051511  047503  050115
8146  053352  051101  000105
8147  053356  052504  044522  043516  DH53:   .ASCIZ  /DURING OFFSET COMMAND/
8148  053364  047440  043106  042523
8149  053372  020124  047503  046515
8150  053400  047101  000104
8151  053404  052101  046440  054101  DH54:   .ASCIZ  /AT MAX POSITIVE OFFSET/
8152  053412  050040  051517  052111
8153  053420  053111  020105  043117
8154  053426  051506  052105     000
8155  053433     101  020124  040515  DH55:   .ASCIZ  /AT MAX NEGATIVE OFFSET/
```

```
8156  053440  020130  042516  040507
8157  053446  044524  042526  047440
8158  053454  043106  042523  000124
8159  053462  045522  051115  004462  DH56:   .ASCIZ  /RKMR2  RKMR3   RKDC    CYL #   HEADER WORD 0/
8160  053470  045522  051115  004463
8161  053476  045522  041504  041411
8162  053504  046131  021440  044011
8163  053512  040505  042504  020122
8164  053520  047527  042122  030040
8165  053526     000
8166  053527     101  052106  051105  DH57:   .ASCIZ  /AFTER WRITE COMMAND WITH OFFSET/
8167  053534  053440  044522  042524
8168  053542  041440  046517  040515
8169  053550  042116  053440  052111
8170  053556  020110  043117  051506
8171  053564  052105     000
8172  053567     104  052101  020101  DH58:   .ASCIZ  /DATA WAS       SHOULD BE/
8173  053574  040527  020123  051411
8174  053602  047510  046125  020104
8175  053610  042502     000
8176  053613     101  052106  051105  DH59:   .ASCIZ  /AFTER DRIVE SELECT COMMAND WITH EVEN PARITY/
8177  053620  042040  044522  042526
8178  053626  051440  046105  041505
8179  053634  020124  047503  046515
8180  053642  047101  020104  044527
8181  053650  044124  042440  042526
8182  053656  020116  040520  044522
8183  053664  054524     000
8184  053667     101  052106  051105  DH60:   .ASCIZ  /AFTER WRITE LOCK ENABLED DURING CONTINUOUS WRITING/
8185  053674  053440  044522  042524
8186  053702  046040  041517  020113
8187  053710  047105  041101  042514
8188  053716  020104  052504  044522
8189  053724  043516  041440  047117
8190  053732  044524  052516  052517
8191  053740  020123  051127  052111
8192  053746  047111  000107
8193  053752  045522  051115  004462  DH61:   .ASCII  /RKMR2  RKMR3   RKCS1   RKCS2   RKDA    WORD    EXPECTED/
8194  053760  045522  051115  004463
8195  053766  045522  051503  004461
8196  053774  045522  051503  004462
8197  054002  045522  040504  053411
8198  054010  051117  004504  054105
8199  054016  042520  052103  042105
8200  054024  005015  004411  004411          .ASCIZ  <CR><LF>/                       @ WRL   WAS     WORD/
8201  054032  020100  051127  004514
8202  054040  040527  004523  047527
8203  054046  042122     000
8204  054051     122  046513  031122  DH62:   .ASCII  /RKMR2  RKMR3   RKCS1   RKCS2   RKDA    WORD    EXPECTED/
8205  054056  051011  046513  031522
8206  054064  051011  041513  030523
8207  054072  051011  041513  031123
8208  054100  051011  042113  004501
8209  054106  047527  042122  042411
8210  054114  050130  041505  042524
8211  054122     104
```

```
8212  054123  000015  004412  004411          .ASCII  <CR><LF>/                                  BEFORE  WAS     WORD/
8213  054130  041011  043105  051117
8214  054136  004505  040527  004523
8215  054144  047527  042122
8216  054150  005015  004411  004411          .ASCIZ  <CR><LF>/                                  WRL/
8217  054156  051127  000114
8218  054162  043101  042524  020122  DH63:   .ASCIZ  /AFTER WRITE LOCK ENABLED FROM AC FAILURE/
8219  054170  051127  052111  020105
8220  054176  047514  045503  042440
8221  054204  040516  046102  042105
8222  054212  043040  047522  020115
8223  054220  041501  043040  044501
8224  054226  052514  042522  000000
8225  054233  000101  052106  051105  DH64:   .ASCIZ  /AFTER MDS DETECTED IN RKCS2/
8226  054240  046440  051504  042040
8227  054246  052105  041505  042524
8228  054254  020104  047111  051040
8229  054262  041513  031123  000000
8230  054267  000101  052106  051105  DH65:   .ASCIZ  /AFTER SEARCHING ALL DRIVES PRESENT/
8231  054274  051440  040505  041522
8232  054302  044510  043516  040440
8233  054310  046114  042040  044522
8234  054316  042526  020123  051120
8235  054324  051505  047105  000124
8236  054332  042524  052123  047040  DH66:   .ASCIZ  /TEST NO.       TRAP PC/
8237  054340  027117  052011  040522
8238  054346  020120  041520  000000
8239  054353  000101  052106  051105  DH67:   .ASCIZ  /AFTER TIMEOUT TO ENABLE WRITE LOCK/
8240  054360  052040  046511  047505
8241  054366  052125  052040  020117
8242  054374  047105  041101  042514
8243  054402  053440  044522  042524
8244  054410  046040  041517  000113
8245  054416  045522  051115  004462  DH68:   .ASCIZ  /RKMR2  RKMR3   RKCS1   RKCS2   RKDC    RKDA/
8246  054424  045522  051115  004463
8247  054432  045522  051503  004461
8248  054440  045522  051503  004462
8249  054446  045522  041504  051011
8250  054454  042113  000101
8251
8252                                    .SBTTL  ERROR OUTPUT DATA
8253
8254                                    .EVEN
8255  054460  001210  001116  005426  DT1:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HCS1,HCS2,SBMR2,SBMR3
8256  054466  005430  005414  005412
8257  054474  005400  005402  005436
8258  054502  005440
8259  054504  001210  001116  005426  DT2:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HCS1,HCS2,HASOF
8260  054512  005430  005414  005412
8261  054520  005400  005402  005416
8262  054526  001210  001116  005426  DT3:    $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,$TMP3,WD1,WD2
8263  054534  005430  005400  005402
8264  054542  001166  001476  001500
8265  054550  001210  001116  005426  DT4:    $TESTN,$ERRPC,HMR2,HMR3,HDC,FRCYL,TOCYL,CALDIF
8266  054556  005430  005420  001360
8267  054564  001362  001370
```

```
8268  054570  001210  001116  005426  DT5:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HDA,HCS1,HCS2,SBMR2,SBMR3
8269  054576  005430  005414  005412
8270  054604  005410  005400  005402
8271  054612  005436  005440
8272  054616  001210  001350          DT6:    $TESTN,TRAPPC
8273  054622  001210  001116  005426  DT7:    $TESTN,$ERRPC,HMR2,HMR3,HDA,WDCNT,HDWD,TEMP1
8274  054630  005430  005410  001514
8275  054636  001532  005442
8276  054642  001210  001116  005426  DT8:    $TESTN,$ERRPC,HMR2,HMR3,HDC,TOCYL,FRCYL,CALDIF
8277  054650  005430  005420  001362
8278  054656  001360  001370
8279  054662  001210  001116  005426  DT9:    $TESTN,$ERRPC,HMR2,HMR3,HDC,TOCYL,RHTAB
8280  054670  005430  005420  001362
8281  054676  001746
8282  054700  001210  001116  005426  DT10:   $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,HDC,HDA
8283  054706  005430  005400  005402
8284  054714  005420  005410
8285
8286                                  .SBTTL  ERROR DATA FORMATS
8287
8288  054720  000002                  DF1:    2
8289  054722     002     000                  .BYTE 2,0
8290  054724  050404                          DH2
8291  054726     006     000                  .BYTE 6,0
8292
8293  054730  000003                  DF2:    3
8294  054732     002     000                  .BYTE 2,0
8295  054734  050404                          DH2
8296  054736     006     000                  .BYTE 6,0
8297  054740  050446                          DH3
8298  054742     001     000                  .BYTE 1,0
8299
8300  054744  000003                  DF3:    3
8301  054746     000     000                  .BYTE 0,0
8302  054750  050367                          DH1
8303  054752     002     000                  .BYTE 2,0
8304  054754  053752                          DH61
8305  054756     007     000                  .BYTE 7,0
8306
8307  054760  000003                  DF4:    3
8308  054762     000     000                  .BYTE 0,0
8309  054764  050367                          DH1
8310  054766     002     000                  .BYTE 2,0
8311  054770  054051                          DH62
8312  054772     007     000                  .BYTE 7,0
8313
8314  054774  000001                  DF5:    1
8315  054776     002     000                  .BYTE   2,0
8316
8317  055000  000003                  DF6:    3
8318  055002     000     000                  .BYTE 0,0
8319  055004  050367                          DH1
8320  055006     002     000                  .BYTE 2,0
8321  055010  050536                          DH6
8322  055012     006     000                  .BYTE 6,0
8323
```

```
8324  055014  000002              DF7:    2
8325  055016     002  000                 .BYTE   2,0
8326  055020  054416                      DH68
8327  055022     006  000                 .BYTE   6,0
8328  055024  000004              DF8:    4
8329  055026     000  000                 .BYTE 0,0
8330  055030  050367                      DH1
8331  055032     002  000                 .BYTE 2,0
8332  055034  050404                      DH2
8333  055036     006  000                 .BYTE 6,0
8334  055040  050455                      DH4
8335  055042     001  000                 .BYTE 1,0
8336
8337  055044  000004              DF9:    4
8338  055046     000  000                 .BYTE 0,0
8339  055050  050367                      DH1
8340  055052     002  000                 .BYTE 2,0
8341  055054  050404                      DH2
8342  055056     006  000                 .BYTE 6,0
8343  055060  050455                      DH4
8344  055062     002  000                 .BYTE 2,0
8345
8346  055064  000003              DF10:   3
8347  055066     000  000                 .BYTE 0,0
8348  055070  050367                      DH1
8349  055072     002  000                 .BYTE 2,0
8350  055074  050404                      DH2
8351  055076     006  000                 .BYTE 6,0
8352
8353  055100  000004              DF11:   4
8354  055102     000  000                 .BYTE 0,0
8355  055104  050367                      DH1
8356  055106     002  000                 .BYTE 2,0
8357  055110  050404                      DH2
8358  055112     006  000                 .BYTE 6,0
8359  055114  050446                      DH3
8360  055116     001  000                 .BYTE 1,0
8361
8362  055120  000002              DF14:   2
8363  055122     002  000                 .BYTE   2,0
8364  055124  052451                      DH40
8365  055126     006  000                 .BYTE   6,0
8366
8367
8368  055130  000003              DF15:   3
8369  055132     000  000                 .BYTE   0,0
8370  055134  050367                      DH1
8371  055136     002  000                 .BYTE   2,0
8372  055140  050620                      DH7
8373  055142     007  000                 .BYTE   7,0
8374
8375
8376  055144  000004              DF17:   4
8377  055146     000  000                 .BYTE   0,0
8378  055150  052674                      DH44
8379  055152     000  000                 .BYTE   0,0
```

```
8380  055154  050367                        DH1
8381  055156  002     000                   .BYTE   2,0
8382  055160  050404                        DH2
8383  055162  006     000                   .BYTE   6,0
8384  055164  000003                DF20:   3
8385  055166  000     000                   .BYTE   0,0
8386  055170  050367                        DH1
8387  055172  002     000                   .BYTE   2,0
8388  055174  053462                        DH56
8389  055176  005     000                   .BYTE   5,0
8390                                ;;*****************************************************************
8391                                .SBTTL  TYPE ERROR ROUTINE
8392                                ;*ENTRY JSR PC,TYP ERR
8393                                ;*RETURN RTS PC
8394                                ;*
8395                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
8396                                ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
8397                                ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
8398                                ;*THE ERROR.
8399                                ;;*****************************************************************
8400  055200  104413               TYPERR: SAVREG
8401  055202  032777  020000  123730       BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUTS?
8402  055210  001107                       BNE     20$             ;YES-BRANCH
8403  055212  113700  001114               MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
8404  055216  042700  177400               BIC     #177400,R0      ;CLEAR SIGN EXTENSION
8405  055222  005300                       DEC     R0              ;FORM INDEX FOR ERROR TABLE
8406  055224  006300                       ASL     R0
8407  055226  006300                       ASL     R0
8408  055230  006300                       ASL     R0
8409  055232  062700  005516         1$:   ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
8410  055236  012037  055252               MOV     (R0)+,2$        ;GET EM POINTER
8411  055242  001404                       BEQ     3$              ;BRANCH IF THERE ISN'T ONE
8412  055244  104401  001201               TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
8413  055250  104401                       TYPE                    ;TYPE ERROR MESSAGE (EM)
8414  055252  000000               2$:     .WORD   0               ;EM POINTER GOES HERE
8415  055254  012037  055270         3$:   MOV     (R0)+,4$        ;GET DH POINTER
8416  055260  001404                       BEQ     5$              ;BRANCH IF THERE ISN'T ONE
8417  055262  104401  001201               TYPE    ,$CRLF          ;TYPE CR-LF
8418  055266  104401                       TYPE                    ;TYPE DATA HEADER
8419  055270  000000               4$:     .WORD   0               ;DH POINTER GOES HERE
8420  055272  012001               5$:     MOV     (R0)+,R1        ;GET DT POINTER
8421  055274  001455                       BEQ     20$             ;BRANCH IF THERE ARE NONE
8422  055276  005004                       CLR     R4              ;SET INDENT SWITCH
8423  055300  012000                       MOV     (R0)+,R0        ;GET DF POINTER
8424  055302  012002                       MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
8425  055304  001446                       BEQ     17$             ;DH NUM IS 0-BRANCH
8426  055306  005104                       COM     R4              ;NO INDENT
8427  055310  104401  001201               TYPE    ,$CRLF
8428  055314  112003               10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
8429  055316  105720                       TSTB    (R0)+           ;BUMP PAST FORMAT WORD
8430  055320  005703                       TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
8431  055322  001407                       BEQ     14$             ;NO - SKIP DATA PRINT
8432  055324  013146               11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
8433  055326  104402                       TYPOC                   ;TYPE IT
8434  055330  005303                       DEC     R3              ;MORE DATA WORDS
8435  055332  001403                       BEQ     14$             ;NO-BRANCH
```

```
8436  055334  104401  055464          TYPE    ,SPACE2         ;TYPE SEPARATORS
8437  055340  000771                  BR      11$             ;LOOP
8438  055342  005302          14$:    DEC     R2              ;MORE DH'S?
8439  055344  003431                  BLE     20$             ;NO-BRANCH
8440  055346  104401  001201          TYPE    ,$CRLF
8441  055352  005760  000002          TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
8442  055356  001404                  BEQ     15$             ;YES-BRANCH BYPASS INDENT
8443  055360  005104                  COM     R4              ;INDENT?
8444  055362  001002                  BNE     15$             ;NO-BRANCH
8445  055364  104401  055464          TYPE    ,SPACE2         ;YES-TYPE SPACES
8446  055370  012037  055376  15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
8447  055374  104401                  TYPE                    ;TYPE DH
8448  055376  000000          16$:    .WORD   0               ;DH POINTER GOES HERE
8449  055400  105710                  TSTB    (R0)            ;TYPE A DT?
8450  055402  001003                  BNE     21$             ;YES-BRANCH
8451  055404  062700  000002          ADD     #2,R0           ;INCREMENT DF POINTER
8452  055410  000754                  BR      14$             ;SEE IF END OF DF BLOCK
8453  055412  104401  001201  21$:    TYPE    ,$CRLF
8454  055416  005704                  TST     R4              ;INDENT?
8455  055420  001335                  BNE     10$             ;NO-BRANCH
8456  055422  104401  055464  17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES .
8457  055426  000732                  BR      10$             ;LOOP
8458  055430  104414          20$:    RESREG
8459  055432  032777  010000  123500  BIT     #SW12,@SWR      ;SEE IF EXIT AFTER 20 ERRORS
8460  055440  001410                  BEQ     25$             ;BR IF NO
8461  055442  023727  001103  000024  CMP     $ERFLG,#20.     ;ELSE SEE IF HAVE 20 ERRORS
8462  055450  001004                  BNE     25$             ;BR IF NO
8463  055452  012706  001100          MOV     #STACK,SP       ;ELSE RESTORE STACK
8464  055456  000137  023226          JMP     ENDRV           ;AND BYPASS DRIVE
8465  055462  000207          25$:    RTS     PC
8466  055464  020040    000   SPACE2: .ASCIZ/  /              ;2 SPACES
8467                          ;  ODT-11 -- V005A
8468
8469                          ; DEC-11-UODPA-A-LA
8470
8471                          ; COPYRIGHT 1969,1970,1972
8472                          ; DIGITAL EQUIPMENT CORPORATION
8473                          ; MAYNARD, MASSACHUSETTS 01754
8474                                  .ENABL  ABS,AMA
8475          055470                  .EVEN
8476          055550                  .=.+60
8477          000000          R0      =       %0      ; REGISTER
8478          000001          R1      =       %1      ;  NAMING
8479          000002          R2      =       %2      ;   CONVENTIONS
8480          000003          R3      =       %3
8481          000004          R4      =       %4
8482          000005          R5      =       %5
8483          000006          SP      =       %6
8484          000007          PC      =       %7
8485          177776          ST      =       177776          ;STATUS REGISTER
8486                          ;
8487          000014          0.TVEC  =       14              ;TRT VECTOR LOCATION
8488          000340          0.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
8489          000020          0.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
8490          000003          TRT     =       000003          ;TRT INSTRUCTION
8491          000006          RTT     =       000006          ;RTT INSTRUCTION
```

```
8492                                    ;
8493                                    ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
8494                                    ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
8495                                    ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
8496                                    ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
8497                                    ;
8498          177562                    O.RDB   =       177562  ;R DATA BUFFER
8499          177560                    O.RCSR  =       177560  ;R C/SR
8500          177566                    O.TDB   =       177566  ;T DATA BUFFER
8501          177564                    O.TCSR  =       177564  ;T C/SR
8502                                    ;
8503                                    ;
8504                                    ; INITIALIZE ODT
8505                                    ;  USE O.ODT FOR A NORMAL ENTRY
8506                                    ;  USE O.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
8507                                    ;  USE O.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
8508                                    ;
8509  055550  000413                    O.ODT:  BR      O.STRT          ;NORMAL ENTRY
8510  055552  000417                            BR      O.RST           ;RESTART
8511  055554  013737  177776  055530  O.ENTR: MOV     ST,O.UST        ;RE-ENTER  --  SAVE STATUS
8512  055562  013737  000016  177776          MOV     O.TVEC+2,ST     ;SET UP LOCAL STATUS
8513  055570  010737  055526                  MOV     PC,O.UPC        ;FAKE THE PC
8514  055574  000137  056726                  JMP     O.BK1
8515                                    ;
8516  055600  012706  055510            O.STRT: MOV     #O.URO,SP       ;SET UP STACK
8517  055604  010637  055524                    MOV     SP,O.USP        ;FAKE THE SAVED STACK
8518  055610  000414                            BR      O.RST1          ;CLEAR BREAKPOINT TABLES
8519  055612  004037  057134            O.RST:  JSR     0,O.SVR         ;SAVE REGISTERS
8520  055616  013777  055546  177716          MOV     O.UIN,@O.ADR1   ;REMOVE THE BREAKPOINT
8521  055624  113704  055532                  MOVB    O.PRI,R4        ;GET ODT PRIORITY
8522  055630  106004                          RORB    R4              ;SHIFT
8523  055632  106004                          RORB    R4              ;  INTO
8524  055634  106004                          RORB    R4              ;    POSITION
8525  055636  110437  177776                  MOVB    R4,ST           ;STORE IN STATUS
8526  055642  000127                    O.RST1: JMP     (PC)+
8527  055644  000403                            BR      O.45
8528  055646  012737  000002  056636          MOV     #RTI,O.RTIT     ;SET TO RTI IF 11/20 OR /05
8529  055654  105037  057555            O.45:   CLRB    O.P             ;DISALLOW PROCEED
8530  055660  012737  000340  000016          MOV     #O.STM,O.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
8531  055666  012737  056716  000014          MOV     #O.BRK,O.TVEC   ;PC TO TRT VECTOR
8532  055674  000447                          BR      O.RALL          ;CLEAR BREAKPOINT TABLES
8533                                    ;
8534                                    ; SPECIAL NAME HANDLER
8535                                    ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URO
8536                                    ;
8537  055676  004537  057356            O.REGT: JSR     5,O.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
8538  055702  012704  057601                    MOV     #O.TL,R4        ;TABLE START ADDRESS
8539  055706  120024                    O.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
8540  055710  001413                            BEQ     O.SP            ;JUMP IF YES
8541  055712  022704  057607                  CMP     #O.TL+O.LG,R4   ;IS THE SEARCH DONE?
8542  055716  101373                          BHI     O.RSP           ;BRANCH IF NOT
8543  055720  042700  177770                  BIC     #177770,R0      ;MASK OFF OCTAL
8544  055724  010004                          MOV     R0,R4
8545  055726  006304                    O.SP1:  ASL     R4
8546  055730  062704  055510                  ADD     #O.URO,R4       ;GENERATE ADDRESS
8547  055734  005202                          INC     R2              ;SET FOUND FLAG
```

```
8548  055736  000444                        BR      0.SCAN          ;GO FIND NEXT CHARACTER
8549  055740  162704  057572        0.SP:   SUB     #0.TL-7,R4      ;CORRECT CONSTANT
8550  055744  000770                        BR      0.SP1
8551                                ;
8552                                ; ← HANDLER - OPEN INDEXED ON THE PC
8553                                ;
8554  055746  004737  057502        0.ORPC: JSR     PC,0.TCLS
8555  055752  010502                        MOV     R5,R2           ;CURRENT ADDRESS IN R2
8556  055754  061202                        ADD     @R2,R2          ;COMPUTE
8557  055756  006202                        ASR     R2              ;MOVE ONE BIT TO CARRY
8558  055760  103421                        BCS     0.ERR           ;ERROR IF ODD NUMBER
8559  055762  006302                        ASL     R2              ;RESTORE WORD
8560  055764  005722                        TST     (R2)+           ; AND INCREMENT BY TWO
8561  055766  010205                        MOV     R2,R5           ;UPDATE CAD
8562  055770  000137  056242                JMP     0.OP2           ;GO FINISH UP
8563                                ;
8564                                ; B HANDLER - SET AND REMOVE BREAKPOINTS
8565                                ;
8566  055774  005702        0.BKPT: TST     R2              ;IF NO NUMBER TYPED
8567  055776  001406                        BEQ     0.RALL          ;  REMOVE BREAKPOINT
8568  056000  006204                        ASR     R4              ;CHECK IF ODD
8569  056002  103410                        BCS     0.ERR           ;JUMP IF ODD
8570  056004  006304                        ASL     R4              ;RESTORE ONE BIT
8571  056006  010437  055542                MOV     R4,0.ADR1       ;SET A BREAKPOINT
8572  056012  000412                        BR      0.DCD
8573  056014  012737  057616  055542  0.RALL: MOV     #0.TRTC,0.ADR1  ;CLEAR BREAKPOINT
8574  056022  000406                        BR      0.DCD
8575                                ;
8576                                ; COMMAND DECODER - ODT11
8577                                ;
8578                                ;   REGISTERS R0-R4 MAY BE USED,
8579                                ;    REGISTER R5 WILL BE CONSIDERED SAFE
8580                                ;
8581  056024  052705  000001        0.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
8582  056030  012700  000077                MOV     #'?,R0          ;  ? TO BE TYPED
8583  056034  004537  057434                JSR     5,0.FTYP        ; OUTPUT ?
8584  056040  004537  057534        0.DCD:  JSR     5,0.CRLS        ;TYPE  <CR><LF>*
8585  056044  005004        0.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
8586  056046  005002                CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
8587  056050  004537  057356        0.SCAN: JSR     5,0.GET         ;GET A CHAR, RETURN IN R0
8588  056054  022700  000060                CMP     #'0,R0          ;COMPARE WITH ASCII 0
8589  056060  101013                        BHI     0.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
8590  056062  022700  000067                CMP     #'7,R0          ;COMPARE WITH ASCII 7
8591  056066  103410                        BLO     0.CLGL          ;CHECK LEGALITY IF NOT OCTAL
8592  056070  042700  177770                BIC     #177770,R0      ;CONVERT TO BCD
8593  056074  006304                        ASL     R4              ; MAKE ROOM
8594  056076  006304                        ASL     R4              ;  IN
8595  056100  006304                        ASL     R4              ;    R4
8596  056102  060004                        ADD     R0,R4           ;PACK THREE BITS IN R4
8597  056104  005202                        INC     R2              ;R2 HAS NUMERIC FLAG
8598  056106  000760                        BR      0.SCAN          ;   AND TRY AGAIN
8599  056110  005001        0.CLGL: CLR     R1              ;CLEAR INDEX
8600  056112  120061  057565  0.LGL1: CMPB    R0,0.LGCH(R1)   ;DO THE CODES MATCH?
8601  056116  001405                        BEQ     0.LGL2          ;JUMP IF YES
8602  056120  005201                        INC     R1              ; SET INDEX FOR NEXT SEARCH
8603  056122  020127  000014                CMP     R1,#0.CLGT      ;IS THE SEARCH DONE?
```

```
 8604  056126  103336                    BHIS    0.ERR           ;    OOPS!
 8605  056130  000770                    BR      0.LGL1          ;RE-LOOP
 8606  056132  006301            0.LGL2: ASL     R1              ;MULTIPLY BY TWO
 8607  056134  000171  056140            JMP     @0.LGDR(R1)     ;GO TO PROPER ROUTINE
```

```
 8608
 8609  056140  056170                  O.LGDR: O.WRD   ;  /    OPEN WORD
 8610  056142  056222                          O.CRET  ;  CARRIAGE RETURN    CLOSE
 8611  056144  055676                          O.REGT  ;  $  REGISTER OPS
 8612  056146  056532                          O.GO    ;  G  GO TO ADDRESS K
 8613  056150  056234                          O.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
 8614  056152  055746                          O.ORPC  ;  ←  OPEN RELATED, INDEX - PC
 8615  056154  056266                          O.BACK  ;  ↑  OPEN PREVIOUS
 8616  056156  056276                          O.OFST  ;  O  OFFSET
 8617  056160  056354                          O.WSCH  ;  W  SEARCH WORD
 8618  056162  056350                          O.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
 8619  056164  055774                          O.BKPT  ;  B  BREAKPOINTS
 8620  056166  056640                          O.PROC  ;  P  PROCEED
 8621          000030                  O.LGL   =       .-O.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS
 8622                                  ;
 8623                                  ; PROCESS / - OPEN WORD
 8624                                  ;
 8625  056170  005702                  O.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
 8626  056172  001410                          BEQ     O.WRDA          ;SKIP OTHERWISE
 8627  056174  010405                          MOV     R4,R5           ;   PUT VALUE IN CAD
 8628  056176  006205                  O.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
 8629  056200  103711                  O.ERR2: BCS     O.ERR           ;JUMP IF ODD ADDRESS
 8630  056202  006305                          ASL     R5              ;RESTORE THE CARRY BIT
 8631  056204  011500                          MOV     @R5,R0          ;GET CONTENTS OF WORD
 8632  056206  004537  057272                JSR     5,O.CADV        ;GO GET AND TYPE OUT @CAD
 8633  056212  000714                          BR      O.DCD1          ;GO BACK TO DECODER
 8634  056214  042705  000001          O.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
 8635  056220  000766                          BR      O.WRD1          ;GO BACK TO MAIN-LINE
 8636                                  ;
 8637                                  ; PROCESS CARRIAGE RETURN
 8638                                  ;
 8639  056222  004737  057502          O.CRET: JSR     PC,O.TCLS       ;CLOSE LOCATION
 8640  056226  052705  000001                  BIS     #1,R5           ;CLOSE EVERYTHING
 8641  056232  000702                          BR      O.DCD           ;RETURN TO DECODER
 8642                                  ;
 8643                                  ; PROCESS <LF>, OPEN NEXT WORD
 8644                                  ;
 8645  056234  004737  057502          O.OP1:  JSR     PC,O.TCLS       ;CLOSE PRESENT CELL
 8646  056240  005725                          TST     (R5)+           ;GENERATE NEW ADDRESS
 8647  056242  004537  057526          O.OP2:  JSR     5,O.CRLF        ;<CR><LF>
 8648  056246  010500                          MOV     R5,R0           ;NUMBER TO TYPE
 8649  056250  004537  057272                JSR     5,O.CADV        ;  TYPE OUT ADDRESS
 8650  056254  012700  000057                MOV     #'/,R0          ;TYPE A /
 8651  056260  004537  057434                JSR     5,O.FTYP
 8652  056264  000744                          BR      O.WRD1          ;GO PROCESS IT
 8653                                  ;
 8654                                  ; PROCESS ↑, OPEN PREVIOUS WORD
 8655                                  ;
 8656  056266  004737  057502          O.BACK: JSR     PC,O.TCLS
 8657  056272  005745                          TST     -(R5)           ;GENERATE NEW ADDRESS
 8658  056274  000762                          BR      O.OP2           ;GO DO THE REST
 8659                                  ;
 8660                                  ; PROCESS O, COMPUTE OFFSET
 8661                                  ;
 8662  056276  006205                  O.OFST: ASR     R5              ;GET LOW ORDER BIT
 8663  056300  103737                          BCS     O.ERR2          ;ERROR IF CLOSED
```

```
 8664  056302  006305                        ASL     R5              ;RESTORE WORD
 8665  056304  012700  000040                MOV     #' ,R0          ;TYPE ONE BLANK
 8666  056310  004537  057434                JSR     5,0.FTYP        ; AS A SEPARATOR
 8667  056314  160504                        SUB     R5,R4           ;COMPUTE
 8668  056316  005304                        DEC     R4
 8669  056320  005304                        DEC     R4              ;     16 BIT OFFSET
 8670  056322  010400                        MOV     R4,R0           ;TYPE A
 8671  056324  010402                        MOV     R4,R2           ;SAVE R4
 8672  056326  004537  057272                JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
 8673  056332  010200                        MOV     R2,R0
 8674  056334  006200                        ASR     R0              ;DIVIDE BY TWO
 8675  056336  103402                        BCS     0.0F1           ;BRANCH IF ODD
 8676  056340  004537  057272                JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
 8677  056344  000137  056044        0.0F1:  JMP     0.DCD1          ;ALL DONE
 8678                                ;
 8679                                ; SEARCHES - $MSK   HAS THE MASK
 8680                                ;            $MSK+2 HAS THE FWA
 8681                                ;            $MSK+4 HAS THE LWA
```

```
 8682
 8683  056350  005201                  O.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
 8684  056352  000401                          BR      O.WDS
 8685  056354  005001                  O.WSCH: CLR     R1              ;SET WORD SEARCH
 8686  056356  005702                  O.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
 8687  056360  001621                  O.ERR1: BEQ     O.ERR           ;ERROR IF NO OBJECT
 8688  056362  013702  055536                  MOV     O.MSK+2,R2      ;SET ORIGIN
 8689  056366  013705  055534                  MOV     O.MSK,R5        ;SET MASK
 8690  056372  005105                          COM     R5              ;AND COMPLEMENT IT
 8691  056374  020237  055540          O.WDS2: CMP     R2,O.MSK+4      ; IS THE SEARCH ALL DONE?
 8692  056400  101217                          BHI     O.DCD           ;  YES
 8693  056402  011200                          MOV     @R2,R0          ; GET OBJECT
 8694  056404  005701                          TST     R1              ;NO
 8695  056406  001027                          BNE     O.EFF1          ;BRANCH IF EFFECTIVE SEARCH
 8696  056410  010046                          MOV     R0,-(SP)
 8697  056412  010403                          MOV     R4,R3           ;EXCLUSIVE OR
 8698  056414  040400                          BIC     R4,R0           ; IS DONE
 8699  056416  042603                          BIC     (SP)+,R3        ;       IN A VERY
 8700  056420  050003                          BIS     R0,R3           ;    FANCY MANNER HERE
 8701  056422  040503                          BIC     R5,R3           ;AND RESULT WITH MASK
 8702  056424  001016          O.WDS3: BNE     O.WDS4          ;RE-LOOP IF NO MATCH
 8703  056426  010446                          MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
 8704  056430  004537  057526                  JSR     5,O.CRLF        ;TYPE <CR,LF>
 8705  056434  010200                          MOV     R2,R0           ;GET READY TO TYPE
 8706  056436  004537  057272                  JSR     5,O.CADV        ;  TYPE ADDRESS
 8707  056442  012700  000057                  MOV     #'/,R0          ;SLASH TO R0
 8708  056446  004537  057434                  JSR     5,O.FTYP        ;TYPE IT
 8709  056452  011200                          MOV     @R2,R0          ;GET CONTENTS
 8710  056454  004537  057272                  JSR     5,O.CADV        ;TYPE CONTENTS
 8711  056460  012604                          MOV     (SP)+,R4        ; RESTORE R4
 8712  056462  005722          O.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
 8713  056464  000743                          BR      O.WDS2          ;    RETURN
 8714  056466  020004          O.EFF1: CMP     R0,R4           ; IS (X)=K?
 8715  056470  001755                          BEQ     O.WDS3          ;TYPE IF EQUAL
 8716  056472  010003                          MOV     R0,R3           ;(X) TO R3
 8717  056474  060203                          ADD     R2,R3           ;(X)+X
 8718  056476  005203                          INC     R3
 8719  056500  005203                          INC     R3              ;(X)+X+2
 8720  056502  020304                          CMP     R3,R4           ;IS (X)+X+2=K?
 8721  056504  001747                          BEQ     O.WDS3          ;BRANCH IF EQUAL
 8722  056506  042700  177400                  BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
 8723  056512  110000                          MOVB    R0,R0           ;EXTEND SIGN
 8724  056514  000257                          CCC
 8725  056516  006300                          ASL     R0              ;MULTIPLY BY TWO
 8726  056520  005200                          INC     R0              ;ADD TWO
 8727  056522  005200                          INC     R0
 8728  056524  060200                          ADD     R2,R0           ;ADD PC
 8729  056526  020004                          CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
 8730  056530  000735                          BR      O.WDS3
 8731                          ;
 8732                          ; PROCESS G - GO
 8733                          ;
 8734  056532  105037  057555  O.GO:   CLRB    O.P             ;DISALLOW PROCEED
 8735  056536  006204                  ASR     R4              ;CHECK LOW ORDER BIT
 8736  056540  103617                  BCS     O.ERR2          ;ERROR IF ODD NUMBER
 8737  056542  006304                  ASL     R4              ;RESTORE WORD
```

```
8738  056544  010437  055526                  MOV     R4,0.UPC        ;SET UP NEW PC
8739  056550  112737  000340  177776          MOVB    #0.STM,ST       ;SET HIGH PRIORITY
8740  056556  004537  057224                  JSR     5,0.RSTT        ;RESTORE TELETYPE
8741  056562  105037  057554          0.TBIT: CLRB    0.T             ;CLEAR BOTH
8742  056566  042737  000020  055530          BIC     #0.TBT,0.UST    ;  T-BIT FLAGS
8743  056574  017737  176742  055546          MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
8744  056602  013777  057616  176732          MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
8745  056610  012600                  0.G02:  MOV     (SP)+,R0        ;RESTORE
8746  056612  012601                          MOV     (SP)+,R1        ; R0
8747  056614  012602                          MOV     (SP)+,R2        ;   THRU
8748  056616  012603                          MOV     (SP)+,R3
8749  056620  012604                          MOV     (SP)+,R4        ;
8750  056622  012605                          MOV     (SP)+,R5        ;    R5
8751  056624  012606                          MOV     (SP)+,SP        ; AND SP
8752  056626  013746  055530                  MOV     0.UST,-(SP)     ;    AND STATUS
8753  056632  013746  055526                  MOV     0.UPC,-(SP)     ;    AND PC
8754  056636  000006                  0.RTIT: RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
8755                                          ;
8756                                          ; PROCESS P - PROCEED
8757                                          ;   ONLY ALLOWED AFTER A BREAKPOINT
8758                                          ;
8759  056640  105737  057555          0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
8760  056644  001645                          BEQ     0.ERR1          ;NOT LEGAL
8761  056646  105037  057555                  CLRB    0.P             ;CLEAR PROCEED FLAG
8762  056652  005702                          TST     R2              ;WAS COUNT SPECIFIED?
8763  056654  001402                          BEQ     0.PR1           ;NO
8764  056656  010437  055544                  MOV     R4,0.CT         ;YES, PUT AWAY COUNT
8765  056662  112737  000340  177776  0.PR1:  MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
8766  056670  004537  057224                  JSR     5,0.RSTT        ;RESTORE TTY
8767  056674  112737  000340  177776  0.C1:   MOVB    #0.STM,ST       ;SET HIGH PRIORITY
8768  056702  105237  057554                  INCB    0.T             ;SET T-BIT FLAG
8769  056706  052737  000020  055530          BIS     #0.TBT,0.UST    ;SET T-BIT
8770  056714  000735                          BR      0.G02
8771                                          ;
8772                                          ; BREAKPOINT HANDLER
8773                                          ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
8774                                          ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
8775                                          ;  AND GIVES CONTROL TO THE COMMAND DECODER
8776                                          ;
8777  056716  012637  055526          0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
8778  056722  012637  055530                  MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
8779  056726  004037  057134          0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
8780  056732  105737  057554                  TSTB    0.T             ;CHECK FOR T-BIT SET
8781  056736  001311                          BNE     0.TBIT          ;JUMP IF SET
8782  056740  013777  055546  176574          MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
8783  056746  105737  055532                  TSTB    0.PRI           ;CHECK IF PRIORITY
8784  056752  100003                          BPL     0.BK2           ;  IS AS SAME AS USER PGM
8785  056754  113705  055530                  MOVB    0.UST,R5        ;PICK UP USER UST IF SO
8786  056760  000407                          BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
8787  056762  113705  055532          0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
8788  056766  000257                          CCC                     ;CLEAR CARRY
8789  056770  106005                          RORB    R5              ;SHIFT LOW ORDER BITS
8790  056772  106005                          RORB    R5              ;  INTO
8791  056774  106005                          RORB    R5              ;    HIGH ORDER
8792  056776  106005                          RORB    R5              ;      POSITION
8793  057000  110537  177776          0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
```

```
8794  057004  013705  055526                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
8795  057010  005745                          TST     -(R5)           ;SUBTRACT TWO
8796  057012  010537  055526                  MOV     R5,0.UPC        ;FROM THE USER'S PC
8797  057016  020537  055542                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
8798  057022  001417                          BEQ     0.B2            ;JUMP IF FOUND
8799  057024  004537  057172                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
8800  057030  004537  057526                  JSR     5,0.CRLF
8801  057034  012704  057560                  MOV     #0.BD,R4        ;ERROR, NOTHING FOUND
8802  057040  012703  057561                  MOV     #0.BD+1,R3
8803  057044  004537  057420                  JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
8804  057050  010500                          MOV     R5,R0
8805  057052  042737  000020  055530          BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
8806  057060  000420                          BR      0.B3            ; AND CONTINUE
8807  057062  005337  055544          0.B2:   DEC     0.CT
8808  057066  003302                          BGT     0.C1            ;JUMP IF REPEAT
8809  057070  012737  000001  055544          MOV     #1,0.CT         ;RESET COUNT TO 1
8810  057076  105237  057555                  INCB    0.P             ;ALLOW PROCEED
8811  057102  004537  057172                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
8812  057106  012700  000102                  MOV     #'B,R0
8813  057112  004537  057434                  JSR     5,0.FTYP        ;TYPE "B"
8814  057116  013700  055542                  MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
8815  057122  004537  057272          0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
8816  057126  005005                          CLR     R5              ;CLEAR CAD
8817  057130  000137  056040                  JMP     0.DCD           ;GO TO DECODER
8818                                  ;
8819                                  ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
8820                                  ;
8821  057134  012637  057552          0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
8822  057140  010637  055524                  MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
8823  057144  012706  055524                  MOV     #0.USP,SP       ;SET TO INTERNAL STACK
8824  057150  010546                          MOV     R5,-(SP)        ;SAVE
8825  057152  010446                          MOV     R4,-(SP)        ; REGISTERS
8826  057154  010346                          MOV     R3,-(SP)        ;1
8827  057156  010246                          MOV     R2,-(SP)        ; THRU
8828  057160  010146                          MOV     R1,-(SP)        ;     5
8829  057162  013746  057552                  MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
8830  057166  005746                          TST     -(SP)
8831  057170  000200                          RTS     R0
8832                                  ;
8833                                  ; SAVE TELETYPE STATUS
8834                                  ;
8835  057172  113737  177560  057556  0.SVTT: MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
8836  057200  113737  177564  057557          MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
8837  057206  105037  177560                  CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
8838  057212  105037  177564                  CLRB    0.TCSR          ;  BITS IN BOTH C/SR
8839  057216  004537  057526                  JSR     5,0.CRLF        ;TYPE <CR,LF>
8840  057222  000205                          RTS     R5
8841                                  ;
8842                                  ; RESTORE TELETYPE STATUS
8843                                  ;
8844  057224  004537  057526          0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
8845  057230  105737  177564                  TSTB    0.TCSR          ;WAIT READY ON PRINTER
8846  057234  100375                          BPL     .-4
8847  057236  032737  004000  177560          BIT     #4000,0.RCSR    ;CHECK BUSY FLAG ON READER
8848  057244  001403                          BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
8849  057246  105737  177560                  TSTB    0.RCSR          ;WAIT READY
```

```
                                        BPL     .-4             ;     ON READER
057252  100375
057254  113737  057556  177560        0.RSE1: MOVB    0.CSR1,0.RCSR   ;RESTORE
057262  113737  057557  177564                MOVB    0.CSR2,0.TCSR   ;  THE STATUS REGISTERS
057270  000205                                RTS     R5
                                        ;
                                        ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
                                        ; WORD IS IN R0
                                        ;
057272  010246                          0.CADV: MOV     R2,-(SP)        ;SAVE R2
057274  012704  057615                        MOV     #0.BUF+6,R4     ;BUFFER START ADDRESS
057300  012746  000060                        MOV     #'0,-(SP)               ;CONSTANT ASCII 0
057304  010002                          0.SPC:  MOV     R0,R2           ; GET
057306  042702  177770                        BIC     #177770,R2      ;    OCTAL CHARACTER
057312  061602                                ADD     @SP,R2          ;CONVERT TO ASCII
057314  110244                                MOVB    R2,-(R4)        ;STORE IN BUFFER
057316  006200                                ASR     R0              ;SHIFT THIS MESS
057320  006200                                ASR     R0              ; RIGHT
057322  006200                                ASR     R0              ;   THREE WHOLE PLACES
057324  020427  057610                        CMP     R4,#0.BUF+1     ;DONE?
057330  101365                                BHI     0.SPC           ;   NO
057332  042700  177776                        BIC     #177776,R0      ;GET LAST BIT
057336  062600                                ADD     (SP)+,R0                ;CONVERT TO ASCII
057340  110044                                MOVB    R0,-(R4)        ;AND PUT IT AWAY
057342  012703  057615                        MOV     #0.BUF+6,R3     ;LWA
057346  004537  057420                        JSR     5,0.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
057352  012602                                MOV     (SP)+,R2        ;RESTORE R2
057354  000205                                RTS     R5
                                        ;
                                        ; GENERAL CHARACTER INPUT ROUTINE
                                        ; CHARACTER INPUT GOES TO R0
                                        ;
057356  105737  177560                  0.GET:  TSTB    0.RCSR          ;WAIT FOR
057362  100375                                BPL     .-4             ;   INPUT FROM KEYBOARD
057364  113700  177562                        MOVB    0.RDB,R0        ;GET A CHARACTER
057370  004537  057434                        JSR     5,0.FTYP        ;ECHO CHARACTER
057374  042700  177600                        BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
057400  001766                                BEQ     0.GET           ;IGNORE NULLS
057402  122700  000040                        CMPB    #40,R0          ;CHECK FOR SPACES
057406  001763                                BEQ     0.GET           ;IGNORE NULLS
057410  122700  000073                        CMPB    #';,R0          ;CHECK FOR SEMI-COLON
057414  001760                                BEQ     0.GET           ;IGNORE THEM IF FOUND
057416  000205                                RTS     R5
                                        ;
                                        ; GENERAL CHARACTER OUTPUT ROUTINE
                                        ;  ADDRESS OF FIRST BYTE IN R4,
                                        ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
                                        ;
057420  020304                          0.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
057422  103426                                BLO     0.TYP1          ;  EXIT WHEN DONE
057424  112400                                MOVB    (R4)+,R0        ;GET A CHARACTER
057426  004537  057434                        JSR     5,0.FTYP        ;TYPE ONE CHARACTER
057432  000772                                BR      0.TYPE          ;LOOP UNTIL DONE
                                        ;
                                        ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
                                        ;
057434  105737  177564                  0.FTYP: TSTB    0.TCSR          ;CHECK STATUS
```

```
 8906  057440  100375                    BPL     .-4             ;WAIT UNTIL READY
 8907  057442  110037  177566            MOVB    R0,0.TDB        ;TYPE ONE CHARACTER
 8908  057446  120037  000045            CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
 8909  057452  001012                    BNE     0.TYP1          ;NO
 8910  057454  113746  000044            MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
 8911  057460  105737  177564  0.TYP2:   TSTB    0.TCSR
 8912  057464  100375                    BPL     0.TYP2
 8913  057466  105037  177566            CLRB    0.TDB           ;GENERATE NULL FILLER
 8914  057472  105316                    DECB    @SP
 8915  057474  003371                    BGT     0.TYP2
 8916  057476  005726                    TST     (SP)+           ;POP STACK
 8917  057500  000205  0.TYP1:   RTS     R5
 8918                                    ;
 8919                                    ; CLOSE WORD OR BYTE AND EXIT,
 8920                                    ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
 8921                                    ;
 8922  057502  006205  0.TCLS:  ASR     R5              ;GET LOW ORDER BIT
 8923  057504  103405                    BCS     0.TC            ;JUMP IF ALREADY CLOSED
 8924  057506  006305                    ASL     R5
 8925  057510  005702                    TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
 8926  057512  001401                    BEQ     0.CLS1          ;NO CHANGE TO THE OPEN CELL
 8927  057514  010415                    MOV     R4,@R5          ;STORE WORD
 8928  057516  000207  0.CLS1:  RTS     PC
 8929  057520  005746  0.TC:    TST     -(SP)           ;POP EXTRA CELL FROM STACK
 8930  057522  000137  056024            JMP     0.ERR           ;AND SCREAM BLOODY MURDER
 8931                                    ;
 8932                                    ; 0.CRLF - TYPE <CR,LF>
 8933                                    ; 0.CRLS - TYPE <CR,LF>*
 8934                                    ;
 8935  057526  012703  057563  0.CRLF:  MOV     #0.CR+1,R3      ;LWA <CR,LF>
 8936  057532  000402                    BR      0.CRS
 8937  057534  012703  057564  0.CRLS:  MOV     #0.CR+2,R3      ;LWA <CR,LF>*
 8938  057540  012704  057562  0.CRS:   MOV     #0.CR,R4        ;FWA
 8939  057544  004537  057420            JSR     5,0.TYPE        ;TYPE SOMETHING
 8940  057550  000205                    RTS     R5
 8941                                    ;
 8942  057552  000000  0.XXX:   .WORD   0               ;TEMPORARY STORAGE
 8943  057554     000  0.T:     .BYTE   0               ;  T-BIT FLAG
 8944  057555     000  0.P:     .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
 8945                                                    ;               = 1 IF PROCEED ALLOWED
 8946  057556     000  0.CSR1:  .BYTE   0               ;SAVE CELL - R C/SR
 8947  057557     000  0.CSR2:  .BYTE   0               ;SAVE CELL - T C/SR
 8948                                    ;
 8949                                    .EVEN
 8950  057560  042502  0.BD:    .WORD   "BE
 8951                                    ;
 8952  057562     015  0.CR:    .BYTE   015     ;  <CR>
 8953  057563     012                    .BYTE   012     ;  <LF>
 8954  057564     052                    .BYTE   '*      ;  *
 8955                                    ;
 8956  057565     057  0.LGCH:  .BYTE   '/     ;  /
 8957  057566     015                    .BYTE   015     ;  CARRIAGE RETURN
 8958  057567     044                    .BYTE   '$      ;  $
 8959  057570     107                    .BYTE   'G      ;  G
 8960  057571     012                    .BYTE   012     ;  <LF>
 8961  057572     137                    .BYTE   '_      ;  _
```

```
                                        .BYTE   '↑      ;  ↑
                                        .BYTE   'O      ;  O
                                        .BYTE   'W      ;  W
                                        .BYTE   'E      ;  E
                                        .BYTE   'B      ;  B
                                        .BYTE   'P      ;  P
000014                  O.CLGT  =       .-O.LGCH                ;TABLE LENGTH
                        ;
                        O.TL:   .BYTE   'S      ;DO        1
                                .BYTE   'P      ;NOT       2
                                .BYTE   'M      ;CHANGE    3
                                .BYTE   0       ;THE       4
                                .BYTE   0       ;ORDER     5
                                .BYTE   'B      ;HERE      6
000006                  O.LG    =       .-O.TL
                        ;
                        O.BUF:                          ;6 CHAR. BUFFER WITH
057615                  .       =       .+6
                                .BYTE                   ;TRAILING BLANK
                                .EVEN
                        ;
                        O.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
                        ;
                        ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
                        ;
055510                  .       =       O.ODT-40
                        O.URO:  0       ;USER R0
                                0       ;      R1
                                0       ;      R2
                                0       ;      R3
                                0       ;      R4
                                0       ;      R5
                        O.USP:  0       ;USER SP
                        O.UPC:  0       ;USER PC
                        O.UST:  0       ;USER ST
                        O.PRI:  7       ;ODT PRIORITY
                        O.MSK:  0       ;MASK
                                0       ;LOW LIMIT
                                0       ;HIGH LIMIT
                        ;
                        ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
                        ;   UIN = CONTENTS
                        ;
                        O.ADR1: 0
                        O.CT:   0
                        O.UIN:  0
                                .END
```

```
ABASE = 177440          1310    1351    1375#
ACDW1 = 000000          1310    1353
ACDW2 = 000000          1310    1354
ACLO  = 000010          1111#
ACPUOP= 000C00          1310    1325
ACT11   005460          1624#   2770*
ADDW0 = 000000          1310    1355
ADDW1 = 000000          1310    1356
ADDW10= 000000          1310    1365
ADDW11= 000000          1310    1366
ADDW12= 000000          1310    1367
ADDW13= 000000          1310    1368
ADDW14= 000000          1310    1369
ADDW15= 000000          1310    1370
ADDW2 = 000000          1310    1357
ADDW3 = 000000          1310    1358
ADDW4 = 000000          1310    1359
ADDW5 = 000000          1310    1360
ADDW6 = 000000          1310    1361
ADDW7 = 000000          1310    1362
ADDW8 = 000000          1310    1363
ADDW9 = 000000          1310    1364
ADEVCT= 000000          1310    1316
ADEVM = 000000          1310    1352
AENV  = 000000          1310    1321
AENVM = 000000          1310    1322
AFATAL= 000000          1310    1313
AMADR1= 000000          1310    1338
AMADR2= 000000          1310    1342
AMADR3= 000000          1310    1345
AMADR4= 000000          1310    1348
AMAMS1= 000000          1310    1332
AMAMS2= 000000          1310    1340
AMAMS3= 000000          1310    1343
AMAMS4= 000000          1310    1346
AMSGAD= 000000          1310    1318
AMSGLG= 000000          1310    1319
AMSGTY= 000000          1310    1312
AMTYP1= 000000          1310    1333
AMTYP2= 000000          1310    1341
AMTYP3= 000000          1310    1344
AMTYP4= 000000          1310    1347
APASS = 000000          1310    1315
APRIOR= 000000          1310
APTCSU= 000040          5949    6121#
APTENV= 000001          5896    5942    6077    6119#
APTSIZ= 000200          2702    6118#
APTSPO= 000100          5944    6079    6120#
ASWREG= 000000          1310    1323
ATESTN= 000000          1310    1314
ATTN    005370          1487#   3090    4949    4968    4994
AUNIT = 000000          1310    1317
AUSWR = 000000          1310    1324
AVECT1= 000000          1310    1349
AVECT2= 000000          1310    1350
BADINT  027566          5658#
```

F14


```
BADTMO  027500          2848    5623#
BAI   = 000020          1074#   4100    4119    4159    4196    4254    4322    4367    4464    4502    4578    4623
BA16  = 000400          1060#
BA17  = 001000          1061#
BIT0  = 000001          1003#   1057    1089    1108    4248    4314    4361    4496    4570    4617    5559
BIT00 = 000001           993#   1003
BIT01 = 000002           992#   1002
BIT02 = 000004           991#   1001
BIT03 = 000010           990#   1000
BIT04 = 000020           989#    999
BIT05 = 000040           988#    998
BIT06 = 000100           987#    997
BIT07 = 000200           986#    996
BIT08 = 000400           985#    995    5831
BIT09 = 001000           984#    994    5839    5907
BIT1  = 000002          1002#   1090
BIT10 = 002000           983#   1062    1080    1099    1131    1145    1159    1172    1186    5884
BIT11 = 004000           982#   1063    1081    1100    1117    1132    1146    1160    1173    1187    5846
BIT12 = 010000           981#   1064    1082    1101    1133    1147    1161    1174    1188
BIT13 = 020000           980#   1065    1083    1102    1118    1134    1148    1162    1175    1189    5891
BIT14 = 040000           979#   1066    1084    1103    1119    1135    1149    1163    1176    1190    1203    5251    5817
BIT15 = 100000           978#   1067    1068    1085    1104    1120    1136    1206    5251
BIT2  = 000004          1001#   1091    1110
BIT3  = 000010          1000#   1073    1092    1111
BIT4  = 000020           999#   1074    1093    1112    1125    1153    1180
BIT5  = 000040           998#   1075    1094    1113    1126    1140    1154    1167    1181
BIT6  = 000100           997#   1058    1076    1095    1114    1127    1141    1155    1168    1182
BIT7  = 000200           996#   1059    1077    1096    1115    1128    1142    1156    1169    1183
BIT8  = 000400           995#   1060    1078    1097    1116    1129    1143    1157    1170    1184
BIT9  = 001000           994#   1061    1079    1098    1130    1144    1158    1171    1185
BPTVEC= 000014          1010#
BSE   = 000200          1096#
BSERR   001534          1468#   3947*   3962*   3971*   3972    3989*   4042    4415    4484    4557
BSE20   002356          1476#   3997
BSE22   003356          1477#   3949    3986    4004    5354
BYP     024636          2917    2939    2943    2958    3010    3018    3036    3040    3044    3048    5021#
CALADD  001376          1406#   4068*   4434*   5247    5253    5255
CALCLK  026564          5404#
CALDIF  001370          1403#   8265    8276
CCLR  = 100000          1068#   3270    3281    3313    3338    3354    3386    3403    3411    3467    3476    3482    3512
                        3528    3542    3637    3682    3693    3765    3794    3880    3886    3897    3911    4151    4182
                        4547    4722    4732
CCSP    027412          3253    3454    3593    3662    3737    3839    4051    4412    4706    4756    5588#   5597
CCYL    001364          1401#   5332*   5340*   5346*   5358
CDT   = 002000          1062#
CERR  = 100000          1067#   2897    2996    3743    3959    4023    4086    4109    4129    4169    4205    4262    4332
                        4376    4451    4472    4511    4588    4632    5058
CFMT  = 010000          1064#
CKMR2   027126          3274    3285    3306    3317    3346    3358    3379    3390    3890    3901    4176    4186    5503#
CKMR3   027210          3278    3289    3310    3321    3351    3362    3383    3394    3460    3761    3894    3905    4179
                        4189    4529    5522#
CKSWR = 104407          5816    5880    5906    6692#
CLEAR = 000005          1045#
CLKOF   027072          5428    5489#
CLKON   026764          5408    5464#
CLOCK   027036          2800    5478#
```

```
CLRFLG  023730          2720    4805#
COE     = 001000        1098#
CONTC   = 000003        1398#   3507    5594
COUNT   001402          1410#   5405*   5418*   5479*   5481*
CR      = 000015         918#   5988    5998    6704    6709    6711    6716    6720    6728    6736    6746    6754    6759
                        6764    6769    6775    6781    6787    6794    6799    6805    6813    6818    6823    6827    6832
                        6836    6842    6844    6849    6854    6858    6860    6864    6868    6873    6878    6883    6889
                        6898    6907    6914    6920    6925    6931    6938    6947    6949    6955    6964    6973    6984
                        6995    7001    7008    7017    7023    7027    7036    7045    7053    7058    7063    7070    7074
                        7085    7093    7102    7109    7118    7124    7132    7137    7145    7150    7155    7162    7172
                        7182    7190    7195    7202    7213    7220    7229    7241    7249    7254    7261    7267    7272
                        7280    7285    7294    7304    7310    7317    7324    7332    7340    7346    7354    7358    7371
                        7389    7397    7715    8260    8212    8216
CRLF    = 000200         919#   5959    5998
CTO     = 004000        1063#
CYL     001436          1427#
CYLADD  001374          1405#   3882    5096*   5097*   5098*   5099*   5100*   5101*
CYLDIF  001372          1404#   5081*   5082*   5083*   5084*   5085*   5086*   5087    5089*   5133    5137    5140    5161
                        5165    5168    5190*   5191*
DATAO   001516          1459#   4099    4118    4122    4195    4198    4250    4321    4363    4498    4577    4619
DATAO1  001520          1460#
DATAI   001522          1461#   4158    4253    4316    4366    4463    4501    4572    4622
DCK     = 100000        1104#
DCLO    = 000020        1112#
DDISP   = 177570         925#   1287    2690
DDPCH   005456          1623#   2766*   2905    3139
DDT     = 000400        1116#
DDUMP   005454          1622#   2733*   4805
DF1     054720          1664    1670    1676    1682    1687    1693    1699    1705    8288#
DF10    055064          1710    1715    1720    1725    1730    1735    1740    1750    1755    1765    1781    1836    1841
                        1846    1861    1866    1891    1931    1956    1961    1986    2011    2016    2021    2026    2036
                        2051    2056    2061    2066    2071    2076    2086    2106    2111    2116    2121    2146    2151
                        2156    2161    2171    2176    2181    2186    2191    2196    2201    2211    2216    2221    2246
                        2271    2276    2291    2296    2301    2326    2331    2341    2346    2416    2487    8346#
DF11    055100          1745    1801    1806    1831    2031    2166    2206    8353#
DF14    055120          8362#
DF15    055130          2281    2286    2306    2311    2421    2426    2431    8368#
DF17    055144          2441    2446    8376#
DF2     054730          8293#
DF20    055164          1871    8384#
DF3     054744          2126    2136    8300#
DF4     054760          2131    2141    8307#
DF5     054774          2316    8314#
DF6     055000          8317#
DF7     055014          2336    8324#
DF8     055024          1811    1816    1821    1851    1876    1881    1886    1896    1901    1906    1921    1936    1946
                        1966    1976    1991    2001    2041    2081    2096    2226    2236    2251    2261    2266    2391
                        2396    2547    2622    2627    8328#
DF9     055044          1760    1770    1775    1786    1791    1796    1826    1856    1911    1916    1926    1941    1951
                        1971    1981    1996    2006    2046    2091    2101    2231    2241    2256    2321    2552    2582
                        2587    2612    2617    8337#
DH1     050367          1662    1668    1674    1680    1685    1691    1697    1703    2334    7865#   8302    8309    8319
                        8330    8339    8348    8355    8370    8380    8386
DH10    051005          7915#
DH11    051034          1929    1934    1939    1944    1949    7919#
DH12    051110          1954    1959    1964    1969    1974    1979    7927#
```

```
DH13    051140          1984    1989    1994    1999    2004    7931#
DH14    051224          7940#
DH15    051255          2014    2209    7945#
DH16    051302          2029    2034    2299    7949#
DH17    051341          1859    1864    1889    2084    2389    2394    2610    2615    7955#
DH18    051365          1738    1743    1829    1834    1839    7959#
DH19    051412          1779    2054    7963#
DH2     050404          7868#   8290    8295    8332    8341    8350    8357 -  8382
DH20    051435          2059    7967#
DH21    051470          1763    2064    7972#
DH22    051516          2189    2204    7976#
DH23    051550          2074    7981#
DH24    051602-         1799    1804    1819    1824    2545    2550    7986#
DH25    051627          1869    2109    2114    2344    7990#
DH26    051652          1718    1723    1748    1773    1784    1884    1894    2414    2419    2424    2429    7994#
DH27    051702          1708    1713    1758    1768    1874    1879    7998#
DH28    051733          2119    8003#
DH29    051767          2104    8008#
DH3     050446          7874#   8297    8359
DH30    052023          2269    2279    2284    8013#
DH31    052055          2144    2154    8018#
DH32    052111          1728    1733    1753    1789    1794    1899    1904    8023#
DH33    052143          8028#
DH34    052167          2159    2164    8032#
DH35    052223          2169    8037#
DH36    052254          8042#
DH37    052336          2174    2179    2184    2319    8051#
DH38    052372          2149    2194    2199    8056#
DH39    052416          2304    2309    2580    2585    2620    2625    8060#
DH4     050455          7876#   8334    8343
DH40    052451          8065#   8364
DH41    052526          2485    8073#
DH42    052553          2439    8077#
DH43    052624          2444    8084#
DH44    052674          8091#   8378
DH45    052753          2214    2219    2224    2229    2234    2239    2274    8099#
DH46    053012          2244    2249    2254    2259    2264    8105#
DH47    053057          8112#
DH48    053120          2024    8118#
DH49    053161          2049    8124#
DH5     050467          7878#
DH50    053221          2069    2079    2089    2094    2099    2124    2339    8130#
DH51    053267          8137#
DH52    053322          8142#
DH53    053356          1809    1814    1909    1914    8147#
DH54    053404          8151#
DH55    053433          8155#
DH56    053462          8159#   8388
DH57    053527          8166#
DH58    053567          8172#
DH59    053613          2019    2039    2044    8176#
DH6     050536          7885#   8321
DH60    053667          2129    8184#
DH61    053752          8193#   8304
DH62    054051          8204#   8311
DH63    054162          2134    2139    8218#
```

```
DH64    054233          2009    2289    8225#
DH65    054267          2294    8230#
DH66    054332          2314    8236#
DH67    054353          2324    2329    8239#
DH68    054416          8245#   8326
DH7     050620          7894#   8372
DH8     050667          1844    1849    1854    1919    1924    7901#
DH9     050732          7907#
DI    = 040000          1066#
DISPLA  001142          1287#   2690*   2698*   5861*   5883*
DISPRE  000174          1215#   2698
DLT   = 100000          1085#   3966
DLY     024620          4730    5012#   5015
DMD   = 000040          1126#
DOTIM   005512          1642#   2799*   2808*
DPAT1   001524          1462#
DPAT2   001526          1463#
DRA   = 000001          1108#   2933    3026
DRDY  = 000200          1115#
DRIVS   005464          1626#   2881*   2907*   2924    3014*   3128    4666    4699    4849*
DRIV0   005466          1631#   2781    2883    2982    3070    4832
DRIV1   005470          1632#
DRIV2   005472          1633#
DRIV3   005474          1634#
DRIV4   005476          1635#
DRIV5   005500          1636#
DRIV6   005502          1637#
DRIV7   005504          1638#
DROT  = 000040          1113#
DRPAR = 000010          1092#
DRVMSK= 000007          1072#   2900    2999
DRVPTR  001356          1394#   2781*   3133    3143*
DSC   = 040000          1119#
DSWR  = 177570           924#   1286    2689
DTE   = 010000          1101#   3963
DTYE  = 000040          1094#
DT1     054460          1663    1669    1675    1681    1686    1692    1698    1704    1709    1714    1719    1724    1729
                        1734    1739    1749    1754    1759    1764    1769    1774    1780    1785    1790    1795    1810
                        1815    1820    1825    1835    1840    1845    1850    1855    1860    1865    1875    1880    1885
                        1890    1895    1900    1905    1910    1915    1920    1925    1930    1935    1940    1945    1950
                        1955    1960    1965    1970    1975    1980    1985    1990    1995    2000    2005    2010    2015
                        2020    2025    2035    2040    2045    2050    2055    2060    2065    2070    2075    2080    2085
                        2090    2095    2100    2105    2110    2115    2120    2145    2150    2155    2160    2170    2175
                        2180    2185    2190    2195    2200    2210    2215    2220    2225    2230    2235    2240    2245
                        2250    2255    2260    2265    2270    2275    2290    2295    2300    2320    2325    2330    2340
                        2345    2390    2395    2415    2440    2445    2486    2546    2551    2581    2586    2611    2616
                        2621    2626    8255#
DT10    054700          2335    8282#
DT2     054504          1744    1800    1805    1830    2030    2165    2205    8259#
DT3     054526          2125    2130    2135    2140    8262#
DT4     054550          8265#
DT5     054570          2280    2285    2305    2310    2420    2425    2430    8268#
DT6     054616          2315    8272#
DT7     054622          8273#
DT8     054642          8276#
DT9     054662          1870    8279#
```

J14


```
D.ACLO= 000100          1168#  3756  3760  3785  4524  4528  4542
D.BRHM= 000100          1155#  3284  3316  3357  3389  3900  4185
D.CART= 000400          1157#  3284  3316  3334  3900  4185
D.DDT = 000400          1143#
D.DOOR= 000200          1156#  3266  3389  3685  3900  4185
D.DRA = 000040          1140#  3273  3305  3345  3378  3889  4175
D.DRDY= 000200          1142#  3093  3674  3781  3889  4175
D.DROT= 020000          1175#
D.DSC = 040000          1149#  3096  3463  3889  4175
D.FLT = 000200          1169#  3459  3760  3875  3893  4178  4528
D.FORM= 001000          1144#
D.FWD = 002000          1159#
D.HDFL= 000200          1183#
D.HDHM= 000040          1154#  3284  3316  3357  3389  3696  5203
D.IDAE= 000040          1167#
D.LIMD= 020000          1189#
D.LOAD= 010000          1161#  5220
D.MHD = 000400          1184#
D.NMOV= 010000          1188#
D.NXF = 000400          1170#  3870  3893
D.OFF = 002000          1145#
D.PAR = 001000          1171#  3459
D.PIP = 020000          1148#
D.PLO = 004000          1187#
D.REV = 004000          1160#
D.RTZ = 020000          1162#
D.SECT= 000020          1180#
D.SKI = 002000          1172#
D.SPIN= 010000          1147#  3301  3374  3670  3820  3889  4175
D.SPLS= 010000          1174#  4528
D.SPOK= 001000          1158#  3900  4185
D.SUNS= 040000          1190#
D.TFOK= 000020          1153#  3900  4185
D.TIB = 002000          1186#
D.UNLD= 040000          1163#
D.UNS = 040000          1176#
D.VV  = 000100          1141#  3209  3223  3273  3305  3341  3406  3417  3526  3564  3789  3800  3847
                        3852   3917  4175  4553
D.WCUR= 000040          1181#
D.WGAT= 000100          1182#
D.WLE = 004000          1173#  4171  4178  4275
D.WRL = 004000          1146#  4055  4062  4153  4175  4237  4420  4428
D.XERR= 001000          1185#
ECCW  = 020000          1134#
ECH   = 000100          1095#
EMTVEC= 000030          1013#  2673*  2674*
EM1     043163          4858   7371#
EM10    043737          1696   7437#
EM100   050301          7852#
EM101   050336          7857#
EM11    044022          1702   7446#
EM12    044066          1707   1717  1727  1737  1798  2053  2058  2063  2083  2108  2158  2188  2268
                        2303   2413  7452#
EM13    044124          1742   1803  1888  2038  2113  2163  7457#
EM14    044146          1828   7460#
EM15    044173          1833   7464#
```

```
EM16    044225          1838    7469#
EM17    044252          1808    1848    1873    1883    1898    1933    1963    1988    2078    2223    2248    2388    2544
                        2619    7473#
EM18    044273          1757    1772    1788    1853    1908    1938    1968    1993    2043    2088    2228    2253    2318
                        2549    2579    2609    7476#
EM19    044314          1813    1818    1878    1893    1903    1918    1943    1973    1998    2093    2233    2258    2393
                        2624    7479#
EM2     043237          1661    7379#
EM20    044335          1767    1783    1793    1823    1913    1923    1948    1978    2003    2098    2238    2263    2584
                        2614    7482#
EM21    044356          1712    1722    1752    1762    2283    2308    2343    2418    7485#
EM22    044400          1843    7488#
EM23    044441          1928    1983    7494#
EM24    044471          1778    7499#
EM25    044527          1953    7505#
EM26    044575          1958    2298    7512#
EM27    044637          2033    2073    2118    7518#
EM28    044664          2013    7522#
EM29    044716          2018    7527#
EM3     043327          5627    7389#
EM30    044743          2028    7531#
EM31    044774          2103    7536#
EM32    045025          2143    7541#
EM33    045061          2148    7546#
EM34    045112          2153    7551#
EM35    045153          2168    7557#
EM36    045207          7562#
EM37    045252          2178    7568#
EM38    045302          7572#
EM39    045344          1858    7578#
EM4     043402          5662    7397#
EM40    045413          1863    7585#
EM41    045451          7591#
EM42    045522          2183    7598#
EM43    045547          2193    7602#
EM44    045601          2198    7607#
EM45    045643          2208    7613#
EM46    045720          2213    2243    7621#
EM47    045777          2218    7629#
EM48    046037          2023    7635#
EM49    046077          2048    7641#
EM5     043456          1667    7405#
EM50    046133          2068    2328    7646#
EM51    046175          2123    2128    2133    2138    7652#
EM52    046250          2008    7660#
EM53    046275          2288    7665#
EM54    046325          2293    7670#
EM55    046375          2203    7677#
EM56    046430          2313    7682#
EM57    046474          2173    2323    2338    7688#
EM6     043477          1673    7408#
EM60    046522          7692#
EM61    046560          2273    7697#
EM62    046606          2423    7701#
EM63    046626          2278    2428    7704#
EM64    046647          2333    7707#
```

```
EM65    046774          7722#
EM66    047016          7725#
EM68    047052          2438    2443    7730#
EM7     043520          1679    7411#
EM74    047115          2484    7736#
EM8     043607          1684    7421#
EM80    047142          1732    7740#
EM81    047201          7746#
EM82    047245          7753#
EM83    047317          1747    7761#
EM84    047354          7766#
EM85    047412          7771#
EM86    047455          7777#
EM87    047510          7782#
EM88    047560          7789#
EM89    047605          7793#
EM9     043663          1690    7429#
EM90    047645          7799#
EM91    047704          7805#
EM92    047731          7809#
EM93    047762          1868    7814#
EM94    050025          7820#
EM95    050074          7827#
EM96    050130          7832#
EM97    050156          7836#
EM98    050217          7842#
EM99    050250          7847#
ENDRV   023226          4649#   5416    8464
ERRVEC= 000004          1006#   2687    2688*   2699*   2707*   2708*   2717*   2718*   2784*   2788*   2795*   2830*   2848*
                        5822    5823*   5825*   5828*
ESEC    001414          1416#
FATT1   024444          3830    4965#
FATT2   024540          3867    4020    4992#
FCYL    025270          5131#
FCYLOP  025360          5154#
FHDHM   025560          5200#   5208
FHDTAB  025702          4072    4438    5233#
FLOAD   025634          5217#
FMTE  = 000020          1093#
FMT1    001512          1456#   5243*   5244*   5245*   5250
FORMAT  001510          1455#   4070*   4436*   5243
FRCYL   001360          1399#   8265    8276
FRDY    024226          2876    2890    2895    2989    2994    3078    3085    3220    3414    3427    3516    3532    3546
                        3709    3797    3826    3863    3914    3956    4016    4083    4106    4126    4164    4202    4259
                        4329    4373    4448    4469    4507    4550    4585    4629    4726    4736    4912#   4915    5039
                        5053
FSEC17  025200          5108#
FTITLE  001354          1392#   4816    4818*
GBA     024130          2752    4872#
GDRVS   023770          2750    4827#
GETSP   027452          3258    3263    3298    3330    3371    3401    3474    3491    3572    3598    3608    3619    3628
                        3667    3680    3691    3702    3776    3844    4059    4149    4235    4269    4281    4425    4487
                        4518    4537    4711    4718    5605#   5611
GINT    024156          2754    4885#
GNS   = ****** U        1214    6684    6685    6686    6687    6688    6690    6692    6693    6694    6695    6696    6697
                        6698
```

```
GO     = 000001  1057#
GSTAT    024652  3173  3265  3272  3283  3300  3315  3333  3340  3356  3373  3388  3405  3458
                 3468  3477  3483  3563  3600  3620  3638  3669  3684  3695  3742  3766  3846
                 3888  3899  3958  4054  4061  4085  4108  4128  4152  4167  4184  4204  4236
                 4331  4375  4419  4427  4450  4471  4510  4587  4631  4975  4979  4999  5003
                 5035# 5057  5068  5080  5095  5202  5219
GTSWR  = 104406  6690#
HASOF    005416  1597#  3087  3090  4933*  4949  8259
HBA      005406  1593#  4929*
HCS1     005400  1590#  2897  2996  3743  3959  4023  4086  4109  4129  4168  4205  4262  4332
                 4376  4451  4472  4511  4588  4632  4926*  5058  8255  8259  8262  8268  8282
HCS2     005402  1591#  2929  2931  2935  2947  3022  3024  3028  3031  3966  4132  4208  4335
                 4379  4591  4635  4739  4927*  8255  8259  8262  8268  8282
HDA      005410  1594#  4283  4284  4532  4533  4930*  5320  5321  8268  8273  8282
HDB      005422  1599#
HDC      005420  1598#  4934*  5332  5340  5343  5345*  5346  8265  8276  8279  8282
HDS      005412  1595#  2933  3026  4931*  8255  8259  8268
HDTAB    001542  1472#  4074  4440  5235
HDWD     001532  1466#  8273
HD1      001506  1454#  5237*  5238*  5239*  5240*  5241*  5242*  5249
HEAD     001504  1453#  4069*  4076*  4077  4078*  4092*  4093  4435*  4442*  4443  4444*  4456*  4457
                 5237  5331*  5338*  5347*  5378
HER      005414  1596#  3963  4932*  8255  8259  8268
HMR1     005424  1600#  4935*  5523
HMR2     005426  1601#  2899  2998  3093  3096  3174  3209  3223  3266  3301  3334  3341  3374
                 3406  3417  3463  3526  3564  3670  3674  3685  3696  3781  3789  3800  3820
                 3847  3852  3917  4055  4062  4153  4237  4420  4428  4553  4936*  5081  5203
                 5220  5508  8255  8259  8262  8265  8268  8273  8276  8279  8282
HMR3     005430  1602#  3756  3785  3870  3875  4171  4275  4524  4542  4937*  5069  5096  5529
                 8255  8259  8262  8265  8268  8273  8276  8279  8282
HOLD     024264  4916  4919  4926#
HPAT     005434  1604#  4939*
HPOS     005432  1603#  4938*
HT     = 000011   916#  5957  5998
HVRC   = 000400  1097#
HWC      005404  1592#  4928*
HZ       001400  1408#  5481
IDAE   = 002000  1099#
IE     = 000100  1058#
ILF    = 000001  1089#
INTER    027716  4900  5703#
INVCYL   005356  1480#
IOTVEC = 000020  1011#  2671*  2672*
IR     = 000100  1076#
LCLKF    005506  1640#  2786*
LCVEC    001344  1383#  2787
LF     = 000012   917#  5992  5998  6704  6709  6711  6716  6720  6728  6736  6746  6754  6759
                 6764  6769  6775  6781  6787  6794  6799  6805  6813  6818  6823  6827  6832
                 6836  6842  6844  6849  6854  6858  6860  6864  6868  6873  6878  6883  6889
                 6898  6907  6914  6920  6925  6931  6938  6947  6949  6955  6964  6973  6984
                 6995  7001  7008  7017  7023  7027  7036  7045  7053  7058  7063  7070  7074
                 7085  7093  7102  7109  7118  7124  7132  7137  7145  7150  7155  7162  7172
                 7182  7190  7195  7202  7213  7220  7229  7241  7249  7254  7261  7267  7272
                 7280  7285  7294  7304  7310  7317  7324  7332  7340  7346  7354  7358  7371
                 7389  7397  7715  8200  8212  8216
LIMERR   001536  1469#
```

```
LKS     001342          1381#   2785    5470*   5494*
LOOP    026740          5171    5425    5452#
LPCNT   001462          1439#   5154*   5172    5420*   5435    5457*
LPTIM   001464          1440#   5437*
MCLK  = 000400          1129#
MDS   = 001000          1079#   2929    3022    4739
MDSERR  001540          1470#
MEMBAS= 172100          1387#   2710*
MEMERR  027654          2712    5689#
MEMPAR  011062          2707#
MEMVEC= 000114          1386#   2712*   2713*
MERD  = 001000          1130#
MESMSK= 000017          1124#
MEWD  = 002000          1131#
MIND  = 000200          1128#
MSG1    034062          4819    6704#
MSG10   035351          2880    6832#
MSG11   035376          5736    6836#
MSG12   035436          2926    6842#
MSG13   035445          2809    6844#
MSG14   035533          5021    6854#
MSG15   035556          2949    3144    6858#
MSG16   035570          3171    6860#
MSG17   035614          4003    6864#
MSG18   035644          3452    6868#
MSG19   035676          3591    6873#
MSG2    034755          2734    6787#
MSG20   035727          3660    6878#
MSG21   035760          3734    6883#
MSG22   036021          3837    6889#
MSG23   036102          5415    6898#
MSG24   036167          4049    6907#
MSG25   036213          6911#
MSG26   036232          4409    6914#
MSG27   036273          3130    6920#
MSG28   036323          4704    6925#
MSG29   036365          3822    6931#
MSG3    035026          2749    6794#
MSG30   036431          3261    6938#
MSG31   036513          3509    5596    5610    6947#
MSG32   036521          3328    6949#
MSG33   036561          3295    6955#
MSG34   036646          3296    3369    6964#
MSG35   036734          3368    6973#
MSG36   037031          3398    6984#
MSG37   037126          3257    3262    3297    3329    3370    3473    3490    3571    3597    3607    3618    3627    3642
                        3679    3690    3843    4058    4148    4234    4268    4280    4424    4486    4710    4717    6995#
MSG38   037172          3493    7001#
MSG39   037330          3494    3557    3567    7017#
MSG4    035060          2751    6799#
MSG40   037373          3566    7023#
MSG41   037505          3596    7036#
MSG42   037565          3606    7045#
MSG43   037643          3617    7053#
MSG44   037673          3626    7058#
MSG45   037723          3641    7063#
```

```
MSG46   037774          3665    3700    7070#
MSG47   040177          3256    3678    7093#
MSG48   040257          3689    7102#
MSG49   040454          3739    4481    7124#
MSG5    035123          2753    6805#
MSG50   040527          3773    4516    4535    7132#
MSG51   040561          3842    7137#
MSG52   040671          3252    3453    3592    3661    3735    3838    4050    4410    4705    7150#
MSG53   040772          4057    4233    4267    4279    4423    7162#
MSG54   041063          4147    4241    7172#
MSG55   041152          3495    3551    7182#
MSG56   041232          3988    7190#
MSG57   041436          4482    7213#
MSG58   041504          4709    7220#
MSG59   041566          4716    7229#
MSG6    035202          5708    6813#
MSG60   041671          4754    7241#
MSG61   042001          4747    7254#
MSG62   042047          4701    7261#
MSG63   042143          4755    7272#
MSG64   042336          4219    7294#
MSG65   042467          3260    3327    3397    7310#
MSG66   042535          7317#
MSG67   042660          3489    7332#
MSG68   042735          4753    7340#
MSG69   042775          3400    3666    3701    3775    4517    4536    7346#
MSG7    035236          2768    6818#
MSG70   043052          3399    3774    7354#
MSG71   043100          3570    7358#
MSG8    035273          3251    6823#
MSG9    035317          3472    6827#
MSP     = 000100        1127#
MULT6   026170          5276    5302#
M.ALGN= 040000          1203#
M.CADD= 017760          1202#   5097
M.CDIF= 017760          1195#   5082    5191
M.DPID  030040          5436    5751#
M.DP40  030076          5762#   5784
M.DP41  030132          5767    5775#
M.DP42  030142          5773    5779#
M.DP44  030174          5786    5792#
M.DP50  030206          5760    5797#
M.DRV =  000007         1194#
M.HEAD= 007000          1205#
M.ID  =  000003         1201#   5524
M.OFST= 017760          1196#
M.PAR =  100000         1206#   5561
M.SECT= 000760          1204#   5070
M.SER =  077770         1197#
NED     = 010000        1082#   2935    2947    3028    3031    3484    3519    3549    3601    3613    3621    3633    3639
                        3767
NEM     = 004000        1081#
NUDRV   013004          3111#   4668
NXF     = 000004        1091#
OFFERR  001502          1450#
OFFSET= 000015          1049#
```

```
OFST  = 000004          1110#
OPI   = 020000          1102#
OR    = 000200          1077#
O.ADR1  055542          8520*  8571*  8573*  8743   8744*  8782*  8797   8814   9005#
O.BACK  056266          8615   8656#
O.BD    057560          8801   8802   8950#
O.BKPT  055774          8566#  8619
O.BK1   056726          8514   8779#
O.BK2   056762          8784   8787#
O.BK3   057000          8786   8793#
O.BRK   056716          8531   8777#
O.BUF   057607          8859   8868   8873   8978#
O.B2    057062          8798   8807#
O.B3    057122          8806   8815#
O.CRDV  057272          8632   8649   8672   8676   8706   8710   8815   8858#
O.CLGL  056110          8589   8591   8599#
O.CLGT= 000014          8603   8968#
O.CLS1  057516          8926   8928#
O.CR    057562          8935   8937   8938   8952#
O.CRET  056222          8610   8639#
O.CRLF  057526          8647   8704   8800   8839   8844   8935#
O.CRLS  057534          8584   8937#
O.CRS   057540          8936   8938#
O.CSR1  057556          8835*  8851   8946#
O.CSR2  057557          8836*  8852   8947#
O.CT    055544          8764*  8807*  8809*  9006#
O.C1    056674          8767#  8808
O.DCD   056040          8572   8574   8584#  8641   8692   8817
O.DCD1  056044          8585#  8633   8677
O.EFF   056350          8618   8683#
O.EFF1  056466          8695   8714#
O.ENTR  055554          8511#
O.ERR   056024          8558   8569   8581#  8604   8629   8687   8930
O.ERR1  056360          8687#  8760
O.ERR2  056200          8629#  8663   8736
O.FTYP  057434          8583   8651   8666   8708   8813   8884   8900   8905#
O.GET   057356          8537   8587   8881#  8886   8888   8890
O.GO    056532          8612   8734#
O.GO2   056610          8745#  8770
O.LG  = 000006          8541   8976#
O.LGCH  057565          8600   8956#  8968
O.LGDR  056140          8607   8609#  8621
O.LGL = 000030          8621#
O.LGL1  056112          8600#  8605
O.LGL2  056132          8601   8606#
O.MSK   055534          8688   8689   8691   8998#
O.ODT   055550          1223   8509#  8987
O.OFST  056276          8616   8662#
O.OF1   056344          8675   8677#
O.OPI   056234          8613   8645#
O.OP2   056242          8562   8647#  8658
O.ORPC  055746          8554#  8614
O.P     057555          8529*  8734*  8759   8761*  8810*  8944#
O.PRI   055532          8521   8783   8787   8997#
O.PROC  056640          8620   8759#
O.PR1   056662          8763   8765#
```

```
0.RALL  056014          8532    8567    8573#
0.RCSR= 177560          8499#   8835    8837*   8847    8849    8851*   8881
0.RDB = 177562          8498#   8883
0.REGT  055676          8537#   8611
0.RSE1  057254          8848    8851#
0.RSP   055706          8539#   8542
0.RST   055612          8510    8519#
0.RSTT  057224          8740    8766    8844#
0.RST1  055642          8518    8526#
0.RTIT  056636          8528*   8754#
0.SCAN  056050          8548    8587#   8598
0.SP    055740          8540    8549#
0.SPC   057304          8861#   8869
0.SP1   055726          8545#   8550
0.STM = 000340          8488#   8530    8739    8765    8767
0.STRT  055600          8509    8516#
0.SVR   057134          8519    8779    8821#
0.SVTT  057172          8799    8811    8835#
0.T     057554          8741*   8768*   8780    8943#
0.TBIT  056562          8741#   8781
0.TBT = 000020          8489#   8742    8769    8805
0.TC    057520          8923    8929#
0.TCLS  057502          8554    8639    8645    8656    8922#
0.TCSR= 177564          8501#   8836    8838*   8845    8852*   8905    8911
0.TDB = 177566          8500#   8907*   8913*
0.TL    057601          8538    8541    8549    8970#   8976
0.TRTC  057616          8573    8744    8983#
0.TVEC= 000014          8487#   8512    8530*   8531*
0.TYPE  057420          8803    8874    8897#   8901    8939
0.TYP1  057500          8898    8909    8917#
0.TYP2  057460          8911#   8912    8915
0.UIN   055546          8520    8743*   8782    9007#
0.UPC   055526          8513*   8738*   8753    8777*   8794    8796*   8995#
0.URD   055510          8516    8546    8988#
0.USP   055524          8517*   8822*   8823    8994#
0.UST   055530          8511*   8742*   8752    8769*   8778*   8785    8805*   8996#
0.WDS   056356          8684    8686#
0.WDS2  056374          8691#   8713
0.WDS3  056424          8702#   8715    8721    8730
0.WDS4  056462          8702    8712#
0.WRD   056170          8609    8625#
0.WRDA  056214          8626    8634#
0.WRD1  056176          8628#   8635    8652
0.WSCH  056354          8617    8685#
0.XXX   057552          8821*   8829    8942#
0.45    055654          8527    8529#
PACK  = 000003          1044#   3218    3412    3530    3795    3912    4548
PARAM   001352          1391#   2652*   2655*   2746
PARSRT  010556          1220    2652#
PAT   = 000020          1125#   3457
PCA   = 004000          1132#
PCD   = 010000          1133#
PCLKF   005510          1641#   2790*   2797*   5468    5492
PCVEC   001346          1384#   2791    2798
PCYL    001366          1402#
PFSRT   013136          3154#   5738
```

```
PGE    = 002000          1080#
PIP    = 020000          1118#
PIRQ   = 177772           923#
PIRQVE= 000240           1017#
PKRB    001340          1380#
PKS     001334          1378#  2789   2796   5473*  5496*
PKSB    001336          1379#  5472*
PPTP    005462          1625#  2737*
PRGSRT  010572          2653   2656#  4859
PR0    = 000000           940#  5464
PR1    = 000040           941#
PR2    = 000100           942#
PR3    = 000140           943#
PR4    = 000200           944#
PR5    = 000240           945#  1377
PR6    = 000300           946#
PR7    = 000340           947#  2658   2708   2713   2778   2801   5409   5733   5734
PS     = 177776           920#   921
PSEC    001412          1415#
PSW    = 177776           921#
PWRVEC= 000024          1012#  2677*  2678*  5723*  5732*  5733*
QKCYLD  025470          5160   5164   5167   5180#
RDCHR  = 104410         5589   5606   6313   6693#
RDCYLA  025130          3881   5094#
RDCYLD  025044          5079#  5132   5136   5139
RDDATA= 000021          1051#  3954
RDGATE= 100000          1136#
RDHEAD= 000025          1053#
RDLIN  = 104411         4827   6385   6694#
RDOCT  = 104412         4872   4885   6695#
RDSEC   024774          5067#  5109   5113   5116   5274
RDTAB   004356          1478#
RDY    = 000200         1059#  4912   5184
RECAL  = 000013          1048#
RESREG= 104414          5440   6445   6497   6697#  8458
RESVEC= 000010          1007#
RHTAB   001746          1473#  5281   5283   5287   5290   8279
RKASOF= 000016          1030#  2839   4933   4968   4994
RKBA   = 000004         1025#  2835   3949*  3997*  4074*  4099*  4118*  4158*  4195*  4250*  4253*  4316*  4321*
                        4325   4363*  4366*  4369   4440*  4463*  4498*  4501*  4572*  4577*  4581   4619*  4622*
                        4625   4929
RKCS1  = 000000         1023#  2832   2893*  2992*  3083*  3218*  3270*  3281*  3313*  3338*  3354*  3386*  3403*
                        3411*  3412*  3425*  3467*  3476*  3482*  3512*  3514*  3528*  3530*  3542*  3544*  3637*
                        3682*  3693*  3707*  3765*  3794*  3795*  3824*  3861*  3880*  3886*  3897*  3911*  3912*
                        3954*  4014*  4081*  4104*  4124*  4151*  4162*  4182*  4200*  4257*  4327*  4371*  4446*
                        4467*  4505*  4547*  4548*  4583*  4627*  4722*  4724*  4732*  4734*  4912   4926   5037*
                        5182*  5184
RKCS2  = 000010         1027#  2833   2874*  2888*  2892*  2987*  2991*  3076*  3082*  3484   3513*  3519   3529*
                        3543*  3549   3601   3613   3621   3633   3639   3767   4100*  4119*  4159*  4196*  4254*
                        4322*  4367*  4464*  4502*  4578*  4623*  4723*  4733*  4927   5036*  5051*  5055*
RKDA   = 000006         1026#  2836   3950*  3974*  3998*  4077*  4102*  4121*  4161*  4256*  4292*  4297*  4301*
                        4303   4324*  4351*  4355*  4359*  4360   4443*  4504*  4580*  4607*  4611*  4615*  4616
                        4930
RKDB   = 000024         1032#  2841   4134   4210   4337   4381   4593   4637
RKDC   = 000020         1031#  2840   3860*  3951*  4934
RKDS   = 000012         1028#  2837   4931
```

```
RKECPS= 000030  1036#  2845  4938
RKECPT= 000032  1037#  2846  4939
RKER  = 000014  1029#  2838  4932
RKMR1 = 000026  1033#  2842  3172*  3264*  3271*  3282*  3314*  3332*  3339*  3355*  3387*  3404*  3457*
                3683*  3694*  3887*  3898*  4183*  4935   4974*  4978*  4998*  5002*  5056*  5067*  5079*
                5094*  5181*  5201*  5218*
RKMR2 = 000034  1034#  2843  4936  5190
RKMR3 = 000036  1035#  2844  4937
RKPRI   001332  1377#  4901
RKVEC   001330  1376#  2773*  4889*  4892*  4899
RKWC  = 000002  1024#  2834  3953*  4075*  4101*  4120*  4160*  4197*  4255*  4323*  4368*  4441*  4465*
                4503*  4579*  4624*  4928
RLS   = 000010  1073#
RTT   = 000006  8491#
SAVREG= 104413  5404   6430  6471  6696#  8400
SBMR2   005436  1609#  3273*  3284*  3305*  3316*  3345*  3357*  3378*  3389*  3889*  3900*  4175*  4185*
                5504*  5505   5507*  5508   8255   8268
SBMR3   005440  1610#  3277*  3288*  3309*  3320*  3350*  3361*  3382*  3393*  3459*  3760*  3893*  3904*
                4178*  4188*  4528*  5525*  5526   5528*  5529   8255   8268
SBPAR   027306  5506   5527  5547#
SCLR  = 000040  1075#  2874  2888  2987  3076  5051
SCOP1 = 104415  2885   2984  3073  3212  6698#
SCOP1S  027360  5571#  6698
SDC   = 000000  6703
SEC     001404  1412#  5406*  5419*  5482*
SECNT   001410  1414#
SECTOR  001416  1417#  5069*  5070*  5071*  5072*  5073*  5074*  5110   5114   5117   5275*  5279   5280*
                5281*  5292   5302*  5303   5304*  5305*  5330*  5339*  5348*  5370
SEEK  = 000017  1050#  3861  4014
SELDRV= 000001  1043#  2893  2992  3514  3544  4724  4734  5037  5182
SETINT  024210  2774   4890  4899#
SIZFLG  005514  1643#  2775*  2878  4845*  4865*
SKI   = 000002  1090#
SORT    026062  5272#
SPACE2  055464  8436   8445  8456  8466#
SPAR  = 020000  1065#
SPBAR = 000040  1397#  5591  5608
SRTSPL= 000011  1047#  3824
SRTTAB  002152  1474#  5284
ST    = 177776  8485#  8511  8512*  8525*  8739*  8765*  8767*  8793*
STACK = 001100   911#  2657  2669  2777  2826  2872  2886  2980  2985  3068  3074  3124  3164
                3204   3213  3246  3447  3586  3655  3729  3815  3941  4041  4404  4663  4693
                5698   5735  8463
START   010566  1218   2655#
START1  011152  2728#
STKLMT= 177774   922#
ST2     011220  2735   2746#
ST3     011264  2736   2764#
ST4     011316  2748   2769  2772#
ST5     011342  2755   2777#  4795
SUBCLR  024714  3168   3206  3215  3248  3292  3324  3365  3422  3449  3560  3588  3610  3630
                3657   3704  3731  3778  3817  3834  3857  3908  3943  4010  4046  4096  4115
                4144   4192  4230  4245  4307  4406  4460  4478  4493  4539  4563  4713  4760
                5051#
SUM     001466  1442#
SUM1    001472  1444#
```

```
SVAL   = 100000          1120#
SWR      001140          1286#  2667   2689*  2691   2697*  2704*  5571   5625   5633   5638   5660   5669   5674
                         5694   5817   5831   5833   5839   5846   5884   5891   5903   5907   6209   6246*  8401
                         8459
SWREG    000176          1216#  2697   6209   6222
SW0    = 000001           975#
SW00   = 000001           965#   975
SW01   = 000002           964#   974
SW02   = 000004           963#   973
SW03   = 000010           962#   972
SW04   = 000020           961#   971
SW05   = 000040           960#   970
SW06   = 000100           959#   969
SW07   = 000200           958#   968
SW08   = 000400           957#   967
SW09   = 001000           956#   966
SW1    = 000002           974#
SW10   = 002000           955#
SW11   = 004000           954#
SW12   = 010000           953#   8459
SW13   = 020000           952#   5625   5660   8401
SW14   = 040000           951#   5638   5674
SW15   = 100000           950#
SW2    = 000004           973#
SW3    = 000010           972#
SW4    = 000020           971#
SW5    = 000040           970#
SW6    = 000100           969#
SW7    = 000200           968#
SW8    = 000400           967#
SW9    = 001000           966#   5571   5633   5669   5694
TBITVE= 000014           1008#
TEMP1    005442          1612#  2875*  2889*  2894*  2899*  2900*  2901   2988*  2993*  2998*  2999*  3000   3077*
                         3084*  3219*  3413*  3426*  3515*  3531*  3545*  3708*  3741*  3745*  3764*  3769*  3796*
                         3825*  3862*  3866*  3913*  3955*  4015*  4019*  4082*  4105*  4125*  4163*  4201*  4258*
                         4328*  4372*  4447*  4468*  4506*  4549*  4584*  4628*  4725*  4729*  4735*  4914*  4966*
                         4970*  4996*  5012   5014*  5035   5038*  5041*  5052*  5108*  5120*  5131*  5143*  5180
                         5183*  5186*  5192*  5200*  5205*  5217*  5222*  5253*  5256*  5257   5407*  5413*  5503
                         5505*  5507   5510*  5513*  5522   5523*  5524*  5525   5526*  5528   5531*  5534*  5553*
                         5561*  8273
TEMP2    005444          1613#  3829*  3945*  3975*  3976   3996*  4972*  5133   5137   5140   5155*  5156*  5157*
                         5158*  5161   5165   5168   5207*  5254*  5255*  5257*  5258
TEMP3    005446          1614#  3946*  3978   3992*  3993
TEMP4    005450          1615#  3986*
TEMP5    005452          1616#
TIMUP    001406          1413#  5411   5421*  5423   5467*  5478*  5484*
TIM1     001452          1434#
TIM2     001454          1435#
TIM3     001456          1436#
TIM4     001460          1437#
TITLE    023750          2730   2767   4816#
TKVEC  = 000060          1015#
TOCYL    001362          1400#  4067*  4433*  8265   8276   8279
TPVEC  = 000064          1016#
.RAPPC   001350          1389#  5690*  8272
TRAPVE= 000034           1014#  2675*  2676*  5734*
```

```
TRT    = 000003          8490#   8983
TRTVEC= 000014           1009#
TRUERR  026212           4135    4211    4338    4382    4594    4638    5316#
TSTATN  024412           3429    3469    3478    3711    4947#
TST1    011524           2802    2824#
TST10   013376           3210    3224    3244#
TST11   014504           3445#
TST12   015312           3584#
TST13   015564           3634    3640    3653#
TST14   016060           3727#
TST15   016400           3813#
TST16   017112           3939#
TST17   017544           3981    3983    4039#
TST2    011666           2849    2870#
TST20   021650           4402#
TST21   023226           4661#
TST22   023262           4667    4691#
TST3    012276           2879    2978#
TST4    012622           3008    3065#
TST5    013004           3122#
TST6    013136           3162#
TST7    013274           3167    3202#
TYPDS  = 104405          4784    6688#
TYPE   = 104401          2734    2749    2751    2753    2768    2809    2880    2909    2926    2949    3130    3144    3152
                         3171    3193    3194    3251    3252    3256    3257    3260    3261    3262    3295    3296    3297
                         3327    3328    3329    3368    3369    3370    3397    3398    3399    3400    3452    3453    3472
                         3473    3489    3490    3493    3494    3495    3501    3509    3539    3551    3557    3566    3567
                         3570    3571    3591    3592    3596    3597    3606    3607    3617    3618    3626    3627    3641
                         3642    3660    3661    3665    3666    3678    3679    3689    3690    3700    3701    3734    3735
                         3739    3773    3774    3775    3822    3837    3838    3842    3843    3988    4003    4007    4008
                         4049    4050    4057    4058    4147    4148    4219    4233    4234    4241    4267    4268    4279
                         4280    4409    4410    4423    4424    4481    4482    4486    4516    4517    4535    4536    4701
                         4704    4705    4709    4710    4716    4717    4747    4753    4754    4755    4782    4785    4819
                         4858    5021    5415    5596    5610    5627    5662    5708    5736    5886    5894    5962    6056
                         6182    6220    6221    6224    6237    6248    6267    6320    6326    6331    6335    6340    6341
                         6343    6346    6350    6414    6416    6556    6684#   8412    8413    8417    8418    8427    8436
                         8440    8445    8447    8453    8456
TYPERR  055200           5893    8400#
TYPOC  = 104402          6223    6685#   8433
TYPON  = 104404          6687#
TYPOS  = 104403          2912    2952    3148    3498    3536    3554    4750    5024    5630    5665    5711    6686#
T1      001420           1419#
T10     001422           1420#   2875    2889    2894    2988    2993    3077    3084    3413    3426    3515    3531    3545
                         3708    3796    3825    3862    3913    4549    4725    4735    5038    5052    5183
T100    001430           1423#   3829
T50     001424           1421#   4015
T500    001426           1422#
T5000   001432           1424#   4242    4490    5108    5131
T50000  001434           1425#   3764    3866    3955    4019    4082    4105    4125    4163    4201    4258    4328    4372
                         4447    4468    4506    4584    4628
UFE    = 000400          1078#   2931    3024
UNLDRV  012622           2961    3054#
UNLOAD= 000007           1046#   3083    3425    3707
UNS    = 040000          1103#
UPE    = 020000          1083#
VV     = 000100          1114#
```

```
WCE    = 040000          1084#  4132   4208   4335   4379   4591   4635
WDCNT    001514          1457#  8273
WD1      001476          1447#  4134*  4210*  4337*  4381*  4593*  4637*  8262
WD2      001500          1448#  4122*  4198*  4325*  4369*  4581*  4625*  8262
WLE    = 004000          1100#
WORD     001530          1465#  5355*  5356   5358   5368*  5369*  5370   5375*  5376*  5377*  5378
WRDATA= 000023           1052#  4104   4162   4257   4467   4505
WRHEAD= 000027           1054#  4081   4446
WRL    = 004000          1117#
WRTCHK= 000031           1055#  4124   4200   4327   4371   4583   4627
WRTGAT= 040000           1135#
$APTHD   001000          1245   1251#
$ASTAT= ****** U         6099   6114
$ATYC    031430          6070   6072#
$ATY1    031404          6068#
$ATY3    031412          5947   6069#
$ATY4    031422          5899   6071#
$AUTOB   001134          1283#  6217   6366
$BASE    001260          1351#  2772*  2831   4876*  4878*
$BDADR   001122          1278#
$BDDAT   001126          1280#
$BELL    001174          1302#  5886   5919
$CDW1    001264          1353#
$CDW2    001266          1354#
$CHARC   031154          5964*  5974*  5981   5990*  5995#
$CKSWR   032100          6209#  6692
$CMTAG   001100          1266#  2664   2665   2673   2679   2680   2681
$CM3   = 000000          1296#
$CM4   = 000004          1296#  1297#  1298#  1299#  1300#
$CNTLG   032765          6220   6361#
$CNTLU   032760          6237   6335   6360#
$CPUOP   001232          1325#
$CRLF    001201          1304#  2909   3152   3193   3194   3501   3539   4007   4008   5894   5919   5963   5998
                         6248   6340   6360   6419   8412   8417   8427   8440   8453
$DBLK    031374          6022   6056   6064#
$DB2D    033274          6471#  6533
$DB2O    033154          4005   6430#
$DDW0    001270          1355#
$DDW1    001272          1356#
$DDW10   001314          1365#
$DDW11   001316          1366#
$DDW12   001320          1367#
$DDW13   001322          1368#
$DDW14   001324          1369#
$DDW15   001326          1370#
$DDW2    001274          1357#
$DDW3    001276          1358#
$DDW4    001300          1359#
$DDW5    001302          1360#
$DDW6    001304          1361#
$DDW7    001306          1362#
$DDW8    001310          1363#
$DDW9    001312          1364#
$DECVL   033454          6473   6519#
$DEVCT   001214          1316#  2721*  2782*  3134   4665*  4666
$DEVM    001262          1352#
```

```
$DOAGN  023704          4778    4787    4793#
$DTBL   031364          6025    6060#
$ENDAD  023674          1231    4789#   5914
$ENDCT  023642          2679    4780#
$ENDMG  023713          4782    4797#
$ENULL  023710          4785    4796#
$ENV    001224          1321#   5896    5942    6077    6101
$ENVM   001225          1322#   2702    5944    5949    6079
$EOP    023606          2853    2927    3131    4770#   5649    5685
$EOPCT  023634          2679*   4777#   4781
$ERFLG  001103          1269#   5573    5806    5835    5837    5843*   5865    5881*   5919    8461
$ERMAX  001115          1275#   2682*   5837    5860*   5865
$ERROR  030476          2673    5879#
$ERRPC  001116          1276#   5888*   5889*   5890    5919    8255    8259    8262    8265    8268    8273    8276    8279
                        8282
$ERRTB  005516          1658#   8409
$ERTTL  001112          1273#   5887*   5919
$ESCAP  001172          1301#   2681*   5689*   5693*   5859*   5910    5912    5919
$ETABL  001224          1320#
$ETEND  001330          1257    1373#
$FATAL  001206          1313#   6105*
$FFLG   031650          6068*   6071*   6099    6108*   6116#
$FILLC  001156          1294#   5967    5998
$FILLS  001155          1293#   5998
$GDADR  001120          1277#
$GDDAT  001124          1279#
$GET42  023664          4786#
$GTSWR  032150          6221#   6690
$HD   = 000000           887
$HIBTS  001000          1252#
$HIOCT  033152          6407*   6418#
$ICNT   001104          1270#   5850*   5851    5853*   5864
$INTAG  001135          1284#   6249    6366
$ITEMB  001114          1274#   5890*   5898    5919    8403
$LF     001202          1305#   5919    5998    6350    6360    6419
$LFLG   031647          6109*   6115#
$LPADR  001106          1271#   2683*   5636    5672    5841*   5857*   5862    5864
$LPERR  001110          1272#   2684*   5575    5578*   5699    5841    5858*   5864    5909
$MADR1  001236          1338#
$MADR2  001242          1342#
$MADR3  001246          1345#
$MADR4  001252          1348#
$MAIL   001204          1253    1257    1311#   2701    5856    5896    5942
$MAMS1  001234          1332#
$MAMS2  001240          1340#
$MAMS3  001244          1343#
$MAMS4  001250          1346#
$MBADR  001002          1253#
$MFLG   031646          6069*   6075    6110*   6114#
$MNEW   033003          6224    6364#
$MSGAD  001220          1318#   6085*   6088
$MSGLG  001222          1319#   6090*
$MSGTY  001204          1312#   6083    6091*   6103    6107*
$MSWR   032772          6221    6362#
$MTYP1  001235          1333#
$MTYP2  001241          1341#
```

```
$MTYP3  001245          1344#
$MTYP4  001251          1347#
$MULT   033564          6575#
$MXCNT  030474          5854    5864#
$NULL   001154          1292#   5969    5998
$NWTST= 000001          2814#   2816    2856#   2858    2963#   2965    3056#   3058    3113#   3115    3155#   3157    3196#
                        3198    3230#   3232    3433#   3435    3576#   3578    3645#   3647    3715#   3717    3806#   3808
                        3923#   3925    4030#   4032    4393#   4395    4651#   4653    4670#   4672
$OCNT   032074          6154*   6183*   6196#
$OCTVL  033256          3175    6432    6457#
$OMODE  032076          6149*   6153*   6158    6161*   6172*   6198#
$OVER   030460          5818    5834    5842    5852    5861#
$PASS   001212          1315#   2701*   3166    4001    4695    4774*   4775*   4783    4796    5848    5865
$PASTM  001006          1255#
$PWRCT  030036          5729*   5730*   5740#
$PWRDN  027752          2677    5723#   5732
$PWRUP  027764          5723    5729#
$QUES   001200          1303#   5919    5998    6267    6343    6360    6416    6419
$RDCHR  032362          6280#   6693
$RDDEC= ******  U       6696
$RDLIN  032502          6308#   6694
$RDOCT  033014          6380#   6695
$RDSZ = 000022          6301#
$RESRE  033734          6641#   6697
$RTNAD  023706          4795#
$R2A  = ******  U       6698
$SAVRE  033676          6625#   6696
$SB2D   033470          6531#
$SCOPE  030216          2671    5815#
$SETUP= 000137          2649#   2670    2671    2673    2675    2677    2679    2680    2681    2683    4772    5816    5880
                        5906    5914    6204    6366
$STUP = 177777          2649#
$SUPRS  033524          3192    4006    6548#
$SVLAD  030424          5826    5855#
$SVPC = 000244          1229#   1234
$SWR  = 167400           873#    887     891     892     893     894     895     896     897     898    1300    1301    1302
                        2680    2681    2683    2684    2825    2871    2979    3066    3123    3163    3203    3245    3446
                        3585    3654    3728    3814    3940    4040    4403    4662    4692    4767    4773    4788    4794
                        4796    5807    5808    5809    5810    5811    5817    5829    5831    5832    5835    5836    5837
                        5844    5845    5846    5858    5861    5864    5871    5872    5873    5874    5875    5884    5891
                        5903    5907    5919
$SWREG  001226          1323#   2704
$SWRMK= 000000           898     899    5811    5812    5833
$TESTN  001210          1314#   3125*   5856*   8255    8259    8262    8265    8268    8272    8273    8276    8279    8282
$TIMES  001170          1300#   2680*   2825*   2871*   2979*   3066*   3123*   3163*   3203*   3245*   3446*   3585*   3654*
                        3728*   3814*   3940*   4040*   4403*   4662*   4692*   4773*   5844*   5851    5854*   5864
$TKB    001146          1289#   3505    5588    5605    6202    6213    6230    6284    6290
$TKS    001144          1288#   3503    6202    6211    6227    6251*   6282    6288
$TMP0   001160          1296#   3505*   3506*   3507    4242*   4264*   4490*   4513*   5317    5320*   5323*   5327    5329*
                        5330    5364*   5383*
$TMP1   001162          1297#   4243*   4248    4272*   4314    4361    4491*   4496    4521*   4570    4617    5318    5321*
                        5324*   5325*   5331    5335    5337*   5338    5363*   5382*
$TMP2   001164          1298#   4283*   4286    4288    4290    4295    4300*   4301    4310    4312    4349    4353    4358*
                        4359    4532*   4566    4568    4605    4609    4614*   4615
$TMP3   001166          1299#   4284*   4303*   4324    4360*   4533*   4580    4616*   8262
$TN   = 000023           874#    887    2802    2814    2825#   2849    2856    2871#   2879    2963    2979#   3008    3056
```

```
                    3066#  3113   3123#  3125   3126   3155   3163#  3167   3196   3203#  3210   3224   3230
                    3245#  3433   3446#  3576   3585#  3634   3640   3645   3654#  3715   3728#  3806   3814#
                    3923   3940#  3981   3983   4030   4040#  4393   4403#  4651   4662#  4667   4670   4692#
$TNPWR  033404      6478   6479   6499#
$TPB    001152      1291#  5987*  5998
$TPFLG  001157      1295#  5936   5998
$TPS    001150      1290#  5985   5998
$TRAP   033772      2675   6661#
$TRAP2  034014      6672#  6683
$TRP  = 000016      6676#  6685#  6686#  6687#  6688#  6689#  6690   6691#  6692   6693#  6694#  6695#  6696#
                    6697#  6698#  6699#
$TRPAD  034026      6666   6683#
$TSTM   001004      1254#
$TSTNM  001102      1268#  3126*  4772*  5806   5833   5855*  5856   5861   5865   5883   5919
$TTYIN  032736      6310   6311   6323   6341   6355   6359#
$TYPBN= ****** U    6689
$TYPDS  031160      6010#  6688
$TYPE   030676      5936#  6096   6676   6684
$TYPEC  031110      5966   5973   5980   5985#  5986   6253
$TYPEX  031156      5991   5993   5996#
$TYPOC  031676      6152#  6685
$TYPON  031712      6151   6154#  6687
$TYPOS  031652      6147#  6686
$UNIT   001216      1317#  2722*  2783*  3136*  3141   3145   3496   3543   3552   4948   4967   4993   5036
                    5055   5504
$UNITM  001010      1256#
$USWR   001230      1324#
$VECT1  001254      1349#
$VECT2  001256      1350#
$XTSTR  030230      5820#
$$GET4= 000000      4788#
$OFILL  032075      6148*  6152*  6162   6197#
$40CAT= ****** U    5817   5893
.     = 055550      1210#  1214#  1219#  1222#  1229   1230#  1232#  1234#  1235#  1241   1242#  1244#  1246#
                    1265#  1306   1472#  1473#  1474#  1476#  1477#  1478#  2668   2683   2684   4796   4800
                    5725   5781   5864   5865   5919   5998   6064#  6117#  6202   6359#  6360   6366   6419
                    6457#  6519#  8475#  8476#  8621   8846   8850   8882   8906   8968   8976   8979#  8987#
.$ASTA= ****** U    6069   6072
.$X   = 001000      1241#  1246
```

```
CALIB    1519#
CKWD12   1507#
CKWD3    1513#
COMMEN   1018#
DRCLR    1514#
ENDCOM   1018#
ERROR     912#  2852  2877  2891  2896  2918  2940  2944  2955  2959  2990  2995  3011  3019  3037
          3041  3045  3049  3079  3086  3089  3092  3095  3098  3169  3207  3216  3221  3225  3249
          3268  3275  3279  3286  3290  3293  3303  3307  3311  3318  3322  3325  3336  3343  3347
          3352  3359  3363  3366  3376  3380  3384  3391  3395  3408  3415  3419  3423  3428  3430
          3450  3461  3465  3470  3480  3486  3517  3533  3547  3561  3589  3603  3611  3615  3623
          3631  3635  3658  3672  3675  3687  3698  3705  3710  3712  3732  3747  3758  3762  3771
          3779  3783  3787  3791  3798  3802  3818  3827  3831  3835  3849  3854  3858  3864  3868
          3872  3877  3884  3891  3895  3902  3906  3909  3915  3919  3944  3957  3961  3965  3968
          3980  3982  4011  4017  4021  4025  4047  4064  4084  4088  4097  4107  4111  4116  4127
          4136  4139  4145  4155  4165  4170  4173  4177  4180  4187  4190  4193  4203  4212  4215
          4231  4246  4260  4266  4277  4308  4330  4339  4342  4374  4383  4386  4407  4430  4449
          4453  4461  4470  4474  4479  4494  4508  4515  4526  4530  4540  4544  4551  4555  4564
          4586  4595  4598  4630  4639  4642  4714  4727  4737  4744  4761  5040  5054  5188  5351
          5691
ESCAPE   1018#
GETPRI   1018#
GETSWR   1018#
HDCHK3   1532#
HDTBL    1536#
LOOP     1497#  2884  2983  3072  3212
MSG      2814#  2816  2856#  2858  2963#  2965  3056#  3058  3113#  3115  3155#  3157  3196#  3198  3230#
         3232  3433#  3435  3576#  3578  3645#  3647  3715#  3717  3806#  3808  3923#  3925  4030#  4032
         4393#  4395  4651#  4653  4670#  4672
MULT     1018#
NEWTST   1018#  2814  2856  2963  3056  3113  3155  3196  3230  3433  3576  3645  3715  3806  3923
         4030  4393  4651  4670
OWNTAG   1259#  1375
POP      1018#  6051  6111  6112  6408  6604  6646
PUSH     1018#  6010  6072  6074  6095  6382  6575  6626
QKPACK   1575#  3410  3793  3911  4546
QKSEEK   1540#  4013
QKSRT    1571#  3816
QKUNLD   1579#  3421  3703
RDATA    1552#
RDHDR    1530#
REPORT   1018#
SBOUND   1567#  4306  4561
SCOPE     913#  2824  2870  2978  3065  3122  3162  3202  3244  3445  3584  3653  3727  3813  3939
         4039  4402  4661  4691  4771
SETPRI   1018#
SETTRA   6676#  6685  6686  6687  6688  6690  6692  6693  6694  6695  6696  6697  6698
SETUP    1018#  2662
SKIP     1018#  2802  2849  2879  3008  3167  3210  3224  3634  3640  3981  3983  4667
SLASH    1018#
SPACE    1018#
STARS    1018#  1227  1238  1240  1247  1261  1306  1309  2814  2823  2856  2869  2963  2977  3056
         3064  3113  3121  3155  3161  3196  3201  3230  3243  3433  3444  3576  3583  3645  3652
         3715  3726  3806  3812  3923  3938  4030  4038  4393  4401  4651  4660  4670  4690  4764
         5803  5867  5921  6000  6067  6124  6201  6204  6272  6301  6368  6421  6460  6522  6540
         6563  6610  6655  8390  8399
```

```
SWRSU    1018#   2685#
TRMTRP   6676#
TYPBIN   1018#
TYPDEC   1018#   4783
TYPNAM   1018#
TYPNUM   1018#
TYPOCS   1018#   2910    2950    3146    3496    3534    3552    4748    5022    5628    5663    5709
TYPOCT   1018#   6222
TYPTXT   1018#
WDATA    1546#   4103    4466
WRCHK    1560#   4123    4199    4326    4370    4582    4626
WRHDR    1525#   4080    4445
$$CMRE   1259#
$$CMTM   1259#   1296    1297    1298    1299
$$ESCA   1018#
$$NEWT   1018#   2814    2856    2963    3056    3113    3155    3196    3230    3433    3576    3645    3715    3806    3923
         4030    4393    4651    4670
$$SET    6676#   6685    6686    6687    6688    6690    6692    6693    6694    6695    6696    6697    6698
$$SETM   2701#
$$SKIP   1018#   2802    2849    2879    3008    3167    3210    3224    3634    3640    3981    3983    4667
.EQUAT    873#    908
.HEADE    873#    877
.SETUP    873#   2649
.SWRHI    873#    887
.SWRLO    873#    899#
.$ACTI    873#   1225
.$APTB   1307#
.$APTH    873#   1236
.$APTY    873#   6065
.$CATC    873#   1208
.$CMTA    873#   1259
.$DB2D    873#   6458
.$DB2O    873#   6419
.$EOP     873#   4762
.$ERRO    873#   5865
.$MULT    873#   6561
.$RDOC    873#   6366
.$READ    873#   6199
.$SAVE    873#   6608
.$SB2D    873#   6520
.$SCOP    873#   5801
.$SUPR    873#   6538
.$TRAP    873#   6653
.$TYPD    873#   5998
.$TYPE    873#   5919
.$TYPO    873#   6122

. ABS.  057620     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 .NOW.SEQ/SOL/CRF/NL:MD:TOC/EQ:SDC=DZR6JB.CMB
 RUN-TIME: 76 66 8 SECONDS
```

RUN-TIME RATIO: 267/151=1.7
CORE USED: 33K (65 PAGES)

Spooler runtime 27 Seconds, 115 KCS, 761 disk reads, 3 disk writes, 196 pages