RK611/RK06

FUNCTIONAL CONT DIAG

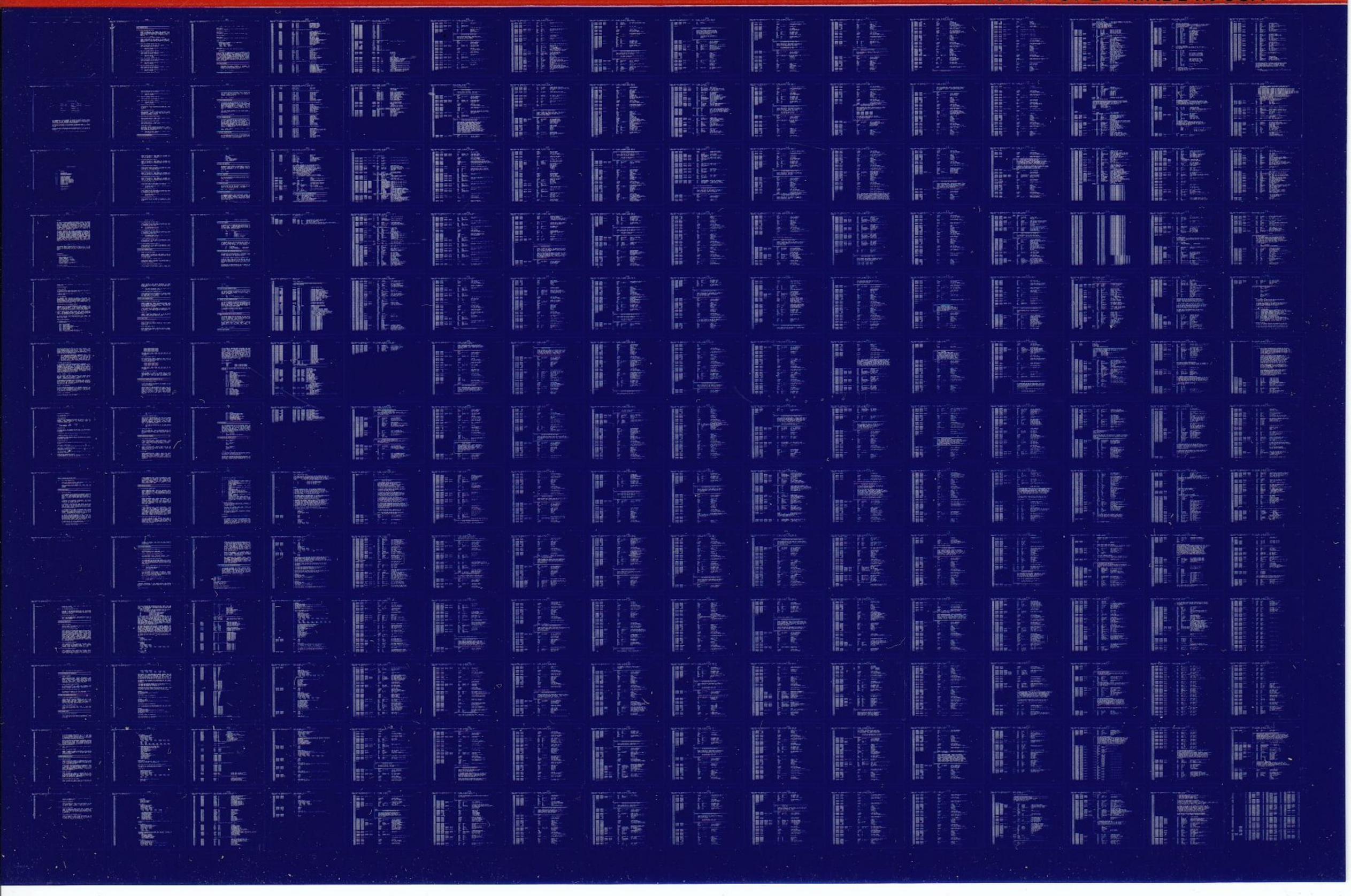
MD-11-DZR6K-D

EP-DZR6K-D-DL-C

APR 1977

FICHE 1 OF 2

MADE IN USA



RK611/RK06

FUNCTIONAL CONT DIAG
MD-11-DZR6K-D

EP-DZR6K-D-DL-C

FICHE 2 OF 2

APR 1977
MADE IN USA

B01

EOF1DZR6JDSEQ	00010000	770323	POP10 411	10HDR1DZR6KDSEQ	00010000	770323

```
                        .REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6K-D-D
PRODUCT NAME:	RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DATE:	FEBRUARY, 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	MARV TEGROTENHUIS

TABLE OF CONTENTS

1.0 ABSTRACT

2.0 REQUIREMENTS
2.1 HARDWARE REQUIREMENTS
2.2 PRELIMINARY PROGRAMS

3.0 OPERATING PROCEDURE
3.1 LOADING PROCEDURE
3.2 STARTUP PROCEDURE
3.3 CONSOLE SWITCH REGISTER
3.4 SOFTWARE SWITCH REGISTERS
3.5 CONTROL C (↑C) OPERATION
3.6 CONTROL S (↑S) OPERATION
3.7 CONTROL Q (↑Q) OPERATION
3.8 UNIBUS ADDRESS
3.9 EXECUTION TIME

4.0 PROGRAM DESCRIPTION

5.0 ERROR REPORTING

6.0 SUBROUTINES

1.0 ABSTRACT

THE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC (DZR6K) IS A SERIES OF TESTS THAT COMPLETES THE TESTING OF AN RK611/RK06 SUBSYSTEM. THE DISKLESS CONTROLLER DIAGNOSTIC AND THE RK06 DRIVE DIAGNOSTIC ARE PREREQUISITES TO THE RUNNING OF THIS PROGRAM. THE PURPOSE OF THIS PROGRAM IS TO TEST THOSE AREAS IN THE CONTROLLER THAT COULD NOT BE TESTED IN A DISKLESS ENVIRONMENT AND THOSE AREAS OF THE DRIVE THAT COULD NOT BE TESTED UNTIL CONTROLLER OPERATION IN A DIAGNOSTIC OR MAINTENANCE MODE HAS BEEN TESTED.

THE TESTS PERFORMED ARE MAINLY FUNCTIONALLY ORIENTED BUT DIAGNOSTIC MODE IS USED IN NUMEROUS OCCASSIONS TO ACCOMPLISH THE OBJECTIVES, MAINLY THE FORCING OF ERRORS. IN THESE CASES, THE CONTROLLER IS PLACED IN DIAGNOSTIC MODE AND OPERATION IS CLOCKED PART WAY THROUGH. DIAGNOSTIC MODE IS THEN RESET AND THE CONTROLLER ALLOWED TO COMPLETE THE OPERATION. DEPENDING ON THE OPERATION AND HOW FAR THROUGH IT BEFORE DIAGNOSTIC MODE IS RESET VARIOUS ERROR CONDITIONS CAN BE MADE TO OCCUR. THIS DOCUMENT DOES NOT ATTEMPT TO EXPLAIN WHY THESE ERROR CONDITIONS ARE SET BUT THE INDIVIDUAL TEST DESCRIPTIONS SPECIFY WHAT ERROR IS BEING FORCED AND THE PROCEDURE USED TO FORCE IT.

C A U T I O N

THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING A ↑C. IF THE PROGRAM IS HALTED USING THE HALT KEY THE POSSIBILITY EXISTS THAT THE CARTRIDGE FORMAT WILL BE INCORRECT, THE CYLINDER ADDRESS IN THE DRIVE MAY BE INVALID, OR THE DRIVE MAY NOT BE READY.

2.0 REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

PDP-11 SYSTEM (16K MEMORY)
CONSOLE TERMINAL
DECTAPE, PAPERTAPE, OR DISK
LINE CLOCK (KW11-L) (OPTIONAL)
PARITY OPTION (MM11) (OPTIONAL)
RK611 CONTROLLER
AT LEAST 1 AND UP TO 8 RK06 DRIVES
FORMATTED RK06K ON EACH DRIVE

2.2 PRELIMINARY PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC (ALL PARTS) AND THE

UNIBUS RK06 DRIVE DIAGNOSTIC (ALL PARTS) SHOULD HAVE RUN SUCCESSFULLY.

3.0 OPERATING PROCEDURE

3.1 LOADING PROCEDURE

THE PROGRAM CAN BE LOADED FROM BINARY TAPE USING THE ABSOLUTE LOADER OR FROM XXDP MEDIA SUPPORTED BY XXDP.

IT CAN BE LOADED AND RUN UNDER APT OR ACT AND IT CAN BE CHAINED BY XXDP.

3.2 STARTUP PROCEDURE

THE PROGRAM START LOCATION IS 200(8). THIS START WILL AUTOMATICALLY SIZE THE SYSTEM UNLESS RUNNING UNDER APT. THE PROGRAM ASSUMES THE STANDARD UNIBUS ADDRESS, VECTOR ADDRESS, AND BUS PRIORITY LEVEL (177440, 210, AND 4 RESPECTIVELY). IF STARTED AT 200 AND THE XXDP MEDIA IS RK06 (PROGRAM LOADED FROM RK06) DRIVE 0 IS NOT TESTED.

LOCATION 204(8) IS THE PROGRAM RESTART.

LOCATION 214(8) IS THE PARAMETERIZATION START LOCATION. THE OPERATOR WILL BE ASKED TO IDENTIFY THE BUS ADDRESS, VECTOR ADDRESS, AND BUS PRIORITY. IF THE PROGRAM WAS LOADED FROM RK06, THE OPERATOR WILL BE ASKED TO MOUNT A WORK CARTRIDGE ON DRIVE 0 OR TO PLACE IT OFF-LINE IF IT IS NOT TO BE TESTED.

LOCATION 220(8) IS THE PHASE LOCKED LOOP CLOCK ADJUSTMENT START. THE PROGRAM FIRST RUNS THE FIRST THREE TESTS AND THEN JUMPS TO THE ADJUSTMENT ROUTINE. THE PROGRAM WILL CONTINUE TO LOOP IN THIS ROUTINE UNTIL THE PROCESSOR IS HALTED.

ALL DRIVES THAT ARE TO BE TESTED MUST BE ON-LINE, READY, AND WRITE LOCK RESET. IF ALL THREE CONDITIONS ARE NOT MET, THAT DRIVE IS NOT TESTED.

3.3 CONSOLE SWITCH REGISTER

THE CONSOLE SWITCH REGISTER IS USED TO PROVIDE PROGRAM CONTROL AS DESCRIBED BELOW:

SW15 - HALT ON ERROR
SW14 - LOOP ON TEST
SW13 - INHIBIT ERROR REPORT
SW12 - ABORT PROGRAM AFTER 20 ERRORS
SW11 - INHIBIT ITERATIONS
SW10 - BELL ON ERROR
SW09 - LOOP ON ERROR
SW08 - EXECUTE TEST NUMBER SPECIFIED IN SW<7-0>.
SW<7-0> - EXECUTE THIS TEST IF SW08 SET.

EXECUTING A SPECIFIC TEST MUST BE USED WITH CAUTION. SOME TESTS REQUIRE OTHERS TO BE RUN TO FORMAT THE PACK IN A SPECIFIC MANNER OR WRITE SPECIFIC DATA. TESTS THAT REQUIRE OTHERS TO BE RUN INDICATE THIS IN THE TEST DESCRIPTION. IT IS SUGGESTED THAT THE PROGRAM BE RUN IN THE DEFAULT SEQUENCE THE FIRST TIME AFTER IT HAS BEEN LOADED.

NOTE: TEST 3 MUST BE RUN BEFORE ANY SUBSEQUENT TEST. THIS TEST DETERMINES WHICH DRIVES ARE ON THE DRIVE BUS FOR ALL FOLLOWING TESTS. LIKEWISE, TEST 20 MUST BE RUN BEFORE ANY TEST SUBSEQUENT TO IT. THIS TEST READS THE BAD SECTOR FILES AND BUILDS TABLES USED BY THE FOLLOWING TESTS. THESE TESTS, HAVING BEEN RUN ONCE, NEED NOT BE RUN AGAIN IF A DIFFERENT TEST IS SELECTED.

3.4 '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 SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING '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 SETTINGS ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROES 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.

3.5 CONTROL C (↑C) OPERATION

IF ↑C IS TYPED AT ANY TIME DURING THE PROGRAM EXECUTION THE PROGRAM IS HALTED IMMEDIATELY. IF A MONITOR IS PRESENT (XXDP CHAIN, ACT, APT) THE PROGRAM RETURNS CONTROL TO THE MONITOR. IF NO MONITOR IS PRESENT, THE CPU IS HALTED. DEPRESSING THE CONTINUE KEY WILL DO A PROGRAM RESTART.

3.6 CONTROL S (↑S) OPERATION

IF ↑S IS TYPED AT ANY TIME THE PROGRAM WILL GO INTO A STALL LOOP UNTIL A CONTROL Q (↑Q) IS TYPED.

3.7 CONTROL Q (↑Q) OPERATION

IF A ↑S HAS BEEN TYPED, TYPING THE ↑Q CANCELS THE STALL INITIATED BY THE ↑S.

3.8 UNIBUS ADDRESSES

STANDARD UNIBUS ADDRESSES ARE ASSUMED FOR THE KW11-L AND MM11 OPTIONS. THESE ADDRESSES MAY BE CHANGED BY CHANGE THE APPROPRIATE MEMORY LOCATIONS. THE FOLLOWING TAGS AND LOCATIONS HAVE BEEN USED:

KW11-L	TAG	LOCATION
UNIBUS ADDRESS	KWLADD	1710
VECTOR ADDRESS	KWLVEC	1712

3.9 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM FOR ONE DRIVE IS APPROXIMATELY 65 SECONDS AND EACH SUBSEQUENT PASS IS APPROXIMATELY 2 MINUTES 20 SECONDS.

THE EXECUTION TIME FOR MULTIPLE DRIVES IN THE FIRST PASS IS:

((NUMBER OF DRIVES) X 45 SEC) + 20 SEC

FOR SUBSEQUENT PASSES THE RUN TIME IS THE PRODUCT OF 2 MINUTES 20 SECONDS TIMES THE NUMBER OF DRIVES PLUS 25 SECONDS FOR EACH DRIVE AFTER THE FIRST.

4.0 PROGRAM DESCRIPTION

THE FOLLOWING TEST SEQUENCE IS EXECUTED ASSUMING TWO OR MORE DRIVES.

FIRST PASS - FIRST DRIVE:
ALL TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE.

FIRST PASS - ALL REMAINING DRIVES:
STATUS VALID TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH DRIVE.

THEN MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH COMBINATION OF DRIVES.

SECOND AND ALL SUBSEQUENT PASSES:
THE SAME SEQUENCE OF TESTING IS REPEATED EXCEPT FOR TEST ITERATIONS WHICH ARE SPECIFIED FOR EACH TEST.

****

```
                        **BASIC INTERFACE AND OPTION TESTS
                        ************************************************************

                        TEST 1  RK611 BASE ADDRESS TEST

                                CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES
                                NOT CAUSE A NON-EXISTANT MEMORY TRAP.

                                TEST 2 INTERRUPT VECTOR ADDRESS TEST  CHECK  THAT  THE
                                INTERRUPT  VECTOR FOR THE RK611 IS SET TO THE EXPECTED
                                ADDRESS.

                        ************************************************************
                        **STATUS VALID TESTS
                        ************************************************************

                        TEST 3  SELECT ALL DRIVES

                                IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT, DETERMINE
                                WHAT  DRIVES  ARE ON-LINE BY SELECTING ALL DRIVES.  IF
                                NON-EXISTENT DRIVE REPORTED MAKE SURE STATUS VALID  IS
                                RESET.   IF  DRIVE  PRESENT MAKE SURE NO ERROR EXISTS,
                                DRIVE IS CYCLED UP, AND  STATUS  VALID  SET,  AND  DSC
                                RESET.

                                IF RUNNING IN APT AUTOMATIC  ENVIRONMENT,  THE  DRIVES
                                IDENTIFIED  IN  ETABLE  ARE TESTED FOR NO ERROR, DRIVE
                                CYCLED UP, AND STATUS VALID SET.

                                IF LOCATION 41 INDICATES THE  XXDP  MEDIA  IS  ON  THE
                                RK06,  DRIVE  0 WILL ONLY BE TESTED IF THE PARAM START
                                (214) WAS USED.  IF THE AUTOMATIC START (200) IS USED,
                                DRIVE  0 IS NOT TESTED.  THE RESTART (204) WILL RETAIN
                                THE TEST STATUS OF DRIVE 0.

                                IF THE PARAM START IS USED, THE OPERATOR  MUST  EITHER
                                PLACE  DRIVE  0  OFF LINE IF IT IS NOT TO BE TESTED OR
                                UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT  IS  TO  BE
                                TESTED.   THE PROGRAM WILL MONITOR OFF LINE AND VOLUME
                                VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.

                                ALL DRIVES TO BE TESTED MUST BE  ON-LINE,  CYCLED  UP,
                                AND WRITE LOCK RESET. ADDRESSES  OF  DRIVES  THAT  ARE
                                NON-EXISTANT EITHER BECAUSE THE DRIVE DOES  NOT  EXIST
                                OR IS OFF-LINE ARE USED TO VERIFY  NON-EXISTANT  DRIVE
                                ERROR DETECTION.  DRIVES  THAT  ARE  ON-LINE  BUT  NOT
                                CYCLED UP OR ARE WRITE LOCKED ARE NOT TESTED.

                                AT COMPLETION OF THE TEST A MESSAGE WILL  BE  GIVEN  TO
                                IDENTIFY THE DRIVES TO BE USED IN TESTING.

                                NOTE:  THIS TEST MUST BE RUN AT LEAST ONCE BEFORE  ANY
                                       OTHER TEST THAT FOLLOWS.
```

```
369
370                                     TEST 4  RELEASE ALL DRIVES
```

```
                        RELEASE ALL DRIVES.  MAKE  SURE  NO  ERROR  SETS  AND
                        STATUS VALID IS RESET.

                .  TEST 5  NON-STANDARD MESSAGES AND SVAL

                        PICK ONE OF THE AVAILABLE DRIVES AND  GET  NON-STANDAD
                        MESSAGES.  MAKE SURE NO ERROR OCCURS AND STATUS VALUD
                        DOES NOT SET  AND  THAT  NON-STANDARD  MESSAGES  CAUSE
                        STATUS VALID TO RESET.

                   TEST 6  WRITING CS2 AND STATUS VALID

                        SELECT AN AVAILABLE DRIVE.  MAKE SURE STATUS VALID  IS
                        SET.  WRITE COMMAND AND STATUS REGISTER 2.  MAKE SURE
                        STATUS VALID RESETS.

                   ****************************************************************
                   **CONTROLLER ERROR TESTS
                   ****************************************************************

                   TEST 7  DRIVE TYPE ERROR

                        CREATE A DRIVE TYPE ERORR MAKE SURE DRIVE  TYPE  ERROR
                        SETS AND STATUS VALID SETS.

                TEST 10 STATUS VALID AND PARITY ERROR

                        ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY.
                        MAKE  SURE  SPAR,  CONTROLLER  ERROR, ATTENTION, DRIVE
                        STATUS CHANGES, DRPAR,  DRIVE  INTERRUPT,  AND  STATUS
                        VALID  SET, ISSUE A CONTROLLER CLEAR.  MAKE SURE DRIVE
                        INTERRUPT AND ATTENTION ARE STILL SET.  SELECT  DRIVE
                        AGAIN  WITH  GOOD  PARITY.  MAKE SURE ATTENTION, DRIVE
                        STATUS  CHANGE,  DRPAR,  CONTROLLLER  ERROR,  DRIVE
                        INTERRUPT, AND STATUS VALID ARE SET AND SPAR IS RESET.
                        ISSUE A CONTROLLER CLEAR.  GET  NON-STANDARD  MESSAGES
                        AND  MAKE  SURE ONLY DRIVE INTERRUPT AND ATTENTION ARE
                        SET.  CLEAR ATTENTION WITH DRIVE  CLEAR.  REPEAT  FOR
                        ALL AVAILIABLE DRIVES.

                TEST 11 UNIT FIELD ERROR ON RELEASE

                        ISSUE A SUBSYSTEM CLEAR.  SELECT AN  AVAILABLE  DRIVE.
                        PUT  CONTROLLER  IN  DIAGNOSTIC MODE.  ISSUE A RELEASE
                        COMMAND.  CLOCK THROUGH PHASE  ADDRESS  2.  TURN  OFF
                        DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD ERROR SETS.

                TEST 12 UNIT FIELD ERROR ON SELECT

                        ISSUE A SUBSYSTEM CLEAR.  SELECT AN  AVAILABLE  DRIVE.
                        PUT  CONTROLLER  IN  DIAGNOSTIC  MODE.  ISSUE A SELECT
                        COMMAND WITH MESSAGE ID = 3 AND DRIVE  SELECTED  =  0.
                        CLOCK  THROUGH  PHASE  ADDRESS 6.  TURN OFF DIAGNOSTIC
```

```
                        MODE.  MAKE SURE UNIT FIELD ERROR SETS.

                ;*****************************************************************
                ;**ATTENTION HANDLING BY CONTROLLER
                ;*****************************************************************

                TEST 13 DOUBLE INTERRUPT

                        ISSUE A SUBSYSTEM CLEAR.  ISSUE A  RECALIBRATE.   MAKE
                        SURE STATUS VALID IS SET.  CHECK THAT SECOND INTERRUPT
                        OCCURS.  AFTER  SECOND  INTERRUPT  CHECK  THAT  STATUS
                        VALID  IS  RESET.   ISSUE  SELECT AND MAKE SURE STATUS
                        VALID SETS.  CLEAR DRIVE.   CHECK  THAT  DRIVE  STATUS
                        CHANGE SETS (BIT 14 OF DRIVE STATUS REGISTER)

                TEST 14 SINGLE INTERRUPT FROM ATTENTION

                        DO A SEEK TO CYLINDER  0.   WAIT  FOR  INTERRUPT  FROM
                        DRIVE  ATTENTION.   LOWER PRIORITY AGAIN AND MAKE SURE
                        ANOTHER INTERRUPT DOES NOT OCCUR.  CLEAR DRIVE.

                TEST 15 RESET ATTENTIONS WITH UNIBUS INIT

                        DO A SEEK TO CYLINDER  0  ON  ALL  AVAILIABLE  DRIVES.
                        ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.

                ;*****************************************************************
                ;**ILLEGAL DISK ADDRESS ERROR TESTS
                ;*****************************************************************

                TEST 16 ILLEGAL DISK ADDRESS (PART 1)

                        ISSUE A SEEK TO CYLINDER 0, HEAD 3.  MAKE SURE ILLEGAL
                        ADDRESS   ERROR   AND  SEEK  INCOMPLETE  SET.   CLEAR
                        CONTROLLER AND CLEAR DRIVE.   REPEAT  FOR  HEADS  4-7,
                        CHECKING  THAT  BOTH  IDAE AND SEEK INCOMPLETE SET FOR
                        HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.

                TEST 17 ILLEGAL DISK ADDRESS (PART 2)

                        ISSUE A SEEK TO CYLINDER  1000,  HEAD  0.   MAKE  SURE
                        ILLEGAL DISK ADDRESS ERROR SETS.  CLEAR CONTROLLER AND
                        DRIVE

                ;*****************************************************************
                ;**WRITE HEADER TESTS
                ;*****************************************************************

                TEST 20 READ BAD SECTOR INFORMATION

                        ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,  TRACK
                        2  TO  GET  THE  FACTORY  DETECTED BAD SECTOR FILE, 26
                        SECTOR MODE.
```

```
                        IF AN ERROR OCCURS, READ SECTOR  2,  4,  6,  OR  10(8)
                        UNTIL  A  SUCCESSFUL  READ  IS  DONE.   IF  NONE  READ
                        SUCCESSFULLY REMOVE THIS DRIVE FROM TEST.  WHEN A READ
                        IS  SUCCESSFUL, TEST THAT THE PACK IS NOT AN ALIGNMENT
                        PADK AND STORE THE ENTRIES FOR LATER USE.

                        REPEAT THIS SERIES OF OPERATIONS FOR FACTORY  DETECTED
                        BAD  SECTORS  24  SECTOR  MODE,  SOFTWARE DETECTED BAD
                        SECTORS 26 SECTOR  MODE,  AND  SOFTWARE  DETECTED  BAD
                        SECTORS  24 SECTOR MODE.  IF THE NUMBER OF BAD SECTORS
                        FOR 24 OR 26 SECTOR MODE EXCEED 20(10)  THE  DRIVE  IS
                        REMOVED FROM TESTING.

                        NOTE:  THIS TEST IS RUN IN THE  FIRST  (QUICK  VERIFY)
                        PASS ONLY.

                TEST 21 FORMAT IN 26 SECTOR FORMAT

                        FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
                        FORMAT.   VERIFY  FORMAT  AND  THAT  DATA LATE DID NOT
                        OCCUR WITH WRITE HEADER  ON  IN  READING  DATA  BUFFER
                        AFTER READ HEADER.

                ;*****************************************************************
                ;**HEADER RECOGNITION TESTS
                ;*****************************************************************

                TEST 22 BAD SECTOR ERROR

                        FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO  HAVE
                        SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR
                        1 (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
                        AND ALL OTHER SECTORS GOOD.

                        ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK
                        0,  SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS.  ISSUE
                        A WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1  OF  400
                        WORDS.  MAKE SURE BAD SECTOR ERROR SET.  ISSUE A WRITE
                        DATA OF 400 WORDS TO CYLINDER 0, TRACK  0,  SECTOR  2.
                        MAKE SURE NO ERROR SETS.

                TEST 23 HEADER VRC ERROR

                        FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO  HAVE
                        16 SECTORS WITH BAD HEADER VRC.  ISSUE A WRITE DATA OF
                        EACH OF THE SECTORS WITH A BAD HEADER VRC.  MAKE  SURE
                        HEADER  VRC  ERROR SETS.  ISSUE A WRITE DATA TO A GOOD
                        HEADER AND MAKE SURE NO ERROR OCCURS.

                TEST 24 BAD SECTOR ERROR AND HVRC ERROR

                        FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
                        BOTH A BAD SECTOR ERROR AND HEADER VRC.  ISSUE A WRITE
```

```
539
540
541
542                     DATA TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY
543                     HEADER VRC ERROR SETS.
544
545             TEST 25 OPERATION INCOMPLETE
546
547                     FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS
548                     THE WRONG FORMAT.  ISSUE A WRITE DATA OF 400 TO
549                     CYLINER 0, TRACK 0, SECTOR 21.  MAKE SURE OPI SET.
550
551             TEST 26 OPI WITH HVRC ERROR
552
553                     FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
554                     ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
555                     ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK
556                     0, SECTOR 17.  THAT BOTH OPERATION INCOMPLETE AND
557                     HEADER VRC SET.
558
559             TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR
560
561                     FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS
562                     AN HVRC ERROR.  ISSUE A WRITE DATA OF 400 WORDS TO
563                     CYLINDER 312, TRACK 0, AND SECTOR 21.  MAKE SURE HVRC
564                     IS NOT SET AT THE END OF THE OPERATION.
```

```
;**********************************************************************
;**DATA TRANSFER TESTS
;**********************************************************************

TEST 30 WRITE AND READ ONE SECTOR

        FORMAT CYLINDER 312,  ALL  TRACKS  AND  CYLINDER  313,
        TRACK  2 TO AGREED WITH BAD SECTOR INFORMATION.  ISSUE
        A WRITE DATA OF ONE SECTOR ON CYLINDER 312,  TRACK  0.
        READ  IT  BACK  TO  MAKE  SURE  IT AGREES WITH WHAT IS
        WRITTEN.

TEST 31 WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT

        ISSUE A WRITE DATA OF  ONE  SECTOR  TO  CYLINDER  312,
        TRACK 2, SECTOR 12 WITH INHIBIT BUS ADDRESS INCREMENT.
        READ DATA BACK TO MAKE SURE EVERY WORD IS THE SAME AND
        CORRECT.

TEST 32 WRITE DATA ADDRESS GREATER THAN 32K

        ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
        MAKE SURE CORRECT DATA IS ON DISK.

        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 32K  OF
               MEMORY IS PRESENT.

TEST 33 READ DATA ADDRESS GREATER THAN 32K

        ISSUE A READ DATA OF 400 WORDS WITH ADDRESS =  177770.
        CHECK MEMORY FOR CORRECT TRANSFER.

        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 32K  OF
               MEMORY IS PRESENT.

TEST 34 WRITE DATA ADDRESS GREATER THAN 64K

        ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.
        MAKE SURE CORRECT DATA IS ON DISK.

        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 64K  OF
               MEMORY IS PRESENT.

TEST 35 READ DATA ADDRESS GREATER THAN 64K

        ISSUE A READ DATA OF 400 WORDS WITH ADDRESS =  377770.
        CHECK MEMORY FOR CORRECT TRANSFER.

        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 64K  OF
               MEMORY IS PRESENT.

TEST 36 WRITE DATA ADDRESS GREATER THAN 96K
```

```
                        ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.
                        MAKE SURE CORRECT DATA IS ON DISK.

                        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 96K  OF
                               MEMORY IS PRESENT.

                TEST 37 READ DATA ADDRESS GREATER THAN 96K

                        ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.

                        CHECK MEMORY FOR CORRECT TRANSFER.

                        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 96K  OF
                               MEMORY IS PRESENT.

                TEST 40 PARTIAL SECTOR WRITE DATA

                        ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312,  HEAD
                        0,  SECTOR  0.   MAKE  SURE THE SECTOR WAS ZERO FILLED
                        CORRECTLY.

                TEST 41 PARTIAL SECTOR READ DATA

                        WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A  KNOWN
                        CONFIGURATION.   ISSUE  A  READ  DATA  OF 103 WORDS TO
                        CYLINDER 312, TRACK 0, SECTOR 0.  MAKE SURE  ONLY  103
                        WORDS GET TRANSFERRED TO MEMORY.

                TEST 42 WRITE DATA WITH NON-EXISTENT MEMORY

                        ISSUE A WRITE DATA OF 1  WORD  USING  ADDRESS  776000.
                        MAKE SURE NON-EXISTENT MEMORY SETS.

                TEST 43 READ DATA WITH NON-EXISTENT MEMORY

                        ISSUE A READ DATA OF  1  WORD  USING  ADDRESS  776000.
                        MAKE SURE NON-EXISTENT MEMORY SETS.

                TEST 44 UNIBUS PARITY ERROR

                        INITIALIZE A MEMORY LOCATION WITH BAD PARITY.  ISSUE A
                        WRITE  DATA  OF  400  WORDS STARTING AT A LOCATION 112
                        WORDS BEFORE THE LOCATION WITH BAD PARITY.  MAKE  SURE
                        THAT UNIBUS PARITY ERROR SETS.

                        NOTE:  THIS TEST IS ONLY  EXECUTED  IF  MEMORY  PARITY
                               OPTION EXISTS FOR BUFFER.

                ;*****************************************************************
                ;*MULTIPLE SECTOR OPERATIONS
                ;*****************************************************************

                TEST 45 TWO SECTOR WRITE DATA (PART 1)
```

```
                                        ISSUE A WRITE DATA OF  1000  WORDS  TO  CYLINDER  312,
                                        TRACK  0,  SECTOR  0.   READ DATA BACK ONE SECTOR AT A
                                        TIME AND MAKE SURE IT IS CORRECT.

                                TEST 46 TWO SECTOR WRITE DATA (PART 2)

                                        ISSUE A WRITE DATA OF  1000  WORDS  TO  CYLINDER  312,
                                        TRACK  0,  SECTOR  23.  READ DATA BACK ONE SECTOR AT A
                                        TIME AND MAKE SURE A MID-TRANSFER SEEK  DID  NOT  TAKE
                                        PLACE.

                                TEST 47 TWO SECTOR WRITE DATA (PART 3)

                                        ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312, TRACK
                                        0, SECTOR 10.  READ DATA BACK ONE SECTOR AT A TIME AND
                                        CHECK ZERO FILL OF SECOND SECTOR.

                                TEST 50 MID-TRANSFER SEEK ON WRITE (PART 1)

                                        ISSUE A WRITE DATA OF  1000  WORDS  TO  CYLINDER  312,
                                        TRACK  0,  SECTOR  25.  READ DATA BACK ONE SECTOR AT A
                                        TIME AND MAKE SURE A MID-TRANSFER SEEK DID TAKE PLACE.

                                TEST 51 MID-TRANSFER SEEK ON WRITE (PART 2)

                                        ISSUE A WRITE DATA OF  1000  WORDS  TO  CYLINDER  312,
                                        TRACK  2,  SECTOR  25.  READ DATA BACK ONE SECTOR AT A
                                        TIME AND MAKE SURE A MID-TRANSFER SEEK DID TAKE PLACE.

                                TEST 52 TWO SECTOR READ DATA (PART 1)

                                        ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK
                                        0, SECTOR 0, VERIFY THAT CORRECT DATA IS READ.

                                        NOTE:  TWO SECTOR WRITE DATA (PART 1) MUST BE EXECUTED
                                               BEFORE THIS TEST.

                                TEST 53 TWO SECTOR READ DATA (PART 2)

                                        ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK
                                        0,  SECTOR 23.  VERIFY THAT CORRECT DATA IS READ AND A
                                        MID-TRANSFER SEEK DOES NOT OCCUR.

                                        NOTE:  TWO SECTOR WRITE DATA (PART 2) MUST BE EXECUTED
                                               BEFORE THIS TEST.

                                TEST 54 TWO SECTOR READ DATA (PART 3)

                                        ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312,  TRACK
                                        0, SECTOR 10.  VERIFY THAT ALL 401 WORDS ARE PLACED IN
                                        MEMORY.

                                        NOTE:  TWO SECTOR WRITE DATA (PART 3) MUST BE EXECUTED
```

```
                                        BEFORE THIS TEST.

                                TEST 55 MID-TRANSFER SEEK ON READ (PART 1)

                                        ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK
                                        0,  SECTOR 25.  VERIFY THAT CORRECT DATA IS READ AND A
                                        MID-TRANSFER SEEK DOES OCCUR.

                                        NOTE:  MID-TRANSFER SEEK ON WRITE  (PART  1)  MUST  BE
                                               EXECUTED BEFORE THIS TEST.

                                TEST 56 MID-TRANSFER SEEK ON READ (PART 2)

                                        ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK
                                        2,  SECTOR 25.  VERIFY THAT CORRECT DATA IS READ AND A
                                        MID-TRANSFER SEEK DOES OCCUR.

                                        NOTE:  MID-TRANSFER SEEK ON WRITE  (PART  2)  MUST  BE
                                               EXECUTED BEFORE THIS TEST.

                                TEST 57 CYLINDER ADDRESS OVERFLOW (PART 1)

                                        ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,  TRACK
                                        2,  SECTOR  25.   MAKE  SURE CYLINDER ADDRESS OVERFLOW
                                        ERROR DOES NOT OCCUR.

                                TEST 60 CYLINDER ADDRESS OVERFLOW (PART 2)

                                        ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632,  TRACK
                                        2,  SECTOR  25.   MAKE  SURE CYLINDER ADDRESS OVERFLOW
                                        ERROR DOES OCCUR.

                                ;*****************************************************************
                                ;*18 BIT DATA TRANSFER TESTS
                                ;*****************************************************************

                                TEST 61 FORMAT IN 24 SECTOR FORMAT

                                        FORMAT CYLINDER 312, TRACK  0,  AND  TRACK  1  FOR  24
                                        SECTOR  FORMAT.   VERIFY FORMAT AND THAT DATA LATE DID
                                        NOT OCCUR WITH WRITE HEADER ON IN READING DATA  BUFFER
                                        AFTER READ HEADER.

                                TEST 62 24 SECTOR FORMAT DATA TRANSFER (PART 1)

                                        ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT TO
                                        CYLINDER 312, TRACK 0, SECTOR 0.  READ SECTOR BACK AND
                                        MAKE SURE IT IS CORRECT.

                                TEST 63 24 SECTOR FORMAT DATA TRANSFER (PART 2)

                                        LOAD A LOCATION WITH BAD PARITY.  ISSUE A  WRITE  DATA
                                        OF  400  WORDS  IN  24  SECTOR FORMAT TO CYLINDER 312,
```

```
                        TRACK 0, SECTOR 0 WITH BUFFER BEGINNING 112 WORDS
                        BEFORE WORD WITH BAD PARITY.  MAKE SURE UNIBUS PARITY
                        ERROR DOES NOT SET.  READ SECTOR BACK AND MAKE SURE IT
                        IS CORRECT.

                        NOTE:  THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
                               EXISTS FOR SPECIFIED LOCATION.

                TEST 64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

                        ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT
                        TO CYLINDER 312, TRACK 0, SECTOR 23.  READ SECTOR BACK
                        AND MAKE SURE IT IS CORRECT.  MAKE SURE THAT
                        MID-TRANSFER SEEK HAS TAKEN PLACE.

                ;*****************************************************************
                ;*SPECIAL DATA TRANSFER TESTS
                ;*****************************************************************

                TEST 65 MULTI-SECTOR DATA TRANSFER AND BSE

                        FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH
                        SECTOR 1 MARKED BAD.  ISSUE A WRITE DATA OF 1000 WORDS
                        TO CYLINDER 312, TRACK 0, SECTOR 0.  MAKE SURE BAD
                        SECTOR ERROR SETS AND RKDA IS CORRECT.  READ SECTOR 0
                        AND MAKE SURE IT IS CORRECT.

                        ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK
                        0, SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS AND THE
                        PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.

                TEST 66 FORMAT TEST

                        FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR
                        FORMAT.  MAKE SURE NO ERRORS SET.  READ SECTORS 0-25
                        AND MAKE SURE DATA CHECK DOES NOT OCCUR.

                ;*****************************************************************
                ;*WRITE CHECK TESTS
                ;*****************************************************************

                TEST 67 WRITE-CHECK WITH NO ERROR

                        WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN
                        PATTERN.  DO A WRITE-CHECK OF 400 WORDS.  MAKE SURE NO
                        ERROR OCCURS.

                TEST 70 WRITE CHECK ERROR (PART 1)

                        WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES.
                        WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME.
                        DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING
                        CONFIGURATIONS:
```

```
844
845
846
847                                     000001 000020 000400 010000
848                                     000002 000040 001000 020000
849                                     000004 000100 002000 040000
850                                     000010 000200 004000 100000
851
852                             MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE
853                             CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUNT
854                             IS CORRECT.
855
856                     TEST 71 WRITE CHECK ERROR (PART 2)
857
858                             WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777 IN
859                             ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0, SECTOR
860                             0 WITH THE SAME DATA EXCEPT WORD 120 HAS ONE OF THE
861                             FOLLWOING CONFIGURATIONS:
862
863                                     177776 177757 177377 167777
864                                     177775 177737 176777 157777
865                                     177773 177677 175777 137777
866                                     177767 177577 173777 077777
867
868                             MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE
869                             CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUNT
870                             IS CORRECT.
871
872                     TEST 72 WRITE CHECK OF PARTIAL SECTOR
873
874                             WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN
875                             CONFIGURATIONS. ISSUE A WRITE CHECK COMMAND OF 110
876                             WORDS MAKING SURE THE 111TH WORD IS DIFFERENT THAN
877                             DATA ON DISK. MAKE SURE WRITE CHECK ERROR DOES NOT
878                             SET.
879
880                     ********************************************************************
881                     **MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT
882                     ********************************************************************
883
884                     TEST 73 MAXIMUM DATA TRANSFER (PART 1)
885
886                             IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF THE
887                             FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO INSURE
888                             THE FORMAT IS CORRECT.
889
890                             ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH ONE
891                             SECTOR WRITES. ISSUE A SEEK TO CYLINDER 0, TRACK 0.
892                             ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 200000
893                             WORDS TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE
894                             CONTROLLER TIME OUT IS NOT SET. CHECK CYLINDER
895                             ADDRESS, DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
896                             READ EACH SECTOR TO MAKE SURE IT WAS WRITTEN
897                             CORRECTLY.
898
899                             NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS
```

```
                                        ARE  PRESENT  IN  THE  FIRST 256 SECTORS ON THE
                                        PACK.

                                TEST 74 MAXIMUM DATA TRANSFER (PART 2)

                                        ZERO OUT FIRST 256 SECTORS OF  THE  DISK  WITH  200000
                                        WORD  WRITE.   SEEK TO CYLINDER 632.  ISSUE A WRITE OF
                                        MAXIMUM DATA TRANSFER 200000 WORD  WRITE.   MAKE  SURE
                                        CONTROLLER  TIME  OUT  IS  NOT  SET.   CHECK  CYLINDER
                                        ADDRESS DISK ADDRESS,  BUS  ADDRESS  AND  WORD  COUNT.
                                        SEEK  TO  CYLINDER 632.  ISSUE A WRITE CHECK OF 200000
                                        WORDS.  MAKE SURE NO ERROR SETS.

                                        NOTE:  THIS TEST IS EXECUTED ONLY IF  NO  BAD  SECTORS
                                               ARE  PRESENT  IN  THE  FIRST 256 SECTORS ON THE
                                               PACK.

                                TEST 75 CONTROLLER TIME OUT

                                        SEEK TO CYLINDER 632.  ISSUE A RECALIBRATE AND DO  NOT
                                        WAIT FOR SECOND INTERRUPT.  NOW ISSUE A READ HEADER OF
                                        CYLINDER 0, TRACK 0.  MAKE SURE  CONTROLLER  TIME  OUT
                                        SETS.

                                ************************************************************
                                **ERRORS DURING DATA TRANSFER
                                ************************************************************

                                TEST 76 LIMIT DETECT ON DATA TRANSFER

                                        ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.   ISSUE
                                        A  SEEK  TO CYLINDER 2 WITH BAD PARITY.  ISSUE A DRIVE
                                        CLEAR.  ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
                                        TRACK 0, HEAD 0.  SEEK INCOMPLETE BECAUSE OF OUTER

                                TEST 77 PROGRAMMING ERROR

                                        ISSUE A SUBSYSTEM CLEAR.  ISSUE A  READ  DATA  OF  400
                                        WORDS  ON  CYLINDER 0, TRACK 0, SECTOR 0.  DURING READ
                                        ISSUE A  WRITE  TO  THE  SPARE  REGISTER.   MAKE  SURE
                                        PROGRAMMING ERROR SETS.

                                TEST 100 ECC HARD

                                        ISSUE A SUBSYSTEM CLEAR.  ISSUE  A  WRITE  DATA  WORDS
                                        CONSISTING OF 177777 TO CYLINDER 0, TRACK 0, SECTOR 0.
                                        NOW WRITE ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR  0.
                                        DURING   WRITE  ISSUE  CONTROLLER  CLEAR.   MAKE  SURE
                                        PROGRAMMING ERROR IS RESET.  NOW ISSUE A READ DATA  TO
                                        CYLINDER  0,  TRACK  0,  HEAD  0 AND AN ECC HARD ERROR
                                        SHOULD SET.

                                TEST 101 DRIVE TIMING ERROR
```

```
                        ISSUE A SUBSYSTEM CLEAR.  SEEK TO CYLINDER 632.  ISSUE
                        A  RECALIBRATE  BUT  DO NOT WAIT FOR SECOND INTERRUPT.
                        PUT CONTROLLER  IN  DIAGNOSTIC  MODE.  ISSUE  A  READ
                        HEADER OF CYLINDER 0, TRACK 0.  CLOCK THROUGH SEEK AND
                        DRIVE  CLEAR  MESSAGES.  TURN  OFF  DIAGNOSTIC  MODE.
                        DRIVE  TIMING  ERROR  SHOULD  SET  BECAUSE  OF NO DATA
                        TRANSISTIONS ON DATA LINE.

                ;*******************************************************************
                ;*ERROR FORCING IN DRIVE
                ;*******************************************************************

                TEST 102 INITIALIZE CLEARING SACK

                        ISSUE A SUBSYSTEM CLEAR.  SELECT AN  AVAILABLE  DRIVE.
                        ISSUE A SUBSYSTEM CLEAR.  PUT CONTROLLER IN DIAGNOSTIC
                        MODE.  ISSUE A SELECT COMMAND WITH MESSAGE ID = 3  AND
                        DRIVE  SELECTED  =  0.  CLOCK THROUGH PHASE ADDRESS 6.
                        TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD  ERROR
                        DOES NOT SET.

                TEST 103 DRIVE OFF TRACK

                        ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.  ISSUE
                        OFFSET  OF  +1200  MICRO-INCHES.  PUT  CONTROLLER  IN
                        DIAGNOSTIC MODE.  ISSUE A WRITE  DATA  OF  1  WORD  TO
                        CYLINDER 0, TRACK 0, SECTOR 0.  CLOCK THROUGH SEEK AND
                        DRIVE  CLEAR  MESSAGES.  TURN  OFF  DIAGNOSTIC  MODE.
                        DRIVE  OFF  TRACK SHOULD SET IN DRIVE.  REPEAT FOR ALL
                        AVAILIABLE DRIVES.

                TEST 104 FILE UNSAFE

                        ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECLAIBRATE.  ISSUE
                        A  READ  HEAD  OF  CYLINDER  0,  TRACK  0 IN 24 SECTOR
                        FORMAT.  DO A SELECT COMMAND IN 26 SECTOR FORMAT.  PUT
                        CONTROLLER  IN  DIAGNOSTIC MODE.  ISSUE A WRITE HEADER
                        TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR  FORMAT.
                        CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
                        INDEX PULSE.  TURN OFF DIAGNOSTIC MODE.  FILE  UNSAFE
                        SHOULD  SET  BECAUSE  OF  ATTEMPTING  TO WRITE THROUGH
                        SECTOR PULSE.  REPEAT FOR ALL AVAILIABLE DRIVES.

                TEST 105 DUMMY TEST FOR PREVIOUS TEST EXIT

                        THIS TEST IS PRESENT TO MAKE  $SWO8TB  TABLE  HAVE  AN
                        ENTRY WHICH RELATES TO "NEWDRV".  THIS IS NECESSARY IF
                        AN ERROR OCCURS IN THE PRECEEDING TEST AND THAT  ERROR
                        ABORTS  THE  TEST.  IF THIS TEST WERE NOT PRESENT, THE
                        PROGRAM WOULD SKIP THE "NEWDRV" ROUTINE AND GO TO  THE
                        TEST FOLLOWING "NEWDRV".

                        IN ADDITION, THE DRIVE IS CLEARED AND  THE  HEADS  ARE
```

ALLOWED TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS FROM THE DRIVE BECOMING READY AT A LATER TIME.

****
**MULTI-DRIVE OPERATIONS
****

TEST 106 RESET ATTENTIONS WITH UNIBUS INIT

DO A RECALIBRATE ON ALL AVAILIABLE DRIVES. ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.

TEST 107 RESET ATTENTIONS WITH SUBSYSTEM CLEAR

DO A RECALIBRATE ON ALL AVAILABLE DRIVES. ISSUE A SUBSYSTEM CLEAR. MAKE SURE ALL ATTENTIONS RESET.

TEST 110 SVAL AND ATTENTION FROM OTHER DRIVE

DO A RECALIBRATE ON ONE AVAILABLE DRIVE. DO A SELECT ON ANOTHER AVAILABLE DRIVE. MAKE SURE STATUS VALID IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS CHANGE REMAINS RESET.

REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.

NOTE: THIS TEST WILL ONLY BE DONE IF AT LEAST TWO DRIVES ARE AVAILABLE.

TEST 111 OVERLAPPED OPERATIONS

DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B. ISSUE A SEEK ON DRIVE A TO CYLINDER 1. IMMEDIATELY ISSUE A WRITE DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B. AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD BE SET AND DRIVE A HAS ATTENTION SET.

REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.

NOTE: IF ONLY ONE DRIVE IS AVAILABLE THE TEST WILL NOT BE DONE.

5.0 ERROR REPORTING

A DETAILED DESCRIPTION OF THE ERROR FORMATS AND REPORTS CONTENTS IS GIVEN HERE. THIS IS ESSENTIALLY THE SAME AS CAN BE FOUND IN THE LISTING COMMENTS UNDER THE ERROR POINTER TABLE.

ERROR POINTER TABLE:

```
1067
1068
1069
1070                    THIS TABLE CONTAINS THE INFORMATION FOR EACH  ERROR  THAT  CAN
1071                    OCCUR.  THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER
1072                    FOUND IN LOCATION SITEMB.  THIS NUMBER INDICATES WHICH ITEM IN
1073                    THE TABLE IS PERTINENT.
1074                        NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1075                        NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS
1076                               EXPLAINED AS FOLLOWS:
1077                        EM              ;;POINTS TO THE ERROR MESSAGE
1078                        DH              ;;POINTS TO THE DATA HEADER
1079                        DT              ;;POINTS TO THE DATA
1080                        DF              ;;POINTS TO THE DATA FORMAT
1081                    EM AND DH ARE ASCIZ DATA.  EM IS ALWAYS A MESSAGE BUT  DH  CAN
1082                    BE  A  MESSAGE  OR  A  SET OF COLUMN LABELS SPACED ACCROSS THE
1083                    PAGE.  DT IS A STRING OF WORDS THAT POINT TO  THE  DATA  TO  BE
1084                    TYPED,  AND DF IS A STRING OF WORDS THAT TELL HOW THE DT WORDS
1085                    ARE TO BE TYPED.  IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A
1086                    PARTICULAR  FORMAT,  IT  IS  REPLACED WITH A ZERO.  THE NORMAL
1087                    USAGE OF THE ERROR TABLE IS TO HAVE A  TABLE  ENTRY  FOR  EACH
1088                    ERROR MESSAGE THAT CAN OCCUR.  IN THE INTEREST OF ECONOMICS OF
1089                    CORE MEMORY, THIS PROGRAM USES THE ERROR TABLE  IN  A  SIGHTLY
1090                    DIFFERENT MANNER AS DESCRIBED BELOW.
1091
1092                    THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT  FORMATS  THAT
1093                    ARE  USED  THROUGHOUT  THE  PROGRAM.  WHEN  AN ERROR IS TO BE
1094                    REPORTED, THE TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT  IS
1095                    CHOSEN  FROM  THE DEFINED SET.  THE TABLE ENTRY CHOSEN IS THEN
1096                    ALTERED BY CHANGING THE FIRST (AND POSSIBLY THE  SECOND)  WORD
1097                    TO  CONTAIN  THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE
1098                    MESSAGE PORTION OF THE REPORT.  THE DATA FIELDS FOR THAT ENTRY
1099                    ARE NEVER CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.
1100
1101                    THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW  WITH
1102                    THE  DEFINITION  OF  THE  ENTRY.  ALL DATA FIELDS ARE TYPED IN
1103                    OCTAL.
1104
1105                    ERROR ITEM 1
1106                        (MESSAGE)
1107                        TST NUM ERR PC  DRIVE
1108                        $TESTN  $ERRPC  DRVNUM
1109
1110                    ERROR ITEM 2
1111                        (MESSAGE)
1112                        (MESSAGE)
1113                        TST NUM ERR PC  DRIVE
1114                        $TESTN  $ERRPC  DRVNUM
1115                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
1116                          T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
1117                        RKBA    RKWC
1118                        T.BA    T.WC
1119
1120                    ERROR ITEM 3
1121                        (MESSAGE)
1122                        TST NUM ERR PC  DRIVE
```

```
                        $TESTN  $ERRPC  DRVNUM
                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1
                          T.CS1   T.CS2   T.DS    T.ER    T.ASF   T.MR1

ERROR ITEMS 4,5,6, AND 7 ARE USED TO REPORT  ERRORS  THAT  ARE
THE  RESULT  OF  A HARDWARE ERROR INDICATOR BEING SET WHEN NOT
EXPECTED, NOT SET WHEN IT IS EXPECTED,  OR  BOTH.   THE  ERROR
REPORT WILL CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2)
ALL THE EXPECTED ERRORS THAT DID NOT OCCUR,  OR  (3)  ALL  THE
EXPEDTED BUT NOT SET ERRORS AND THE UNEXPECTED BUT SET ERRORS.

THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR  THE  REPORT.
INCLUDED  IN  THE  REPORT WILL BE ONE OR MORE OF THE FOLLOWING
STATEMENTS:

"THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"
"THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"
"THE ABOVE ARE ERRORS SET IN OPERATION:"

PRECEEDING ANY ONE OF THESE LINES WILL BE ONE  OR  MORE  LINES
THAT SPECIFY THE EXACT ERROR.  FOLLOWING THE LAST LINE WILL BE
A LINE THAT IDENTIFIES THE OPERATION BEING PERFORMED.

EXAMPLE:
NON-EXISTANT DRIVE
THE ABOVE ARE ERRORS SET IN OPERATION:
DRIVE SELECT

(ADDITIONAL LINES OF INFORMATION)

THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.

EXAMPLE:
NON-EXISTANT DRIVE
THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
DRIVE SELECT

(ADDITIONAL LINES OF INFORMATION)

THIS IS THE RESULT OF AN EXPECTED ERROR THAT  DID  NOT  OCCUR,
I.E.  A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.

EXAMPLE:
NON-EXISTANT MEMORY
THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
UNIBUS PARITY ERROR
THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
WRITE DATA

(ADDITIONAL LINES OF INFORMATION)

THIS IS AN EXAMPLE  OF  NON-EXISTANT  MEMORY  BEING  SET  WHEN
UNIBUS PARITY ERROR WAS EXPECTED.
```

```
                                        ERROR ITEM 4
                                            (DESCRIPTION OF ERROR)
                                            ERROR IN OPERATION
                                            (DESCRIPTION OF OPERATION)
                                            TST NUM ERR PC  DRIVE
                                            STESTN  SERRPC  DRVNUM
                                              RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
                                              T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
                                            RKBA    RKWC
                                            T.BA    T.WC
                                             A00     B00    A01    B01    A02    B02    A03    B03
                                             $REG10  $REG11 $REG12 $REG13 $REG14 $REG15 $REG16 $REG17

                                        THE ERRORS REPORTED BY THIS FORMAT ARE:
                                             CONTROLLER DETECTED DRIVE BUS ERROR
                                             DRIVE DETECTED DRIVE BUS ERROR
                                             SEEK INCOMPLETE
                                             NON-EXECUTABLE DRIVE FUNCTION
                                             DRIVE TIMING ERROR
                                             DRIVE UNSAFE
                                             AC LOW
                                             SPINDLE SPEED LOSS
                                             DRIVE OFF TRACK
                                             ILLEGAL DRIVE ADDRESS ERROR
                                             CYLINDER OVERFLOW
                                             DRIVE TYPE ERROR
                                             FORMAT ERROR
                                             WRITE LOCK ERROR

                                        ERROR ITEM 5
                                        THIS ENTRY IS THE SAME AS  ITEM  4  WITH  THE  ADDITION  OF  A
                                        MESSAGE THAT FOLLOWS.  THIS MESSAGE IS:

                                        "ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"

                                        THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR A00
                                        THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.

                                        ERROR ITEM 6

                                            (DESCRIPTION OF ERROR)
                                            ERROR IN OPERATION
                                            (DESCRIPTION OF OPERATION)
                                            TST NUM ERR PC  DRIVE
                                            STESTN  SERRPC  DRVNUM
                                              RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
                                              T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
                                            RKBA    RKWC
                                            T.BA    T.WC

                                        THE ERRORS REPORTED BY THIS FORMAT ARE:
                                            DATA CHECK
                                            WRITE CHECK
```

```
1235
1236
1237
1238                                        ECC HARD
1239                                        DATA LATE
1240                                        OPERATION INCOMPLETE
1241                                        HEADER VRC ERROR
1242                                        BAD SECTOR ERROR
1243
1244                                    ERROR ITEM 7
1245                                        (DESCRIPTION OF ERROR)
1246                                        ERROR IN OPERATION
1247                                        (DESCRIPTION OF OPERATION)
1248                                        TST NUM ERR PC  DRIVE
1249                                        $TESTN  $ERRPC  DRVNUM
1250                                          RKCS1   RKCS2   RKDS    RKER    RKASOF
1251                                          T.CS1   T.CS2   T.DS    T.ER    T.ASOF
1252
1253                                          THE ERRORS REPORTED BY THIS FORMAT ARE:
1254                                          NON-EXISTANT DRIVE
1255                                          NON-EXISTANT MEMORY
1256                                          CONTROLLER TIME OUT
1257                                          UNIT FIELD ERROR
1258                                          MULTIPLE DRIVE SELECT
1259                                          PROGRAMMING ERROR
1260                                          UNIBUS PARITY ERROR
1261                                          ILLEGAL FUNCTION CODE
1262
1263                                    DESCRIPTON OF OPERATION CAN BE ANY COMMAND,  EITHER  LEGAL  OR
1264                                    ILLEGAL.
1265
1266                                    ERROR ITEM 10
1267                                        (DESCRIPTION OF ERROR)
1268                                        ERROR AT COMPLETION OF OPERATION
1269                                        (DESCRIPITON OF OPERATION)
1270                                        TST NUM ERR PC  DRIVE
1271                                        $TESTN  ERRPC   DRVNUM
1272                                          EXPT    IS
1273                                          $REG10  $REG11
1274
1275                                    THE ERRORS REPORTED BY THIS FORMAT ARE  SOFTWARE  DETECTED  BY
1276                                    COMPARING  EXPECTED RESULTS WITH ACTUAL RESUL S.  THE SPECIFIC
1277                                    ERRORS ARE:
1278                                          WORD COUNT INCORRECT
1279                                          BUS ADDRESS INCORPECT
1280                                          CYLINDER ADDRESS INCORRECT
1281                                          TRACK ADDRESS INCORRECT
1282                                          SECTOR ADDRESS INCORRECT
1283
1284                                    ERROR ITEM 11
1285                                        (ERROR INDICATOR OR STATUS BIT)
1286                                        NOT SET AS A RESULT OF
1287                                        (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION
1288                                        TST NUM ERR PC  DRIVE
1289                                        $TESTN  $ERRPC  DRVNUM
1290                                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1
```

```
                                        T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.MR1

                                ERROR ITEM 12
                                THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:

                                "NOT RESET AS A RESULT OF"

                                ERROR ITEM 13
                                THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:

                                "SET AS A RESULT OF"

                                ERROR ITEM 14
                                THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:

                                "RESEY AS A RESULT OF"

                                ERROR ITEM 15
                                    (HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)
                                    TST NUM ERR PC  DRIVE
                                    $TESTN  $ERRPC  DRVNUM
                                      GOOD    BAD     WORD NUM
                                      $REG10  $REG11  $REG12

                                ERROR ITEM 16
                                ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15.

                        6.0     SUBROUTINES

                                IN THE INTEREST OF CONSERVING MEMORY, IT IS NECESSARY TO  MAKE
                                EXTENSIVE  USE  OF  SUBROUTINES.  HOWEVER, IN THE INTEREST OF
                                PRESERVING CODE READABILITY, SUBROUTINE NAMING IS  DESCRIPTIVE
                                OF  THE  FUNCTION  PERFORMED.  THE SUBROUTINE FUNCTION IS KEPT
                                SMALL AND IN GENERAL A SUBROUTINE ONLY PERFORMS ONE  FUNCTION,
                                I.E., LOAD THE RK611 REGISTER AND START AN OPERATION (TLOADRK)
                                OR WAIT A SPECIFIED NUMBER OF MILLISECONDS  FOR  AN  INTERRUPT
                                (TWATNN  WHERE  NN VARIES FROM CALL TO CALL AND IS THE TIME TO
                                WAIT).  THE FOLLOWING IS A DESCRIPTION OF THE SUBROUTINES  NOT
                                PROVIDED BY SYSMAC:

                                ;*****************************************************************
                                ;LINE CLOCK CALIBRATE
                                ;*****************************************************************

                                        WAITS FOR A LINE  CLOCK  INTERRUPT  TO  CALIBRATE  THE
                                        INTERRUPTS TO A MEANINGFUL TIME VALUE.  IN ADDITION IT
                                        PRESETS THE TIMCNT IF THERE IS NO LINE CLOCK.   TIMCNT
                                        IS USED IN THE LINE CLOCK SIMULATOR.

                                        CALL:   JSR     PC,CLKCAL

                                ;*****************************************************************
                                ;OPTION PRESENT TEST AND SETUP
```

```
1347
1348
1349
1350                    ;*****************************************************************
1351
1352                    ;       THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND
1353                    ;       THE LINE CLOCK ARE ON THE SYSTEM.  FLAGS ARE SET IF
1354                    ;       PRESENT;  CLEARED OTHERWISE, AND THE APPROPRIATE
1355                    ;       INTERRUPT VECTORS ARE SET UP.
1356
1357                    ;       CALL:   JSR     PC,OPTTST
1358
1359                    ;*****************************************************************
1360                    ;LINE CLOCK SIMULATION ROUTINE
1361                    ;*****************************************************************
1362
1363                    ;       THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK.  TO
1364                    ;       DO THIS THE VALUE STORED IN MILCNT IS USED AS THE BASE
1365                    ;       AND REPRESENTS THE NUMBER OF TIMES A DECREMENT AND
1366                    ;       BRANCH LOOP CAN BE DONE IN 1 MILLISECOND.  THE TIMCNT
1367                    ;       VALUE IS DECREMENTED AND IF IT REACHED 0 THE LINE
1368                    ;       CLOCK TICK COUNTER IS BUMPED.  THEN THE TIMCNT IS
1369                    ;       RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
1370                    ;       TICK).
1371
1372                    ;       THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1
1373                    ;       MILLISECOND AND BUMPS THE LINE CLOCK TICK COUNTER FOR
1374                    ;       EACH 16 CALLS.
1375
1376                    ;       CALL:   JSR     PC,MYTIME
1377
1378                    ;*****************************************************************
1379                    ;WAIT FOR INTERRUPT ROUTINE
1380                    ;*****************************************************************
1381
1382                    ;       THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS.
1383                    ;       THE CALL SPECIFIES HOW MANY TICKS OF THE LINE CLOCK
1384                    ;       ARE TO ELAPSE WHILE WAITING FOR INTERRUPT.  IF
1385                    ;       INTERRUPT DOES NOT OCCUR IN THAT PERIOD OF TIME, AN
1386                    ;       ERROR MESSAGE IS PREPARED (BUT NOT PRINTED IN THE
1387                    ;       ROUTINE)AND THEN RETURNS TO THE LOCATION FOLLOWING THE
1388                    ;       CALL.  IF INTERRUPT OCCURS THE RETURN IS BUMPED BY 2.
1389                    ;       NORMALLY AN ERROR CALL WILL BE IN THE LOCATION AFTER
1390                    ;       THE CALL TO INTERRUPT WAIT.
1391
1392                    ;       CALL:   TWAT16 THROUGH TWAT159, TWAT1S, TWAT2S,
1393                    ;       TWAT8S, AND TWATIM
1394
1395                    ;*****************************************************************
1396                    ;"L" REGISTER LOADING ROUTINE
1397                    ;*****************************************************************
1398
1399                    ;       THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
1400                    ;       THE "L" REGISTERS L.CS1-L.DCYL.  L.MRI IS NOT LOADED
1401                    ;       IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED FOR AN
1402                    ;       OPERATION.  L.CS2 IS LOADED FROM DRVNUM.
```

```
                        CALL:   JSR     R4,LRLOAD
                                COMMAND
                                WORD COUNT
                                BUS ADDRESS
                                .BYTE   SECTOR ADDRESS
                                .BYTE   TRACK ADDRESS
                                CYLINDER ADDRESS

                ;*******************************************************************
                ;LOAD RK611 FOR OPERATION
                ;*******************************************************************

                        THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611
                        REGISTER.   THIS  IS  A  STRAIGHT  TRANSFER  WITH  NO
                        CHECKING OR MANIPULATION  OF  THE  REGISTER  CONTENTS.
                        L.CS1  IS  TRANSFERRED  LAST AS IT SHOULD BE IF THE GO
                        BIT IS SET.

                        CALL:   TLOADRK

                ;*******************************************************************
                ;STORE RK611 REGISTERS
                ;*******************************************************************

                        ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
                        WITH  THE  EXCEPTION  OF  THE DATA BUFFER WHICH IS NOT
                        STORED IN THIS ROUTINE.

                        CALL:   TGETRK

                ;*******************************************************************
                ;BIT COUNTER IN A WORD
                ;*******************************************************************

                        THE WORD WHOSE BITS MUST BE COUNTED IS PLACED  ON  THE
                        STACK  BY  THE  CALLING  ROUTINE.   THE NUMBER OF BITS
                        FOUND IN THE WORD ARE PASSED BACK ON THE STACK.

                        CALL:   JSR     R4,BITCNT

                ;*******************************************************************
                ;MAINTENANCE CLOCK ROUTINE
                ;*******************************************************************

                        THE PARAMETERS PASSED TO THIS ROUTINE ARE  LOCATED  IN
                        THE  ADDRESS  AFTER THE CALL.  THE FIRST BYTE CANTAINS
                        THE NUMBER OF PHASES  ADDRESSES  THE  CALLING  ROUTINE
                        WANTS  THE  CONTROLLER  CLOCKED THROUGH AND THE SECOND
                        BYTE CONTAINS THE NUMBER OF CLOCK TRANSITIONS (PARTIAL
                        PHASES) TO BE DONE.

                        CALL:   JSR     R4,MCLOCK
                                .BYTE           ;NUMBER OF CLOCK TRANSISTIONS
```

```
                        .BYTE           ;NUMBER OF PHASE ADDRESSES

;*****************************************************************
;READ AND SORT HEADERS
;*****************************************************************

        THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY THE
        FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
        ASSCENDING ORDER.  CONTROLLER ERRORS ARE CHECKED IN
        THE READ HEADER OPERATION AND DATA LATE IS CHECKED
        AFTER EACH READ OF THE DATA BUFFER.

        CALL:   JSR     R4,RDSTHD
                TCHKOP              ;RETURN POINT IF CERR IN READ
                                    ;HDR
                ERROR   4           ;OR 5, 6, 7
                ERROR   13          ;RETURN IF DATA LATE IN DB
                                    ;UNLOAD
                ERROR   2           ;RETURN IF TO SLOW OR
                                    ;IF HDR 0 NOT FOUND

;*****************************************************************
;GET DRIVE STATUS
;*****************************************************************

        THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT
        IN $REG10 THROUGH $REG17.  THESE REGISTORS ARE FIRST
        CLEARED TO ALL ONES AND THEN IF ERROR OCCURS WHILE
        GETTING STATUS, THE 1'S ARE LEFT IN THE REGISTERS.

        CALL:   JSR     R4,GETDRS
                BR      ERROR PROCESSING        ERROR RETURN
                BR      NO ERROR PROCESSING     GOOD RETURN

;*****************************************************************
;SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
;*****************************************************************

        THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
        COMMAND.  CERR AND DI ARE MONITORED FOR A SHORT PEIROD
        OF TIME DURING WHICH THEY SHOULD BOTH RESET.

        IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.

        IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN
        APPROPRIATE ERROR MESSAGE IS PREPARED AND REPORTED
        WHEN THE ROUTINE RETURN TO THE CALL.  IF EVERY THING
        IS GOOD, THE RETURN SKIPS OVER THE ERROR CALL AND TEST
        ABORT.

        THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY AN
        ERROR MESSAGE AND BRANCH TO END OF TEST.  THIS IS DONE
        BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
```

```
1515
1516
1517
1518                                    THE TEST.
1519
1520                                    CALL:   TSSINIT
1521
1522                            ;*******************************************************************
1523                            ;WORD COUNT AT END OF OPERATION CHECK
1524                            ;*******************************************************************
1525
1526                                    THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE
1527                                    FOR  THE  WORD  COUNT  AGAINST THE SUPPLIED VALUE.  IF
1528                                    UNEQUAL, THE ERROR FLAG (WCERR)  IS  SET  IN  GROUP  4
1529                                    ERROR FLAGS (GRP4ER)
1530
1531                                    CALL:   JSR     R4,CHKWC
1532                                            .WORD           ;EXPECTED WC VALUE
1533
1534                            ;*******************************************************************
1535                            ;BUS  ADDRESS AT END OF OPERATION CHECK
1536                            ;*******************************************************************
1537
1538                                    THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT  THE
1539                                    END  OF  A  TRANSFER BY USING THE INITIAL BUS ADDRESS,
1540                                    ADDING IN THE INITIAL WORD COUNT, AND SUBTRACTING  ANY
1541                                    RESIDUAL  WORD  COUNT.   IF  THIS COMPUTED BA DOES NOT
1542                                    EQUAL THE CONTENTS OF RKBA AN ERROR  FLAG  (BAERR)  IS
1543                                    SET IN GROUP 4 ERROR FIELD (GRP4ER)
1544
1545                                    IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED
1546                                    BUS ADDRESS IS THE STARTING BUS ADDRESS.
1547
1548                                    CALL:   JSR     R4,CHKBA
1549
1550                            ;*******************************************************************
1551                            ;CYLINDER,TRACK,SECTOR TEST AT END OF OPERATION
1552                            ;*******************************************************************
1553
1554                                    THIS ROUTINE CHECKS THAT THE CONTENTS  OF  THE  RKDCYL
1555                                    AND RKDA ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE
1556                                    END  OF  THE  OPERATION.   THE  CONTENTS  OF  THE LOAD
1557                                    REGISTER  STORAGE  ARE  COUNTED ON TO HAVE THE INITIAL
1558                                    VALUES TO MAKE THE NECESSARY CALCULATION.
1559
1560                                    ALL THREE VALUES ARE GENERATED AND STORED IN  EXPECTED
1561                                    VALUES  STORAGE  EXPCYL,  EXPTRK,  EXPSEC.   ALL 3 ARE
1562                                    CHECKED  AND  IF  ONE   OR   MORE   ARE   WRONG,   THE
1563                                    CORRESPONDING BIT IN THE ERROR FLAGS FIELD (GRP4ER) IS
1564                                    SET.
1565
1566                                    CALL:   JSR     R4,CHKCTS
1567
1568                            ;*******************************************************************
1569                            ;OPERATION CHECK ROUTINE
1570                            ;*******************************************************************
```

```
1571
1572
1573
1574                    THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME
1575                    SOFTWARE ERRORS ARE CHECKED.  THE GENERAL PROCEDURE
1576                    FLOW IS AS FOLLOWS:  THE ROUTINE IS CALLED WITH A TRAP
1577                    (TCHKOP).  THE LOCATION FOLLOWING THE TRAP CALL WILL
1578                    HAVE AN ERROR TRAP WHICH THE ROUTINE WILL BYPASS IF NO
1579                    ERROR IS FOUND.  IF AN ERROR IS DETECTED, THE ERROR
1580                    TRAP CALL IS MODIFIED BY THIS ROUTINE SUCH THAT THE
1581                    ERROR TABLE ITEM WILL BE THE PROPER ITEM FOR THE
1582                    FORMAT REQUIRED BY THIS ERROR.  THE ERROR TRAP WILL BE
1583                    MADE EITHER ERROR 4,5,6, 7, OR 10.  REFER TO THE ERROR
1584                    ITEM TABLE FOR A DESCRIPTION OF THE FORMAT AND WHICH
1585                    ERRORS ARE DISPLAYED IN WHAT FORMAT.
1586
1587                    FOR NO EXPECTED ERRORS:
1588                    CALL:   TCHKOP
1589
1590                    FOR EXPECTED ERRORS:
1591                    CALL:   TCHKWE
1592                            .WORD           ;GROUP 1 EXPECTED ERRORS
1593                            .WORD           ;GROUP 2 EXPECTED ERRORS
1594                            .WORD           ;GROUP 3 EXPECTED ERRORS
1595
1596                    WHERE EACH BIT IN THE THREE WORDS FOLLOWING THE CALL
1597                    REPRESENT A SPECIFIC ERROR.  THE BIT ASSIGNMENTS ARE
1598                    GIVEN BELOW:
1599
1600                            GROUP 1 ERRORS:
1601                            BIT 0 - CONTROLLER DETECTED DRIVE BUS
1602                                    PARITY ERROR
1603                            BIT 1 - SEEK INCOMPLETE
1604                            BIT 2 - NON-EXECUTABLE DRIVE FUNCTION
1605                            BIT 3 - DRIVE DETECTED DRIVE BUS PARITY ERROR
1606                            BIT 4 - FORMAT ERROR
1607                            BIT 5 - DRIVE TYPE ERROR
1608                            BIT 6 - AC LOW ERROR
1609                            BIT 7 - SPEED LOSS ERROR
1610                            BIT 8 - DRIVE OFF TRACK ERROR
1611                            BIT 9 - CYLINDER OVERFLOW ERROR
1612                            BIT 10 - ILLEGAL DISK ADDRESS ERROR
1613                            BIT 11 - WRITE LOCK ERROR
1614                            BIT 12 - DRIVE TIMING ERROR
1615                            BIT 13 - NO CERR WITH OTHER ERROR SET ERROR
1616                            BIT 14 - DRIVE UNSAFE ERROR
1617                            BIT 15 - CERR BUT NO OTHER ERROR SET ERROR
1618
1619                            GROUP 2 ERRORS:
1620                            BIT 0 - ECC HARD ERROR
1621                            BIT 1 - DATA CHECK ERROR
1622                            BIT 2 - WRITE CHECK ERROR
1623                            BIT 3 - DATA LATE ERROR
1624                            BIT 4 - OPERATION INCOMPLETE ERROR
1625                            BIT 5 - HEADER VRC ERROR
1626                            BIT 6 - BAD SECTOR ERROR
```

```
                                        BROUP 2 ERRORS:
                                        BIT 0 - NON-EXISTAND DRIVE ERROR
                                        BIT 1 - CONTROLLER TIMEOUT ERROR
                                        BIT 2 - UNIT FIELD ERROR
                                        BIT 3 - MULTIPLE DRIVE SELECT ERROR
                                        BIT 4 - PROGRAMMING ERROR
                                        BIT 5 - NON-EXISTANT MEMORY ERROR
                                        BIT 6 - UNIBUS PARITY ERROR
                                        BIT 7 - ILLEGAL FUNTION ERROR

                                ;*******************************************************************
                                ;BAD SECTOR CHECK
                                ;*******************************************************************

                                        THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
                                        CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED
                                        THEREIN THAT HAVE THE CYLINDER AND TRACK ADDRESS
                                        SPECIFIED IN L.DCYL AND L.DT.  IF A SECTOR IS FOUND IN
                                        THIS FIELD THAT IS BAD FOR THAT CYLINDER AND TRACK,
                                        THE SECTOR NUMBER IS PLACED ON THE STACK.  THE TOTAL
                                        NUMBER OF BAD SECTORS IS PLACED ON THE STACK AFTER THE
                                        ENTIRE FIELD IS SEARCHED.

                                        CALL:   JSR     R4,BDSRCK
                                                ;ADDRESS OF FIELD TO BE SEARCHED;

                                ;*******************************************************************
                                ;DATA GENERATION AND COMPARE ROUTINE
                                ;*******************************************************************

                                        CALLS:  JSR     R4,GENCOM
                                                CONTROL WORD

                                                JSR     R4,GENCOM
                                                CONTROL WORD
                                                LENGTH

                                                JSR     R4,GENCOM
                                                CONTROL WORD
                                                RELOCATION CONSTANT
                                                LENGTH

                                          RETURN:       RTS     R4

                                        R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
                                        THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
                                        CONTROL WORD.  THIS IS UNCONDITIONAL.

                                        THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE
                                        LENGTH IF THE OPERATION REQUIRES DATA COMPARE AND DATA
                                        MISCOMPARED.  IF DATA IS TO BE GENERATED ONLY OR NO
                                        DATA COMPARE ERRORS OCCURRED, THE RETURN IS TO LENGTH
                                        +4.
```

```
1683
1684
1685
1686                    THE THIRD CALL IS IDENTICAL TO THE SECOND.
1687
1688                    DEFINITION OF CONTROL WORD:
1689
1690                    BIT 15     - DO COMPARE OPERATION OF IBUFF (SOURCE) TO
1691                                 OBUFF (DESTINATION).  EXPECTED VALUES ARE
1692                                 IN OBUFF (DESTINATION.
1693                    BIT 14     - RESUME COMPARE OPERATION FROM POINT  LEFT
1694                                 BY LAST COMPARE.
1695                    BIT 13     - INVOKE  MEMORY  MANAGEMENT   FOR   SOURCE
1696                                 (IBUFF).
1697                    BIT 12     - INVOKE MEMORY MANAGEMENT FOR  DESTINATION
1698                                 (OBUFF).
1699                    BIT 11     - REPEAT FIRST  WORD  OF  SELECTED  PATTERN
1700                                 THROUGHOUT OBUFF.
1701                    BIT 10     - CLEAR IBUFF TO PATTERN SELECTED.
1702                    BIT 9      - BUILD HEADERS, CONSIDERING BS FILES
1703                    BIT 8      - BUILD HEADERS, ALL SECTORS INDICATE  GOOD
1704                                 SECTORS.
1705                    BIT 7      - HEADER   OPERATION   SPECIFIED    (EITHER
1706                                 COMPARE OR BUILD).
1707                    BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL  ENCODED.   0
1708                                 INDICATES  NO  DATA  GENERATION, 1 IS ALL
1709                                 ZEROS, AND 7 IS ALL ONES.  OTHER PATTERNS
1710                                 PROVIDED  ARE  PATTERNS 2-6, 8-16.  REFER
1711                                 TO THE PROGRAM LISTING FOR PAT02  THROUGH
1712                                 PAT16.
1713
1714                    EXPLANATION OF CALLS:
1715                        THE CALL WITH CONTROL WORD THE ONLY  PARAMETER  IS
1716                        USED FOR BUILDING OR COMPARING HEADERS OR RESUMING
1717                        A COMPARE OPERATION.
1718
1719                        THE  CALL  WITH  CONTROL  WORD   AND   LENGTH   AS
1720                        PARAMETERS  IS USED FOR DATA GENERATION OR COMPARE
1721                        AND FOR IBUFF INITALIZATION.
1722
1723                        THE CALL WITH CONTROL WORD,  RELOCATION  CONSTANT,
1724                        AND  LENGTH IS USED FOR DATA GENERATION OR COMPARE
1725                        WITH MEMORY MANAGEMENT.
1726
1727                    DESCRIPTION:
1728
1729                    THIS ROUTINE IS MULTI-PURPOSE  AND  WILL  PERFORM  THE
1730                    FOLLOWING:
1731
1732                    A.  BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE.
1733                        THE  ROUTINE  WILL  BUILD  THE HEADERS AS ALL GOOD
1734                        SECTORS (BIT 8)  OR  TAKE  THE  BAD  SECTOR  FILES
1735                        (HARDWARE  OR  SOFTWARE)  FOR  EITHER FORMAT) INTO
1736                        ACCOUNT AND BUILD THE  HEADERS  WITH  THE  SECTORS
1737                        MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK
1738                        ARE LISTED THEREIN (BIT 9).
```

```
1739
1740
1741
1742                    B.  COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT  15).
1743                        THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA.
1744                        A HEADER COMPARE OPERATION MAY BE  SPECIFIED  (BIT
1745                        7)  WHICH  WILL CAUSE THE COMPARE TO BE LIMITED TO
1746                        74(8) OR 102(8)  WORDS  OF  HEADERS.   THE  LENGTH
1747                        DEPENDS  ON THE FORMAT BIT THAT WAS LAST SPECIFIED
1748                        IN L.CS1.  THE HEADERS MAY  BE  BUILT  BEFORE  THE
1749                        COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8
1750                        OR 9).  DATA CAN  ALSO  BE  GENERATED  BEFORE  THE
1751                        COMPARE (NON-ZERO BITS 6-0).
1752
1753                    C.  RESUME COMPARE OPERATION.  IF A COMPARE  OPERATION
1754                        DETECTS  A  MISCOMPARE,  THE  ROUTINE  RETURNS  TO
1755                        CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE
1756                        CAN  BE  RESUMED.   THIS IS DONE BY CALLING GENCOM
1757                        WITH BIT 14 SET IN THE CONTROL WORD.
1758
1759                    D.  DATA   GENERATION   OR   COMPARE   USING    MEMORY
1760                        MANAGEMENT.   MEMORY MANAGEMENT CAN BE INVOKED FOR
1761                        EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH.  IN
1762                        THIS  MANNER, DATA GENERATION CAN BE MADE TO PLACE
1763                        DATA ANYWHERE IN AVAILABLE MEMORY.  LIKEWISE  DATA
1764                        COMPARE  WILL COMPARE THE CONTENTS OF IBUFF TO ANY
1765                        AREA OF AVAILABLE MEMORY.
1766
1767                ;*****************************************************************
1768                ;PHASE LOCKED LOOP CLOCK ADJUSTMENT ROUTINE
1769                ;*****************************************.***********************
1770
1771                        THIS ROUTINE IS ENTERED  VIA  START  LOCATION  220(8).
1772                        THE  PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE
1773                        INTERNAL PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK
1774                        ADJUST   ROUTINE.    THE  ROUTINE  SELECTS  THE  FIRST
1775                        AVAILABLE DRIVE AND SETS  AND  RESETS  THE  DIAGNOSTIC
1776                        MODE BIT IN RKMR1.  INSTRUCTIONS ON WHERE TO SCOPE AND
1777                        WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
1778
1779                        THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
1780                ;
1781                .NLIST  CND,MD,MC,TOC
1782                .LIST   ME
1783                .ENABL  ABS,AMA
1784    167400      $SWR=   167400
1785    000001      $TN=    1
1786                .TITLE  RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
1787                ;*COPYRIGHT (C) 1976 & 1977
1788                ;*DIGITAL EQUIPMENT CORP.
1789                ;*MAYNARD, MASS. 01754
1790                ;*
1791                ;*PROGRAM BY MARV TEGROTENHUIS
1792                ;*
1793                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
1794                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
```

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

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

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

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

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
```

```
                040000          SW14=   40000
                020000          SW13=   20000
                010000          SW12=   10000
                004000          SW11=   4000
                002000          SW10=   2000
                001000          SW09=   1000
                000400          SW08=   400
                000200          SW07=   200
                000100          SW06=   100
                000040          SW05=   40
                000020          SW04=   20
                000010          SW03=   10
                000004          SW02=   4
                000002          SW01=   2
                000001          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

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

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
        000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
```

```
1907    000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1908    000014          TBITVEC=14              ;;"T" BIT
1909    000014          TRTVEC= 14              ;;TRACE TRAP
1910    000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1911    000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1912    000024          PWRVEC= 24              ;;POWER FAIL
1913    000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1914    000034          TRAPVEC=34              ;;"TRAP" TRAP
1915    000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
1916    000064          TPVEC=  64              ;;TTY PRINTER VECTOR
1917    000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1918                    .SBTTL  MEMORY MANAGEMENT DEFINITIONS
1919
1920                    ;*KT11 VECTOR ADDRESS
1921
1922    000250          MMVEC=  250
1923
1924                    ;*KT11 STATUS REGISTER ADDRESSES
1925
1926    177572          SR0=    177572
1927    177574          SR1=    177574
1928    177576          SR2=    177576
1929    172516          SR3=    172516
1930
1931                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1932
1933    172300          KIPDR0= 172300
1934    172302          KIPDR1= 172302
1935    172304          KIPDR2= 172304
1936    172306          KIPDR3= 172306
1937    172310          KIPDR4= 172310
1938    172312          KIPDR5= 172312
1939    172314          KIPDR6= 172314
1940    172316          KIPDR7= 172316
1941
1942                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
1943
1944    172340          KIPAR0= 172340
1945    172342          KIPAR1= 172342
1946    172344          KIPAR2= 172344
1947    172346          KIPAR3= 172346
1948    172350          KIPAR4= 172350
1949    172352          KIPAR5= 172352
1950    172354          KIPAR6= 172354
1951    172356          KIPAR7= 172356
1952
1953    000210          AVECT1= 210                     ;DEFINE RK611 VECTOR INTERRUPT
1954    000240          APRIOR= PR5                     ;DEFINE RK611 PRIORITY
1955    177440          ABASE=  177440                  ;DEFINE RK611 BASE BUS ADDRESS
1956
1957                    .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
1958
1959    000000          RKCS1=  0                       ;CONTROL AND STATUS REGISTER 1
1960    000002          RKWC=   2                       ;WORD COUNT REGISTER
1961    000004          RKBA=   4                       ;BUS ADDRESS REGISTER
1962    000006          RKDA=   6                       ;DESIRED TRACK SECTOR REGISTER
```

```
                                RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
                                RKDS=   12              ;DRIVE STATUS REGISTER
                                RKER=   14              ;ERROR REGISTER
                                RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
                                RKDCYL= 20              ;DESIRED CYLINDER REGISTER
                                RKDB=   24              ;DATA BUFFER
                                RKMR1=  26              ;MAINTENANCE REGISTER 1
                                RKMR2=  34              ;MAINTENANCE REGISTER 2
                                RKMR3=  36              ;MAINTENANCE REGISTER 3
                                RKECPS= 30              ;ECC POSITION INFORMATION
                                RKECPT= 32              ;ECC PATTERN INFORMATION
                                RKSPAR= 22              ;SPARE REGISTER

                                .SBTTL  DRIVE COMMANDS

                                SELDRV= 101             ;SELECT DRIVE
                                PACK=   103             ;PACK ACKNOWLEDGE
                                CLEAR=  105             ;DRIVE CLEAR
                                UNLOAD= 107             ;UNLOAD
                                SRTSPL= 111             ;START SPINDLE
                                RECAL=  113             ;RECALIBRATE
                                OFFSET= 115             ;OFFSET
                                SEEK=   117             ;SEEK
                                RDDATA= 121             ;READ DATA
                                WRDATA= 123             ;WRITE DATA
                                RDHEAD= 125             ;READ HEADER
                                WRHEAD= 127             ;WRITE HEADER AND DATA
                                WRTCHK= 131             ;WRITE CHECK
                                INTR=   300             ;GENERATE INTERRUPT TO CPU

                                .SBTTL  CONTROL AND STATUS REGISTER 1 BITS

000001                          GO=     BIT0            ;GO BIT
000100                          IE=     BIT6            ;INTERRUPT ENABLE
000200                          RDY=    BIT7            ;CONTROLLER READY
000400                          BA16=   BIT8            ;BUS ADDRESS BIT 16
001000                          BA17=   BIT9            ;BUS ADDRESS BIT 17
002000                          CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06)
004000                          CTO=    BIT11           ;CONTROLLER TIMED OUT WAITING FOR
                                                        ;   DRIVE RESPONSE
010000                          CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
020000                          SPAR=   BIT13           ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
040000                          DI=     BIT14           ;DRIVE INTERRUPT
100000                          CERR=   BIT15           ;CONTROLLER ERROR
100000                          CCLR=   BIT15           ;CONTROLLER CLEAR

                                .SBTTL  CONTROL AND STATUS REGISTER 2 BITS

000007                          DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
000010                          RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
000020                          BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
000040                          SCLR=   BIT5            ;CLEAR CONTROLLER AND ALL DRIVES
000100                          IR=     BIT6            ;INPUT READY
000200                          OR=     BIT7            ;OUTPUT READY
000400                          UFE=    BIT8            ;UNIT FIELD ERROR
001000                          MDS=    BIT9            ;MULTIPLE DRIVE SELECT
```

```
2019            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
2020            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
2021            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
2022            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
2023            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
2024            100000          DLT=    BIT15                   ;DATA LATE ERROR
2025
2026                            .SBTTL  ERROR REGISTER BIT DEFINITION
2027
2028            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
2029            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
2030            000004          NXF=    BIT2                    ;NON-EXECUTABLE DRIVE FUNCTION
2031            000010          DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
2032            000020          FMTE=   BIT4                    ;FORMAT ERROR
2033            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
2034            000100          ECH=    BIT6                    ;ECC HARD
2035            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
2036            000400          HVRC=   BIT8                    ;HEADER VRC ERRROR
2037            001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
2038            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
2039            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
2040            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
2041            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
2042            040000          UNS=    BIT14                   ;DRIVE UNSAFE
2043            100000          DCK=    BIT15                   ;DATA CHECK
2044
2045                            .SBTTL  STATUS REGISTER BIT DEFINITION
2046
2047            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
2048                                                            ;  THIS BIT IS RESET)
2049            000004          OFST=   BIT2                    ;DRIVE OFFSET
2050            000010          ACLO=   BIT3                    ;AC LOW
2051            000020          SPDLSS= BIT4                    ;SPEED LOSS
2052            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
2053            000100          VV=     BIT6                    ;VOLUME VALID
2054            000200          DRDY=   BIT7                    ;DRIVE READY
2055            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
2056            004000          WRL=    BIT11                   ;WRITE LOCK
2057            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
2058            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
2059            100000          SVAL=   BIT15                   ;STATUS VALID
2060
2061                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION
2062
2063            000017          MESMSK= 17                      ;MESSAGE MASK
2064
2065            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
2066            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
2067            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
2068            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
2069            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
2070            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
2071            002000          MEWD=   BIT10                   ;MAINTENACNE ENCODED WRITE DATA
2072            004000          PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
2073            010000          PCD=    BIT12                   ;PRECOMPENSATION DELAY
2074            020000          ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
```

```
2075            040000          WRTGAT= BIT14                   ;WRITE GATE
2076            100000          RDGATE= BIT15                   ;READ GATE
2077
2078                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
2079
2080            000040          S.DRA=  BIT5                    ;DRIVE AVAILIABLE
2081            000100          S.VV=   BIT6                    ;VOLUME VALID
2082            000200          S.DRY=  BIT7                    ;DRIVE READY
2083            000400          S.TYPE= BIT8                    ;DRIVE TYPE
2084            001000          S.FORM= BIT9                    ;DRIVE FORMAT
2085            002000          S.OFF=  BIT10                   ;OFFSET
2086            004000          S.WRL=  BIT11                   ;WRITE LOCK
2087            010000          S.SPIN= BIT12                   ;SPINDLE ON
2088            020000          S.PIP=  BIT13                   ;POSITIONING IN PROGRESS
2089            040000          S.DSC=  BIT14                   ;DRIVE STATUS CHANGE
2090
2091                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
2092
2093            000040          S.ICYL= BIT5                    ;ILLEGAL CYLINDER ADDRESS
2094            000100          S.ACLO= BIT6                    ;AC LOW
2095            000200          S.FLT=  BIT7                    ;DRIVE FAULT
2096            000400          S.ILF=  BIT8                    ;ILLEGAL FUNCTION
2097            001000          S.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
2098            002000          S.SKI=  BIT10                   ;SEEK INCOMPLETE
2099            004000          S.WLE=  BIT11                   ;WRITE LOCK ERROR
2100            010000          S.SPLS= BIT12                   ;SPEED LOSS
2101            020000          S.DROT= BIT13                   ;DRIVE OFF TRACK
2102            040000          S.UNS=  BIT14                   ;DRIVE UNSAFE
2103
2104                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
2105
2106            000020          S.XDOK= BIT4                    ;TRANSDUCER OK
2107            000040          S.HOHM= BIT5                    ;HEADS HOME
2108            000100          S.BRHM= BIT6                    ;BRUSHES HOME
2109            000200          S.DOOR= BIT7                    ;DOOR INTERLOCKED
2110            000400          S.CART= BIT8                    ;CARTRAGE INTERLOCK
2111            001000          S.SPOK= BIT9                    ;SPEED OK
2112            002000          S.FWD=  BIT10                   ;FORWARD
2113            004000          S.REV=  BIT11                   ;REVERSE
2114            010000          S.LOAD= BIT12                   ;HEADS LOADING
2115            020000          S.RTZ=  BIT13                   ;RETURN TO ZERO
2116            040000          S.UNLD= BIT14                   ;HEADS UNLOADING
2117
2118                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
2119
2120            000020          S.SECT= BIT4                    ;SECTOR ERROR
2121            000040          S.WCLK= BIT5                    ;WRITE CLOCK AND NO WRITE GATE
2122            000100          S.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
2123            000200          S.HDFL= BIT7                    ;HEAD FAULT
2124            000400          S.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
2125            001000          S.XERR= BIT9                    ;INDEX ERROR
2126            002000          S.DIB=  BIT10                   ;DIBIT ERROR
2127            004000          S.PLO=  BIT11                   ;PLO ERROR
2128            010000          S.NMOV= BIT12                   ;SEEK AND NO MOTION
2129            020000          S.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
2130            040000          S.BRKE= BIT14                   ;SERVO-BRAKE
```

```
                                .SBTTL  COMMON MASKS

                000007          M.DRV=  7                       ;DRIVE CODE
                100000          M.PAR=  BIT15                   ;PARITY
                000003          M.ID=   3                       ;BYTE ID
                017760          M.CDIF= 17760                   ;CYLINDER DIFFERENCE/OFFSET
                017760          M.CADD= 17760                   ;CYLINDER ADDRESS
                077770          M.SER=  77770                   ;DRIVE SERIAL NUMBER
                000760          M.SECT= 760                     ;SECTOR COUNT
                007000          M.HEAD= 7000                    ;HEAD DECODE
                                .SBTTL  TRAP CATCHER

                000000                  .=0
                                ;*ALL UNUSED LOCATIONS OF THE VECTOR AREA CONTAIN
                                ;*A ".+2, IOT" SEQUENCE TO CATCH AND PROCESS ILLEGAL
                                ;*TRAPS AND INTERRUPTS THAT MIGHT OCCUR.
                                ;*THE IOT TRAP WHICH IS TAKEN ON THE ILLEGAL TRAP/INT
                                ;*TRAPS TO THE $SCOPE ROUTINE WHICH (IF THE RETURN PC IS
                                ;*LESS THAN 1002) JUMPS TO THE $ERROR ROUTINE.
                                ;*THE $ERROR ROUTINE WILL REPORT THE ERROR AS FOLLOWS:
                                ;*      PC=YYYYYY UNEXPECTED TRAP TO XXX
                                ;*AND RETURN TO THE PROGRAM AT PC=YYYYYY+2
                                ;*WHERE XXX=LOCATION OF ILLEGAL TRAP
                                ;*      YYYYYY=PC AT TIME OF TRAP
                                ;*NOTE: IF THE PROCESSOR IS NOT AN 11/05 THE PROGRAM
                                ;*      CAN BE STARTED AT ADDRESS 0 AS WELL AS ADDRESS 200.

        000000  000000          $HOCAT: HALT                    ;;HALT
        000002  000737                  BR      .-100           ;;BRANCH TO 177700 & TIME OUT (NOT ON
                                                                ;;11/05)
        000004  001750                  .WORD   START           ;;VECTOR TO STARTING ADDRESS
        000006  000340                  .WORD   340             ;;WITH PRIORITY LEVEL 7
                000174                  .=174
        000174  000000          DISPREG:.WORD   0               ;;SOFTWARE DISPLAY REGISTER
        000176  000000          SWREG:  .WORD   0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRES(ES)
        000200  000137  001750          JMP     @#START ;;GO TO START OF PROGRAM
        000204  000137  001740          JMP     RESTRT          ;JUMP TO RESTART ROUTINE
                000214                  .=214
        000214  000137  001730          JMP     PARM            ;JUMP TO OPERATOR ASSIGNED PARMETERS
                000220                  .=220
        000220  000137  001720          JMP     SETCLK          ;JUMP TO SET CLOCK ROUTINE
                                .SBTTL  APT PARAMETER BLOCK

                                ;;*****************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*****************************************************************
                000224                  .$X=.   ;;SAVE CURRENT LOCATION
                000024                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
        000024  000200                  200     ;;FOR APT START UP
                000044                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
        000044  000224                  $APTHDR ;;POINT TO APT HEADER BLOCK
                000224                  .=.$X   ;;RESET LOCATION COUNTER
                                ;;*****************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
```

```
2187                                    ;INTERFACE SPEC.
2188
2189  000224                            $APTHD:
2190  000224  000000                    $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
2191  000226  001276                    $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
2192  000230  000024                    $TSTM:  .WORD   20.     ;;RUN TIM OF LONGEST TEST
2193  000232  000074                    $PASTM: .WORD   60.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
2194  000234  000740                    $UNITM: .WORD   480.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
2195  000236  000031                            .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .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                  $REGAD: .WORD   0       ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000                  $REG0:  .WORD   0       ;;CONTAINS (($REGAD)+0)
001164  000000                  $REG1:  .WORD   0       ;;CONTAINS (($REGAD)+2)
001166  000000                  $REG2:  .WORD   0       ;;CONTAINS (($REGAD)+4)
001170  000000                  $REG3:  .WORD   0       ;;CONTAINS (($REGAD)+6)
001172  000000                  $REG4:  .WORD   0       ;;CONTAINS (($REGAD)+10)
001174  000000                  $REG5:  .WORD   0       ;;CONTAINS (($REGAD)+12)
001176  000000                  $REG6:  .WORD   0       ;;CONTAINS (($REGAD)+14)
001200  000000                  $REG7:  .WORD   0       ;;CONTAINS (($REGAD)+16)
001202  000000                  $REG10: .WORD   0       ;;CONTAINS (($REGAD)+20)
001204  000000                  $REG11: .WORD   0       ;;CONTAINS (($REGAD)+22)
001206  000000                  $REG12: .WORD   0       ;;CONTAINS (($REGAD)+24)
001210  000000                  $REG13: .WORD   0       ;;CONTAINS (($REGAD)+26)
001212  000000                  $REG14: .WORD   0       ;;CONTAINS (($REGAD)+30)
001214  000000                  $REG15: .WORD   0       ;;CONTAINS (($REGAD)+32)
001216  000000                  $REG16: .WORD   0       ;;CONTAINS (($REGAD)+34)
001220  000000                  $REG17: .WORD   0       ;;CONTAINS (($REGAD)+36)
001222  000000                  $TMP0:  .WORD   0       ;;USER DEFINED
```

```
2252  001224  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
2253  001226  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
2254  001230  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
2255  001232  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
2256  001234  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
2257  001236  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
2258  001240  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
2259  001242  000000                  $TMP10: .WORD   0               ;;USER DEFINED
2260  001244  000000                  $TMP11: .WORD   0               ;;USER DEFINED
2261  001246  000000                  $TMP12: .WORD   0               ;;USER DEFINED
2262  001250  000000                  $TMP13: .WORD   0               ;;USER DEFINED
2263  001252  000000                  $TMP14: .WORD   0               ;;USER DEFINED
2264  001254  000000                  $TMP15: .WORD   0               ;;USER DEFINED
2265  001256  000000                  $TMP16: .WORD   0               ;;USER DEFINED
2266  001260  000000                  $TMP17: .WORD   0               ;;USER DEFINED
2267  001262  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
2268  001264  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
2269  001266  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
2270  001272     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
2271  001273     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
2272  001274  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
2273                                  ;;**************************************************************************
2274                                  .SBTTL  APT MAILBOX-ETABLE
2275
2276                                  ;;**************************************************************************
2277                                  .EVEN
2278  001276                          $MAIL:                          ;;APT MAILBOX
2279  001276  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
2280  001300  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
2281  001302  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
2282  001304  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
2283  001306  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
2284  001310  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
2285  001312  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
2286  001314  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
2287  001316                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
2288  001316     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
2289  001317     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
2290  001320  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
2291  001322  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
2292  001324  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
2293                                  ;*                              BITS 15-11=CPU TYPE
2294                                  ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2295                                  ;*                                      11/70=06,PDQ=07,Q=10
2296                                  ;*                              BIT 10=REAL TIME CLOCK
2297                                  ;*                              BIT  9=FLOATING POINT PROCESSOR
2298                                  ;*                              BIT  8=MEMORY MANAGEMENT
2299  001326     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
2300  001327     000                  $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
2301                                  ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
2302                                  ;*                                      900 NSEC CORE=001
2303                                  ;*                                      300 NSEC BIPOLAR=002
2304                                  ;*                                      500 NSEC MOS=003
2305  001330  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
2306                                  ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2307  001332     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
```

```
 2308  001333  000
 2309  001334  000000  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
 2310  001336  000     $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
 2311  001337  000     $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
 2312  001340  000000  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
 2313  001342  000     $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
 2314  001343  000     $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
 2315  001344  000000  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
 2316  001346  000210  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
 2317  001350  000000  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
 2318  001352  177440  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
 2319  001354  000000  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
 2320  001356  000000  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
 2321  001360          $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
 2322                  $ETEND:
                       .MEXIT
```

```
                                .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

001360                          $ERRTB:

                                ;* EM AND DH ARE ASCIZ DATA.  EM IS ALWAYS A MESSAGE BUT DH
                                ;* CAN BE A MESSAGE OR A SET OF COLUMN LABELS SPACED ACCROSS
                                ;* THE PAGE, DT IS A STRING OF WORDS THAT POINT TO THE DATA TO
                                ;* BE TYPED, AND DF IS A STRING OF WORK THAT TELL HOW THE DT WORDS
                                ;* ARE TO BE TYPED.  IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A
                                ;* PARTICULAR FORMAT, IT IS REPLACED WITH A ZERO.
                                ;*
                                ;* THE NORMAL USAGE OF THE ERROR TABLE IS TO HAVE A TABLE ENTRY FOR
                                ;* EACH ERROR MESSAGE THAT CAN OCCUR.  IN THE INTEREST OF ECONOMICS
                                ;* OF CORE MEMORY, THIS PROGRAM USES THE ERROR TABLE IN A
                                ;* SIGHTLY DIFFERENT MANNERS AS DESCRIBED BELOW.

                                ;* THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT FORMATS THAT ARE USED
                                ;* THROUGHOUT THE PROGRAM.  WHEN AN ERROR IS TO BE REPORTED, THE
                                ;* TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT IS CHOSEN FROM
                                ;* THE DEFINED SET.  THE TABLE ENTRY CHOSEN IS THEN ALTERED
                                ;* BY CHANGING THE FIRST (AND POSSIBLY THE SECOND) WORD TO CONTAIN
                                ;* THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE MESSAGE
                                ;* PORTION OF THE REPORT.  THE DATA FIELDS FOR THAT ENTRY ARE NEVER
                                ;* CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.
                                ;*
                                ;* THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW WITH
                                ;* THE DEFINITION OF THE ENTRY.  ALL DATA FIELDS ARE TYPED IN OCTAL.

                                        ;ERROR ITEM 1
                                ;*      (MESSAGE)
                                ;*      TST NUM ERR PC  DRIVE
                                ;*      $TESTN  $ERRPC  DRVNUM

001360  000000                  EMIN:   .WORD   0
001362  057134                          DH001
001364  060506                          DT001
001366  060600                          DF001

                                        ;ERROR ITEM 2
                                ;*      (MESSAGE)
                                ;*      (MESSAGE)
                                ;*      TST NUM ERR PC  DRIVE
                                ;*      $TESTN  $ERRPC  DRVNUM
                                ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
```

```
                                        ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
                                        ;*      RKBA    RKWC
                                        ;*      T.BA    T.WC
                                        
001370  000000                          EM2N:   .WORD   0
001372  000000                          DH2N:   .WORD   0
001374  060514                                  DT002
001376  060604                                  DF002
                                                ;*ERROR ITEM 3
                                        ;*      (MESSAGE)
                                        ;*      TST NUM ERR PC  DRIVE
                                        ;*      $TESTN  $ERRPC  DRVNUM
                                        ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1
                                        ;*        T.CS1   T.CS2   T.DS    T.ER    T.AST   T.MR1
                                        
001400  000000                          EM3N:   .WORD   0
001402  057162                          DH002A
001404  060464                                  DT003
001406  060624                                  DF003
                                        
                                        ;* ERROR ITEMS 4,5,6,&7 ARE USED TO REPORT ERRORS THAT ARE THE RESULT
                                        ;* OF A HARDWARE ERROR INDICATOR BEING SET WHEN NOT EXPECTED,
                                        ;* NOT SET WHEN IT IS EXPECTED, OR BOTH. THE ERROR REPORT WILL
                                        ;* CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2) ALL THE EXPECTED
                                        ;* ERRORS THAT DID NOT OCCUR, OR (3) ALL THE EXPEDTED BUT NOT SET ERRORS
                                        ;* AND THE UNEXPECTED BUT SET ERRORS.
                                        ;*
                                        ;* THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR THE REPORT.
                                        ;* INCLUDED IN THE REPORT WILL BE ONE OR MORE OF THE FOLLOWING
                                        ;* STATEMENTS:
                                        ;*
                                        ;* "THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"
                                        ;* "THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"
                                        ;* "THE ABOVE ARE ERRORS SET IN OPERATION:"
                                        ;*
                                        ;* PRECEEDING ANY ONE OF THESE LINES WILL BE ONE OR MORE LINES THAT
                                        ;* SPECIFY TJE EXACT ERROR. FOLLOWING THE LAST LINE WILL BE A LINE
                                        ;* THAT IDENTIFIES THE OPERATION BEING PERFORMED.
                                        ;*
                                        ;* EXAMPLE:
                                        ;* NON-EXISTANT DRIVE
                                        ;* THE ABOVE ARE ERRORS SET IN OPERATION:
                                        ;* DRIVE SELECT
                                        ;* (ADDITIONAL LINES OF INFORMATION)
                                        ;*
                                        ;* THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.
                                        ;*
                                        ;* EXAMPLE:
                                        ;* NON-EXISTANT DRIVE
                                        ;* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
                                        ;* DRIVE SELECT
                                        ;* (ADDITIONAL LINES OF INFORMATION)
                                        ;*
                                        ;* THIS IS THE RESULT OF AN EXPECTED ERROR THAT DID NOT OCCUR, I.E.
                                        ;* A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.
                                        ;*
```

```
2435                                    ;* EXAMPLE:
2436                                    ;* NON-EXISTANT MEMORY
2437                                    ;* THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
2438                                    ;* UNIBUS PARITY ERROR
2439                                    ;* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
2440                                    ;* WRITE DATA
2441                                    ;* (ADDITIONAL LINES OF INFORMATION)
2442                                    ;*
2443                                    ;* THIS IS AN EXAMPLE OF NON-EXISTANT MEMORY BEING SET WHEN UNIBUS
2444                                    ;* PARITY ERROR WAS EXPECTED.
2445                                    ;*
2446                                    ;*      ERROR ITEM 4
2447                                    ;*      (DESCRIPTION OF ERROR)
2448                                    ;*      ERROR IN OPERATION
2449                                    ;*      (DESCRIPTION OF OPERATION)
2450                                    ;*      TST NUM ERR PC  DRIVE
2451                                    ;*      $TESTN  $ERRPC  DRVNUM
2452                                    ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF   RKDCYL   RKDA
2453                                    ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF   T.DCYL   T.DA
2454                                    ;*      RKBA    RKWC
2455                                    ;*      T.BA    T.WA
2456                                    ;*        A00     B00     A01     B01     A02     B02     A03     B03
2457                                    ;*        $REG10  $REG11  $REG12  $REG13  $REG14  $REG15  $REG16  $REG17
2458                                    ;*
2459                                    ;*      THE ERRORS REPORTED BY THIS FORMAT ARE:
2460                                    ;*      CONTROLLER DETECTED DRIVE BUS ERROR
2461                                    ;*      DRIVE DETECTED DRIVE BUS ERROR
2462                                    ;*      SEEK INCOMPLETE
2463                                    ;*      NON-EXECUTABLE DRIVE FUNCTION
2464                                    ;*      DRIVE TIMING ERROR
2465                                    ;*      DRIVE UNSAFE
2466                                    ;*      AC LOW
2467                                    ;*      SPINDLE SPEED LOSS
2468                                    ;*      DRIVE OFF TRACK
2469                                    ;*      ILLEGAL DRIVE ADDRESS ERROR
2470                                    ;*      CYLINDER OVERFLOW
2471                                    ;*      DRIVE TYPE ERROR
2472                                    ;*      FORMAT ERROR
2473                                    ;*      WRITE LOCK ERROR
2474
2475  001410  000000                    EM4N:   .WORD   0
2476  001412  000000                    DH4N:   .WORD   0
2477  001414  060514                            DT004
2478  001416  060634                            DF004
2479
2480                                    ;*      ERROR ITEM 5
2481                                    ;*      THIS ENTRY IS THE SAME AS ITEM 4 WITH THE ADDITION
2482                                    ;*      OF A MESSAGE THAT FOLLOWS.  THIS MESSAGE IS:
2483                                    ;*
2484                                    ;*      "ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"
2485                                    ;*
2486                                    ;*      THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR
2487                                    ;*      A00 THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.
2488
2489  001420  000000                    EM5N:   .WORD   0
2490  001422  000000                    DH5N:   .WORD   0
```

```
                                        DTOOS
                                        DFOOS

                                ;*      ERROR ITEM 6
                                ;*      (DESCRIPTION OF ERROR)
                                ;*      ERROR IN OPERATION
                                ;*      (DESCRIPTION OF OPERATION)
                                ;*      TST NUM ERR PC  DRIVE
                                ;*      $TESTN  $ERRPC  DRVNUM
                                ;*        RKCS1  RKCS2  RKDS  RKER  RKASOF  RKDCYL  RKDA
                                ;*        T.CS1  T.CS2  T.DS  T.ER  T.ASOF  T.DCYL  T.DA
                                ;*      RKBA  RKWC
                                ;*      T.BA  T.WC
                                ;*
                                ;*      THE ERRORS REPORTED BY THIS FORMAT ARE:
                                ;*      DATA CHECK
                                ;*      WRITE CHECK
                                ;*      ECC HARD
                                ;*      DATA LATE
                                ;*      OPERATION INCOMPLETE
                                ;*      HEADER VRC ERROR
                                ;*      BAD SECTOR ERROR

001430  000000                  EM6N:  .WORD   0
001432  000000                  DH6N:  .WORD   0
001434  060514                         DTOO6
001436  060720                         DFOO6

                                ;*      ERROR ITEM 7
                                ;*      (DESCRIPTION OF ERR
                                ;*      ERROR IN OPERATION
                                ;*      (DESCRIPTION OF OPERATION)
                                ;*      TST NUM ERR PC  DRIVE
                                ;*      $TESTN  $ERRPC  DRVNUM
                                ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF
                                ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF
                                ;*
                                ;*      THE ERRORS REPORTED BY THIS FORMAT ARE:
                                ;*      NON-EXISTANT DRIVE
                                ;*      NON-EXISTANT MEMORY
                                ;*      CONTROLLER TIME OUT
                                ;*      UNIT FIELD ERROR
                                ;*      MULTIPLE DRIVE SELECT
                                ;*      PROGRAMMING ERROR
                                ;*      UNIBUS PARITY ERROR
                                ;*      ILLEGAL FUNCTION CODE
                                ;*
                                ;*      DESCRIPTON OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR ILLEGAL

001440  000000                  EM7N:  .WORD   0
001442  000000                  DH7N:  .WORD   0
001444  060514                         DTOO7
001446  060744                         DFOO7

                                ;*      ERROR ITEM 10
                                ;*      (DESCRIPTION OF ERROR)
```

```
                                ;*      ERROR AT COMPLETION OF OPERATION
                                ;*      (DESCRIPITON OF OPERATION)
                                ;*      TST NUM ERR PC  DRIVE
                                ;*      $TESTN  $ERRPC  DRVNUM
                                ;*        EXPT    IS
                                ;*        $REG10  $REG11
                                ;*
                                ;*      THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY
                                ;*      COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS.  THE SPECIFIC
                                ;*      ERRORS ARE;
                                ;*      WORD COUNT INCORRECT
                                ;*      BUS ADDRESS INCORRECT
                                ;*      CYLINDER ADDRESS INCORRECT
                                ;*      TRACK ADDRESS INCORRECT
                                ;*      SECTOR ADDRESS INCORRECT

001450  000000                  EM10N:  .WORD   0
001452  057775                          DH010
001454  060564                          DT015
001456  060764                          DF010

                                ;*      ERROR ITEM 11
                                ;*      (ERROR INDICATOR OR STATUS BIT)
                                ;*      NOT SET AS A RESULT OF
                                ;*      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION
                                ;*      TST NUM ERR PC  DRIVE
                                ;*      $TESTN  $ERRPC  DRVNUM
                                ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF   RKMR1
                                ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF   T.MR1

001460  000000                  EM11N:  .WORD   0
001462  060121                          DH011
001464  060514                          DT010
001466  061004                          DF011

                                ;*      ERROR ITEM 12
                                ;*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
                                ;*      "NOT RESET AS A RESULT OF"

001470  000000                  EM12N:  .WORD   0
001472  060150                          DH012
001474  060514                          DT010
001476  061004                          DF011

                                ;*      ERROR ITEM 13
                                ;*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
                                ;*      "SET AS A RESULT OF"

001500  000000                  EM13N:  .WORD   0
001502  060201                          DH013
001504  060514                          DT010
001506  061004                          DF011

                                ;*      ERROR ITEM 14
                                ;*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
                                ;*      "RESET AS A RESULT OF"
```

```
2603  001510  000000                  EM14N:  .WORD   0
2604  001512  060224                          DH014
2605  001514  060514                          DT010
2606  001516  061004                          DF011
2607
2608                                  ;*      ERROR ITEM 15
2609                                  ;*      (HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)
2610                                  ;*      TST NUM ERR PC  DRIVE
2611                                  ;*      $TESTN  $ERRPC  DRVNUM
2612                                  ;*        GOOD    BAD     WORD NUM
2613                                  ;*        $REG10  $REG11  $REG12
2614
2615  001520  000000                  EM15N:  .WORD 0
2616  001522  057'62                          DH002A
2617  001524  060564                          DT015
2618  001526  061026                          DF015
2619
2620                                  ;*      ERROR ITEM 16
2621                                  ;*      ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15
2622
2623  001530  000000                          0
2624  001532  000000                          0
2625  001534  060572                          DT015A
2626  001536  061036                          DF016
2627
```

```
                                .SBTTL   REGISTER STORAGE FOR TEST

001540  000000          T.CS1:  .WORD   0
001542  000000          T.WC:   .WORD   0
001544  000000          T.BA:   .WORD   0
001546  000000          T.DA:   .WORD   0
001550  000000          T.CS2:  .WORD   0
001552  000000          T.DS:   .WORD   0
001554  000000          T.ER:   .WORD   0
001556  000000          T.ASOF: .WORD   0
001560  000000          T.DCYL: .WORD   0
001562  000000          T.SPAR: .WORD   0
001564  000000          T.DB:   .WORD   0
001566  000000          T.MR1:  .WORD   0
001570  000000          T.ECPS: .WORD   0
001572  000000          T.ECPT: .WORD   0
001574  000000          T.MR2:  .WORD   0
001576  000000          T.MR3:  .WORD   0

                                .SBTTL  REGISTER SETUP STORAGE
001600  000100          L.CS1:  .WORD   100             ;PRESET WITH INTERRUPT ENABLE
001602  000000          L.WC:   .WORD   0
001604  000000          L.BA:   .WORD   0
001606                  L.DA:
001606      000         L.DS:   .BYTE   0
001607      000         L.DT:   .BYTE   0
001610  000000          L.CS2:  .WORD   0
001612  000000          L.ASOF: .WORD   0
001614  000000          L.DCYL: .WORD   0
001616  000000          L.MR1:  .WORD   0
                                .SBTTL  PROGRAM DEFINED VARIABLES

001620  000000          RKVEC:  .WORD   0               ;RK VECTOR
001622  000000          RKPRI:  .WORD   0               ;RK PRIORITY
001624  000000          SRTFLG: .WORD   0               ;START FLAG
                                                        ;   0 = 200
                                                        ;   1 = 214
                                                        ;  -1 = 204
001626  000000          DRVNUM: .WORD   0               ;DRIVE UNDER TEST
001630  000000          DRVBIT: .WORD   0               ;WORD TO STORE BIT TO INDICATE DRIVE UNDER TEST
001632  000024          ERRCNT: .WORD   ^D20            ;ERROR COUNTER TO LIMIT ERROR
                                                        ;ERRORS REPORTED IN PROGRAM
001634  000024          ERRLMT: .WORD   ^D20            ;DATA COMPARE ERROR LIMIT
001636  061166          BSF24P: .WORD   BS24            ;POINTER TO BAD SECTORS 24 SECTOR MODE
                                                        ;   (FACTORY)
001640  061042          BSF26P: .WORD   BS26            ;POINTER TO BAD SECTORS 26 SECTOR MODE
                                                        ;   (FACTORY)
001642  000000          BSS24P: .WORD   0               ;POINTER TO BAD SECTORS 24 SECT MODE
                                                        ;  (SOFTWARE)
001644  000000          BSS26P: .WORD   0               ;POINTER TO BAD SECTORS 26 SECTOR MODE
                                                        ;   (SOFTWARE)
001646  000000          BS26CT: .WORD   0               ;COUNT OF BAD SECTORS 26 SECTOR MODE
001650  000000          BS24CT: .WORD   0               ;COUNT OF BAD SECTORS 24 SECTOR MODE
001652  000764          MILCNT: .WORD   ^D500           ;COUNT TO APPROXIMATE 1 MILL SEC
001654  000017          TIMCNT: .WORD   ^D15            ;COUNTER FOR MYTIME ROUTINE
        001660                  .=1660
```

```
2684  001660  000000                        .WORD   0
2685  001662  000000  INTSET:               .WORD   0               ;NON-ZERO IF RK06 INTERRUPT SINCE
2686                                                                ;LAST CLEARED
2687  001664  000000  OPTFLG:               .WORD   0               ;OPTION FLAGS
2688
2689          000001                  DOTST=  BIT0                  ;DRIVE 0 TO BE TESTED FLAG
2690          000002                  MEMSZB= BIT1                  ;MEMORY SIZE REPORT FLAG
2691          000004                  MEMPYB= BIT2                  ;MEMORY PARITY REPORT FLAG
2692          000010                  SRTINS= BIT3                  ;START UP INSTRUCTIONS REPORTED FLAG
2693          000100                  PARPRE= BIT6                  ;PARITY PRESENT FLAG
2694          000200                  BSERPT= BIT7                  ;BSE HAS BEEN REPORTED
2695          000400                  FPFMT=  BIT8                  ;FIRST PASS FORMAT SWITCH
2696          002000                  CP1170= BIT10                 ;CP IS 11/70 FLAG
2697          004000                  DRVRPT= BIT11                 ;DRIVE NUMBERS REPORTED FLAG
2698          100000                  LCLKPR= BIT15                 ;LINE CLOCK PRESENT
2699
2700  001666  000000  DRVDRP:               .WORD   0               ;LIST OF DRIVES DROPPED
2701  001670  000000  MEMPAR:               .WORD   0               ;WORD OF PARITY FLAGS
2702  001672  000000  CSRPTR:               .WORD   0               ;POINTER TO CSR TO USE FOR SETTING BAD
2703                                                                ;   PARITY
2704  001674  000000  LCLKTK:               .WORD   0               ;LINE CLOCK TICK COUNTER
2705  001676  000000  REFMT:                .WORD   0               ;REFORMAT SWITCH FOR HALT
2706
2707                  ;               THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR
2708                  ;               HEADER OR DATA COMPARE CONTINUATION PROCESS.
2709  001700  000000  DESHLD:               .WORD   0               ;DESTINATION HOLD
2710  001702  000000  SRCHLD:               .WORD   0               ;SOURCE HOLD
2711  001704  000000  WRDNUM:               .WORD   0               ;WORD NUMBER IN ERROR HOLD
2712  001706  000000  WRDCNT:               .WORD   0               ;WORDS LEFT IN COMPARE HOLD
2713  001710  177546  KWLADD:               .WORD   177546          ;KW11-L ADDRESS
2714  001712  000100  KWLVEC:               .WORD   100             ;KW11-L VECTOR
2715  001714  172100  MEMBAS:               .WORD   172100          ;MM11 ADDRESS
2716  001716  000114  MMVECA:               .WORD   114             ;MM11 VECRTOR
```

```
2717
2718
2719
2720
2721                                        .SBTTL  PROGRAM SETUP
2722
2723  001720  012737  000002  001624  SETCLK: MOV     #2,SRTFLG       ;SET START FLAG FOR CLOCK ADJUST
2724  001726  000412                          BR      START1
2725
2726  001730  012737  000001  001624  PARM:   MOV     #1,SRTFLG       ;SET START FLAG FOR PARMETER START
2727  001736  000406                          BR      START1
2728
2729  001740  012737  177777  001624  RESTRT: MOV     #-1,SRTFLG      ;LOAD START FLAG FOR PARMETER START
2730  001746  000402                          BR      START1
2731
2732  001750  005037  001624          START:  CLR     SRTFLG          ;CLEAR START FLAG
2733  001754  000005                  START1: RESET                   ;RESET THE WHOLE SYSTEM
2734  001756  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
2735  001762  012746  000340                  MOV     #PR7,-(SP)      ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
2736  001766  012746  001774                  MOV     #1$,-(SP)       ;LOAD START OF PROGRAM
2737  001772  000002                          RTI                     ;LOAD PSW
2738
2739  001774  004737  044626          1$:     JSR     PC,$TKINT       ;INITIALIZE KEYBOARD
2740  002000  005037  001676                  CLR     REFMT           ;CLEAR REFORMAT SWITCH
2741                                        .SBTTL  INITIALIZE THE COMMON TAGS
2742                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2743  002004  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2744  002010  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2745  002012  022706  001140                  CMP     #SWR,R6 ;;DONE?
2746  002016  001374                          BNE     .-6             ;;LOOP BACK IF NO
2747  002020  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2748                                        ;;INITIALIZE A FEW VECTORS
2749  002024  012737  032506  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2750  002032  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2751  002040  012737  033526  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2752  002046  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2753  002054  012737  046466  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2754  002062  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
2755  002070  012737  046310  000024          MOV     #$PWRON,@#PWRVEC ;;POWER FAILURE VECTOR
2756  002076  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2757  002104  013737  032024  032016          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2758  002112  005037  001262                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2759  002116  005037  001264                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2760  002122  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2761  002130  012737  002130  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2762  002136  012737  002136  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2763                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2764                                        ;;EQUAL TO A "-1"  SETUP FOR A SOFTWARE SWITCH REGISTER.
2765  002144  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2766  002150  012737  002204  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2767  002156  012737  177570  001140          MOV     #OSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2768  002164  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2769  002172  022777  177777  176740          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2770  002200  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2771                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
2772  002202  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
```

```
2773  002204  012716  002212          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2774  002210  000002                          RTI
2775  002212  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2776  002220  012737  000174  001142          MOV     #DISPREG,DISPLAY
2777  002226  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2778
2779  002232  005037  001304                  CLR     $PASS           ;;CLEAR PASS COUNT
2780  002236  132737  000200  001317          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2781  002244  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
2782  002246  012737  001320  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2783  002254                          67$:
2784                                  .SBTTL  TYPE PROGRAM NAME
2785                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2786  002254  005227  177777                  INC     #-1             ;;FIRST TIME?
2787  002260  001066                          BNE     68$             ;;BRANCH IF NO
2788  002262  022737  032160  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
2789  002270  001462                          BEQ     68$             ;;BRANCH IF YES
2790  002272  104401  002340                  TYPE    ,69$            ;;TYPE ASCIZ STRING
2791                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2792  002276  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2793  002302  001012                          BNE     70$             ;;BRANCH IF YES
2794  002304  123727  001316  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
2795  002312  001406                          BEQ     70$             ;;BRANCH IF YES
2796  002314  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2797  002322  001005                          BNE     71$             ;;BRANCH IF NO
2798  002324  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2799  002326  000403                          BR      71$
2800  002330  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2801  002336                          71$:
2802  002336  000437                          BR      68$             ;;GET OVER THE ASCIZ
2803                                  ;;69$:  .ASCIZ  <CRLF>*RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC  MAINDEC-11-DZR6K-D*<CRLF>
2804  002436                          68$:
2805  002436  132737  000200  001317          BITB    #BIT7,$ENVM     ;TEST IF DO NOT SIZE
2806  002444  001043                          BNE     3$              ;YES - SKIP
2807  002446  004737  032200                  JSR     PC,$SIZE
2808  002452  023727  032504  000740          CMP     $LSTBK,#740     ;MAKE SURE MEMORY IS SUFFICIENT
2809  002460  103007                          BHIS    2$              ;YES - SKIP
2810  002462  104401  050555                  TYPE    ,OPR005         ;MESSAGE (NOT ENOUGH MEMORY)
2811  002466  012737  000001  032016          MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
2812  002474  000137  031770                  JMP     $EOP
2813  002500  013700  032504          2$:     MOV     $LSTBK,R0       ;GET LAST BANK
2814  002504  012701  000006                  MOV     #6,R1           ;SET SHIFT COUNT
2815  002510  013703  032502                  MOV     $LSTAD,R3       ;GET LAST ADDRESS
2816  002514  005004                          CLR     R4              ;CLEAR R4 FOR OVERFLOW
2817  002516  005737  032236                  TST     $KT11           ;MEM MANAGE PRESENT?
2818  002522  100005                          BPL     23$             ;NO - SKIP
2819  002524  006100                  22$:    ROL     R0              ;SHIFT BANK LEFT
2820  002526  006104                          ROL     R4              ;ADD IN CARRY
2821  002530  005301                          DEC     R1              ;DECREMENT COUNT
2822  002532  001374                          BNE     22$             ;LOOP IF NOT ZERO
2823  002534  050003                          BIS     R0,R3           ;SET BANK BITS IN LAST ADDRESS
2824  002536  112737  000001  001327  23$:    MOVB    #1,$MTYP1       ;FORCE MEMORY TYPE TO 1
2825  002544  110437  001326                  MOVB    R4,$MAMS1       ;STORE UPPER MEMORY ADDRESS
2826  002550  010337  001330                  MOV     R3,$MADR1       ;STORE LOWER ADDRESS
2827  002554  032737  000010  001664  3$:     BIT     #SRTINS,OPTFLG  ;TEST IF ALREDY REPORTED
2828  002562  001005                          BNE     24$             ;YES - SKIP
```

```
2829  002564  104401  051664                  TYPE    ,OPR016         ;TYPE STARTUP INSTRUCTIONS
2830  002570  052737  000010  001664          BIS     #SRTINS,OPTFLG  ;SET REPORTED FLAG
2831  002576                          24$:
2832  002576  022737  000001  001624          CMP     #1,SRTFLG       ;CHECK IF PARAMETER START
2833  002604  001122                          BNE     15$             ;NO, START TESTING
2834  002606  104401  050446          5$:     TYPE    ,OPR001         ;TYPE "RK611 BUS ADDRESS (  ) ="
2835  002612  013746  001352                  MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
2836  002616  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2837  002620  104401  050475                  TYPE    ,OPR002
2838  002624  104412                          RDOCT                   ;GET VALUE
2839  002626  012637  001222                  MOV     (SP)+,$TMPO
2840  002632  001407                          BEQ     7$              ;CHECK IF <CR>
2841  002634  022737  160000  001222          CMP     #160000,$TMPO   ;CHECK IF IN I/O PAGE
2842  002642  101361                          BHI     5$
2843  002644  013737  001222  001352          MOV     $TMPO,$BASE     ;LOAD NEW BUS ADDRESS
2844  002652  104401  050503          7$:     TYPE    ,OPR003         ;TYPE "RK611 VECTOR ADDRESS (  ) ="
2845  002656  013746  001346                  MOV     $VECT1,-(SP)    ;GET $VECT1 FOR TYPOUT
2846  002662  042716  160000                  BIC     #160000,(SP)    ;CLEAR PRIORITY BITS
2847  002666  104402                          TYPOC
2848  002670  104401  050475                  TYPE    ,OPR002
2849  002674  104412                          RDOCT                   ;GET VALUE
2850  002676  012637  001222                  MOV     (SP)+,$TMPO
2851  002702  001412                          BEQ     10$             ;CHECK IF <CR>
2852  002704  022737  001000  001222          CMP     #1000,$TMPO
2853  002712  101757                          BLOS    7$              ;CHECK IF LEGAL
2854  002714  042737  017777  001346          BIC     #17777,$VECT1   ;CLEAR OLD VECTOR
2855  002722  053737  001222  001346          BIS     $TMPO,$VECT1    ;LOAD NEW VECTOR ADDRESS
2856  002730  104401  050533          10$:    TYPE    ,OPR004         ;TYPE "RK611 PRIORITY (  ) ="
2857  002734  005046                          CLR     -(SP)
2858  002736  113716  001347                  MOVB    $VECT1+1,(SP)
2859  002742  006216                          ASR     (SP)            ;SHIFT 5 BITS RIGHT
2860  002744  006216                          ASR     (SP)
2861  002746  006216                          ASR     (SP)
2862  002750  006216                          ASR     (SP)
2863  002752  006216                          ASR     (SP)
2864  002754  104402                          TYPOC
2865  002756  104401  050475                  TYPE    ,OPR002
2866  002762  104412                          RDOCT                   ;GET VALUE
2867  002764  012637  001222                  MOV     (SP)+,$TMPO
2868  002770  001430                          BEQ     15$             ;CHECK IF <CR>
2869  002772  022737  000007  001222          CMP     #7,$TMPO        ;CHECK IF LEGAL
2870  003000  103753                          BLO     10$
2871  003002  022737  000004  001222          CMP     #4,$TMPO
2872  003010  101347                          BHI     10$
2873  003012  006337  001222                  ASL     $TMPO           ;SHIFT 5 BITS LEFT
2874  003016  006337  001222                  ASL     $TMPO
2875  003022  006337  001222                  ASL     $TMPO
2876  003026  006337  001222                  ASL     $TMPO
2877  003032  006337  001222                  ASL     $TMPO
2878  003036  042737  160000  001347          BIC     #160000,$VECT1+1 ;CLEAR OLD PRIORITY
2879  003044  053737  001222  001347          BIS     $TMPO,$VECT1+1  ;LOAD RK611 PRIORITY
2880  003052  005037  001304          15$:    CLR     $PASS           ;SET PASS COUNT TO ZERO
2881  003056  005037  001666                  CLR     DRVDRP          ;CLEAR DROPPED DRIVES LIST
2882  003062  042737  004000  001664          BIC     #DRVRPT,OPTFLG  ;CLEAR DRIVE #'S REPORTED FLAG
2883  003070  004737  034434                  JSR     PC,OPTTST       ;SETUP PARITY CHECK & CLOCK
2884  003074  013700  001346                  MOV     $VECT1,R0       ;STORE VECTOR FOR USE
```

```
 2885  003100  042700  160000            BIC     #160000,R0      ;CLEAR PRIORITY BITS
 2886  003104  010037  001620            MOV     R0,RKVEC
 2887  003110  012710  034354            MOV     #INTHLR,(R0)    ;SETUP INTERRUPT ADDRESS
 2888  003114  113737  001347  001622     MOVB    $VECT1+1,RKPRI  ;STORE PRIORITY FOR USE
 2889  003122  013702  001352            MOV     $BASE,R2        ;SET BASE ADDRESS
 2890  003126  005037  001264            CLR     $ESCAPE         ;CLEAR ESCAPE
 2891  003132  012746  000000            MOV     #PRO,-(SP)      ;SET PRIORITY
 2892  003136  012746  003144            MOV     #16$,-(SP)
 2893  003142  000002                    RTI
 2894  003144                     16$:
 2895
```

```
                                .SBTTL  **BASIC INTERFACE AND OPTION TESTS
                                ;;*****************************************************************
                                ;*TEST 1         RK611 BASE ADDRESS TEST
                                ;*      CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES NOT
                                ;*      CAUSE A NON-EXISTANT MEMORY TRAP. IF A TRAP OCCURS
                                ;*      THE PROGRAM IS HALTED.
                                ;;*****************************************************************
003144  000004                  TST1:   SCOPE
003146  012737  000100  001262          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003154  012706  001100          MOV     #STACK,SP       ;CLEAN OFF STACK
003160  012701  000004          MOV     #4,R1           ;SET POINTER TO VECTOR
003164  012146                  MOV     (R1)+,-(SP)     ;STORE OLD VECTOR CONTENTS
003166  011146                  MOV     (R1),-(SP)
003170  012701  000004          MOV     #4,R1           ;RESET POINTER
003174  012721  034346          MOV     #NEXINT,(R1)+   ;SET VECTOR TO MEM TEST HANDLER
003200  012711  000340          MOV     #PR7,(R1)       ;SET PRIORITY
003204  013702  001352          MOV     $BASE,R2        ;SET POINTER TO RK611 BASE ADDRESS
003210  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT COUNTER
003214  012762  000000  000000  MOV     #0,RKCS1(R2)    ;WRITE CS1 TO SEE IN MEM WILL SET
003222  000240                  NOP
003224  000240                  NOP
003226  000240                  NOP
003230  005737  001662          TST     INTSET          ;TEST IF COUNTER IS 0
003234  001406                  BEQ     1$              ;YES - SKIP ERROR REPORT
003236  012737  053303  001360  MOV     #EM1,EMIN       ;MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
003244  104001                  ERROR   1
003246  000137  043404          JMP     CTRHLT          ;GO TO CONTROLLED HALT
003252  012701  000006  1$:     MOV     #6,R1           ;RESTORE VECTOR
003256  012611                  MOV     (SP)+,(R1)
003260  012641                  MOV     (SP)+,-(R1)

                                ;;*****************************************************************
                                ;*TEST 2         INTERRUPT VECTOR ADDRESS TEST
                                ;*      CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE
                                ;*      EXPECTED ADDRESS. IF INTERRUPT VECTOR IS IN ERROR,
                                ;*      THE PROGRAM IS HALTED.
                                ;;*****************************************************************
003262  000004                  TST2:   SCOPE
003264  012737  000100  001262          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003272  012762  005000  000010          MOV     #CLR,RKCS2(R2) ;CLEAR SUBSYSTEM, SPECIFICALLY TO
                                                                ;CLEAR ANY OLD INTERRUPTS
003300  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT COUNTER
003304  012762  000300  000000  MOV     #RDY!IE,RKCS1(R2)       ;WRITE CS1 TO FORCE INTERRUPT
003312  000240                  NOP
003314  000240                  NOP
003316  000240                  NOP
003320  005737  001662          TST     INTSET          ;TEST IF INTERRUPT OCCURRED
003324  001011                  BNE     3$              ;YES - SKIP ERROR REPORT
003326  105737  001103          TSTB    $ERFLG          ;TEST IF ERFLG ALREADY SET. IF SO THE
                                                        ;INTERRUPT WENT TO THE WRONG VECTOR
                                                        ;AND MESSAGE HAS BEEN REPORTED.
003332  001004                  BNE     2$              ;THEREFORE - EXIT
003334  012737  053303  001360  MOV     #EM1,EMIN       ;MESSAGE (NO INTERRUPT)
003342  104001                  ERROR   1
```

```
  2952  003344  000137  043404          2$:     JMP     CTRHLT          ;GO TO CONTROLLED HALT
  2953  003350                          3$:
  2954
  2955                                  .SBTTL  **STATUS VALID TESTS
  2956
  2957
  2958                                  ;*****************************************************************
  2959                                  ;*TEST 3        SELECT ALL DRIVES
  2960                                  ;*
  2961                                  ;*      IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT,
  2962                                  ;*      DETERMINE WHAT DRIVES ARE ON-LINE BY
  2963                                  ;*      SELECTING ALL DRIVES.  IF NON-EXISTENT DRIVE REPORTED
  2964                                  ;*      MAKE SURE STATUS VALID IS RESET.  IF DRIVE
  2965                                  ;*      PRESENT MAKE SURE NO ERROR EXISTS, DRIVE
  2966                                  ;*      IS CYCLED UP, AND STATUS VALID SET, AND DSC RESET.
  2967                                  ;*
  2968                                  ;*      IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES
  2969                                  ;*      IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE
  2970                                  ;*      CYCLED UP, AND STATUS VALID SET.
  2971                                  ;*
  2972                                  ;*      IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON
  2973                                  ;*      THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM
  2974                                  ;*      START (214) WAS USED. IF THE AUTOMATIC START (200)
  2975                                  ;*      IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204)
  2976                                  ;*      WILL RETAIN THE TEST STATUS OF DRIVE 0.
  2977                                  ;*
  2978                                  ;*      IF THE PARAM START IS USED, THE OPERATOR MUST
  2979                                  ;*      EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED
  2980                                  ;*      OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO
  2981                                  ;*      BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME
  2982                                  ;*      VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.
  2983                                  ;*
  2984                                  ;*      ALL DRIVES TO BE TESTED MUST BE ON-LINE, CYCLED UP, AND
  2985                                  ;*      WRITE LOCK RESET. ADDRESSES OF DRIVES THAT ARE NON-
  2986                                  ;*      EXISTANT EITHER BECAUSE THE DRIVE DOES NOT EXIST OR IS OFF-
  2987                                  ;*      LINE ARE USED TO VERIFY NON-EXISTANT DRIVE ERROR DETECTION.
  2988                                  ;*      DRIVES THAT ARE ON-LINE BUT NOT CYCLED UP OR ARE WRITE
  2989                                  ;*      LOCKED ARE NOT TESTED.
  2990                                  ;*
  2991                                  ;*      AT COMPLETION OF THE TEST A MESSAGE WILL BE GIVEN TO
  2992                                  ;*       IDENTIFY THE DRIVES TO BE USED IN TESTING.
  2993                                  ;*
  2994                                  ;*      NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE
  2995                                  ;*              ANY OTHER TEST THAT FOLLOWS.
  2996                                  ;*****************************************************************
  2997  003350  000004                 TST3:   SCOPE
  2998  003352  012737  000062  001262         MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
  2999  003360  104416                         TSSINIT                 ;CALL SUBSYSTEM CLEAR AND TEST
  3000  003362  104003                         ERROR   3
  3001
  3002  003364  012746  000000                 MOV     #PRO,-(SP)      ;SET PROCESSOR PRIORITY TO ALLOW
  3003  003370  012746  003376                 MOV     #1$,-(SP)       ;RK611 INTERRUPTS
  3004  003374  000002                         RTI
  3005
  3006  003376  013701  001620         1$:     MOV     RKVEC,R1        ;GET VECTOR
  3007  003402  012721  034354                 MOV     #INTHLR,(R1)+ ;LOAD INTERRUPT VECTOR
```

```
3008  003406  012711  000340                  MOV     #PR7,(R1)
3009  003412  012703  001354                  MOV     #$DEVM,R3       ;GET ADDRESS OF DEVICE MAP
3010  003416  132737  000200  001317          BITB    #BIT7,$ENVM     ;TEST IF SHOULD SIZE
3011  003424  001007                          BNE     50$             ;NO - SKIP DRIVE SIZING.
3012  003426  005737  001304                  TST     $PASS           ;TEST IF FIRST PASS
3013  003432  001004                          BNE     50$             ;NO - SKIP TO DRIVE SELECT TEST
3014  003434  032737  004000  001664          BIT     #DRVRPT,OPTFLG  ;TEST IF DRIVE #'S REPORTED
3015  003442  001402                          BEQ     92$             ;NO - GOTO SIZING
3016  003444  000137  004032          50$:    JMP     11$
3017  003450  005013          92$:    CLR     (R3)            ;CLEAR DEVICE MAP
3018  003452  123727  000041  000013          CMPB    @#41,#13        ;TEST IF RK06 IS LOAD DEVICE
3019  003460  001077                          BNE     77$             ;NO - SKIP
3020  003462  022737  000001  001624          CMP     #1,SRTFLG       ;WAS START AT PARAM?
3021  003470  001411                          BEQ     2$              ;YES - SKIP
3022  003472  104401  050776                  TYPE    ,OPR007         ;NO TEST OF DRIVE 0
3023  003476  052737  000001  001666          BIS     #BIT0,DRVDRP    ;SET DRIVE 0 DROPPED
3024  003504  042737  000001  001664          BIC     #DOTST,OPTFLG   ;DR FLAG - NO TEST DRIVE 0
3025  003512  000465                          BR      7$
3026  003514  104401  050626          2$:     TYPE    ,OPR006         ;MESSAGE - SWAP PACK ON DRIVE OFF LINE.
3027  003520  005037  001610                  CLR     L.CS2           ;SET TO DRIVE 0
3028  003524  005037  001232                  CLR     $TMP4           ;CLEAR FOR USE AS A SWITCH
3029  003530  012737  000101  001600  3$:     MOV     #SELDRV,L.CS1   ;LOAD FOR SELECT
3030  003536  012737  003600  001264          MOV     #4$,$ESCAPE     ;SET UP ESCAPE IN CASE OF ERR
3031  003544  104417                          TLOADRK                 ;LOAD RK & DO SELECT
3032
3033  003546  104423                          TWAT16                  ;WAIT 16MS FOR COMPLETION
3034  003550  104002                          ERROR   2               ;NOT DONE ON TIME
3035
3036  003552  104420                          TGETRK                  ;GET RK REGISTER
3037  003554  032737  100000  001540          BIT     #CERR,T.CS1     ;TEST IF CERR
3038  003562  001417                          BEQ     5$              ;NO - SKIP
3039  003564  032737  010000  001550          BIT     #NED,T.CS2      ;TEST IF NED
3040  003572  001002                          BNE     4$              ;YES - SKIP
3041
3042  003574  104421                          TCHKOP                  ;CHECK THE OPERATION AND REPORT THE ERROR
3043  003576  104004                          ERROR   4 ;OR5,6,7      ;AFTER THE ERROR IS REPORTED THE TEST
3044                                                                  ;IS ABORTED
3045  003600  104401  050776          4$:     TYPE    ,OPR007         ;TYPE NO TEST OF DRIVE 0
3046  003604  042737  000001  001664          BIC     #DOTST,OPTFLG   ;DR FLAG - NO TEST OF DRIVE 0
3047  003612  052737  000001  001666          BIS     #BIT0,DRVDRP    ;SET DRV 0 AS DROPPED
3048  003620  000422                          BR      7$              ;SKIP OVER WAIT FOR PACK MOUNT
3049  003622  005737  001232          5$:     TST     $TMP4           ;TEST FLAG DRIVE READY HAS RESET
3050  003626  001010                          BNE     6$              ;YES - SKIP TO CHECK IF IT IS SET AGAIN
3051  003630  032737  000200  001552          BIT     #DRDY,T.DS      ;ELSE CHECK READY
3052  003636  001334                          BNE     3$              ;STILL SET - GET STATUS AGAIN
3053  003640  012737  177777  001232          MOV     #-1,$TMP4       ;ELSE SET FLAG TO INDICATE READY WENT LOW
3054  003646  000730                          BR      3$              ;GO GET STATUS AGAIN
3055
3056  003650  032737  000200  001552  6$:     BIT     #DRDY,T.DS      ;TEST IF READY SET AGAIN
3057  003656  001724                          BEQ     3$              ;NO - GO GET STATUS AGAIN
3058  003660  052737  000001  001664  77$:    BIS     #DOTST,OPTFLG   ;ELSE SET DRV 0 TEST FLAG.
3059
3060  003666  013737  003770  001264  7$:     MOV     35$,$ESCAPE     ;SET UP ESCAPE IN CASE OF ERR
3061  003674  005000                          CLR     R0              ;CLEAR FOR DRIVE NUMBER COUNTER
3062  003676  012701  000001                  MOV     #1,R1           ;SET BIT 0 AS DRIVE SELECTOR
3063
```

```
3064  003702  032737  000001  001664          BIT     #DOTST,OPTFLG   ;TEST DRIVE 0?
3065  003710  001433                          BEQ     9$              ;NO - SKIP
3066
3067  003712  104416                  8$:     TSSINIT                 ;INITIALZE SUBSYSTEM
3068  003714  104003                          ERROR   3               ;ERROR IF NOT SUCCESSFUL
3069
3070  003716  010037  001610                  MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
3071  003722  012737  000.31  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3072  003730  104417                          TLOADRK                 ;LOAD RK REGS
3073
3074  003732  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3075  003734  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3076
3077  003736  104420                          TGETRK                  ;GET RK REGS
3078  003740  032737  100000  001540          BIT     #CERR,T.CS1     ;ERROR?
3079  003746  001022                          BNE     10$             ;YES - SKIP
3080  003750  032737  000200  001552          BIT     #DRDY,T.DS      ;ELSE TEST IF DRIVE READY
3081  003756  001404                          BEQ     35$             ;NO - SKIP
3082  003760  032737  004000  001552          BIT     #WRL,T.DS       ;ELSE TEST IF WRITE LOCKED
3083  003766  001403                          BEQ     36$             ;NO - SKIP
3084  003770  050137  001666          35$:    BIS     R1,DRVDRP       ;SET THIS BIT AS DROPPED DRIVE
3085  003774  000401                          BR      9$
3086  003776                          36$:
3087  003776  050113                          BIS     R1,(R3)         ;SET BIT - DRIVE PRESET IN MAP
3088  004000  005200                  9$:     INC     R0              ;BUMPS TO NEXT DRIVE
3089  004002  006301                          ASL     R1              ;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
3090  004004  032701  000400                  BIT     #BIT8,R1        ;WAS LAST DRIVE DONE?
3091  004010  001740                          BEQ     8$              ;YES - SKIP
3092  004012  000407                          BR      11$             ;ELSE LOOP TO SELECT NEXT DRIVE
3093
3094  004014  032737  010000  001550  10$:    BIT     #NED,T.CS2      ;WAS CERR DUE TO NED?
3095  004022  001366                          BNE     9$              ;YES - BUMP TO NEXT DRIVE
3096
3097  004024  104421                          TCHKOP                  ;ELSE REPORT THE ERRORS
3098  004026  104004                          ERROR   4 ;OR5,6,7
3099  004030  000000                  101$:   .WORD   0
3100  004032  032737  004000  001664  11$:    BIT     #DRVRPT,OPTFLG  ;TEST IF SHOULD REPORT
3101  004040  001037                          BNE     16$             ;NO - SKIP
3102
3103  004042  005713                          TST     (R3)            ;ANY DRIVE AVAILABLE?
3104  004044  001004                          BNE     12$             ;BR IF NOT ZERO
3105  004046  104401  051141                  TYPE    ,OPR008         ;ELSE REPORT NO DRIVES AVAILABLE
3106  004052  000137  043404                  JMP     CTRHLT          ;GO TO CONTROLLED HALT
3107
3108  004056  012701  000200          12$:    MOV     #BIT7,R1        ;SET DRIVE SELECTOR FOR DRIVE 7
3109  004062  012700  000007                  MOV     #7,R0           ;SET DRIVE NUMBER TO 7
3110  004066  104401  051224                  TYPE    ,OPR009         ;TYPE PREFIX TO DRIVE TEST MESSAGE
3111
3112  004072  030113          13$:    BIT     R1,(R3)         ;TEST IF THIS DRIVE TO BE TESTED
3113  004074  001007                          BNE     15$             ;YES - SKIP
3114
3115  004076  005300          14$:    DEC     R0              ;ELSE DECREMENT DRIVE NUMBER
3116  004100  006201                          ASR     R1              ;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
3117  004102  001373                          BNE     13$             ;IF NOT SHIFTED OUT - LOOP
3118  004104  052737  004000  001664          BIS     #DRVRPT,OPTFLG  ;SET DRIVE #'S REPORTED FLAG
3119
```

```
3120  004112  000412                          BR      16$             ;ELSE GO TO STATUS VALID TEST
3121
3122  004114  010037  004030                  15$:    MOV     R0,101$         ;PUT DRIVE NUMBER IN TYPE LOCATION
3123  004120  052737  000060  004030                  BIS     #BIT4!BIT5,101$ ;MAKE IT ASCIZ
3124  004126  104401                                  TYPE                    ;TYPE IT
3125  004130  004030                                  101$
3126  004132  104401  050443                          TYPE    ,SPACE2         ;TYPE SOME SPACES
3127  004136  000757                                  BR      14$             ;LOOP
3128
3129  004140  005000                          16$:    CLR     R0              ;CLEAR DRIVE NUMBER COUNTER
3130  004142  012701  000001                          MOV     #1,R1           ;SET DRIVE SELECTOR TO DRIVE 0
3131  004146  012737  177777  001240                  MOV     #-1,$TMP7       ;SET $TMP7 NEGATIVE
3132  004154  012737  177777  001630                  MOV     #-1,DRVBIT      ;SET DRIVE SELECT BIT NEGATIVE
3133  004162  012737  004264  001264                  MOV     #18$,$ESCAPE    ;SET ESCAPE IN CASE OF ERR
3134  004170  030137  001666                          BIT     R1,DRVDRP       ;HAS DRIVE 0 BEEN DROPPED?
3135  004174  001033                                  BNE     18$             ;YES - SKIP TO DRIVE INC.
3136
3137  004176  104416                          17$:    TSSINIT                 ;CLEAR SUBSYSTEM
3138  004200  104003                                  ERROR   3               ;ERROR FOR BAD CLEAR
3139
3140  004202  010037  001610                          MOV     R0,L.CS2        ;SET DRIVE SELECT
3141  004206  010037  001626                          MOV     R0,DRVNUM       ;SET DRIVE NUMBER
3142  004212  012737  000101  001600                  MOV     #SELDRV,L.CS1   ;SET FOR DRIVE SELECT
3143
3144  004220  104417                                  TLOADRK                 ;LOAD RK REGS
3145  004222  104423                                  THAT16                  ;WAIT FOR INTERRUPT
3146  004224  104002                                  ERROR   2               ;ERROR TO SLOW/NOT COMPLETE
3147  004226  030113                                  BIT     R1,(R3)         ;WAS THAT DRIVE AVAILABLE
3148  004230  001026                                  BNE     19$             ;YES - SKIP
3149
3150  004232  104422                                  TCHKWE                  ;CHECK THAT ERROR OCCURRED
3151  004234  000000                                  .WORD   0               ;NONE OF GROUP 1
3152  004236  000000                                  .WORD   0               ;NONE OF GROUP 2
3153  004240  000001                                  .WORD   1               ;NED IN GROUP 3
3154  004242  104004                                  ERROR   4 ;OR5,6,7      ;ERROR IF NO ERROR OR WRONG ERROR
3155
3156  004244  032737  100000  001552                  BIT     #SVAL,T.DS      ;DID STATUS VALID RESET?
3157  004252  001404                                  BEQ     18$             ;YES - SKIP
3158  004254  012737  055425  001400                  MOV     #EM47,EM3N      ;ELSE MESSAGE (SVAL NOT RESET WITH NED)
3159  004262  104003                                  ERROR   3
3160  004264  005200                          18$:    INC     R0              ;BUMP TO NEXT DRIVE
3161  004266  006301                                  ASL     R1              ;SHIFT DRIVE SELECT BIT
3162  004270  032701  000400                          BIT     #BIT8,R1        ;ALL DRIVES CHECKED
3163  004274  001063                                  BNE     21$             ;YES - EXIT
3164  004276  030137  001666                          BIT     R1,DRVDRP       ;TEST IF DRIVE HAS BEEN DROPPED
3165  004302  001370                                  BNE     18$             ;YES - GET NEXT DRIVE
3166  004304  000734                                  BR      17$             ;ELSE CHECK THIS DRIVE
3167  004306  104421                          19$:    TCHKOP                  ;CHECK NO ERRORS SET
3168  004310  104004                                  ERROR   4 ;OR5,6,7      ;REPORT ALL ERRORS
3169
3170  004312  032737  100000  001552                  BIT     #SVAL,T.DS      ;CHECK SVAL SET
3171  004320  001004                                  BNE     20$             ;YES - SKIP
3172  004322  012737  055506  001400                  MOV     #EM48,EM3N      ;MESSAGE (NO SVAL FROM EXISTANT DR)
3173  004330  104003                                  ERROR   3
3174
3175  004332  012737  000103  001600  20$:    MOV     #PACK,L.CS1     ;ELSE SET TO DO PACK ACK
```

```
3176  004340  104417                          TLOADRK                 ;LOAD RK
3177
3178  004342  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
3179  004344  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3180
3181  004346  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3182  004350  104004                          ERROR   4 ;OR5,6,7      ;YES - REPORT & ABORT TEST
3183
3184  004352  032737  000100  001552          BIT     #VV,T.DS        ;DID VV SET
3185  004360  001005                          BNE     22$             ;YES - SKIP
3186  004362  012737  055370  001400          MOV     #EM46,EM3N      ;MESSAGE (VV DID NOT SET/
3187  004370  104003                          ERROR   3
3188  004372  000734                          BR      18$
3189
3190  004374  032737  040000  001552  22$:    BIT     #DSC,T.DS       ;TEST IF DSC RESET
3191  004402  001410                          BEQ     23$             ;YES - SKIP
3192  004404  012737  056502  001470          MOV     #EMDSC,EM12N
3193  004412  012737  056742  061010          MOV     #EMSCLR,DF011A
3194  004420  104003                          ERROR   3               ;"DSC NOT RESET RESULT OF SUBSYS CLEAR"
3195  004422  000720                          BR      18$
3196
3197  004424  005737  001630  23$:    TST     DRVBIT          ;TEST IF DRVBIT IS NEGATIVE
3198  004430  100315                          BPL     18$             ;NO - SKIP
3199  004432  010137  001630                  MOV     R1,DRVBIT       ;STORE DRIVE SELECT BIT
3200  004436  010037  001240                  MOV     R0,$TMP7        ;STORE DRIVE NUMBER TO BE TESTED
3201  004442  000710                          BR      19$
3202
3203  004444  013737  001240  001626  21$:    MOV     $TMP7,DRVNUM    ;LOAD LOWEST # DRIVE PRESENT INTO DRVNUM
3204
3205  004452  023727  001624  000002          CMP     SRTFLG,#2       ;TEST IF CLOCK ADJUST START
3206  004460  001002                          BNE     25$             ;NO - SKIP
3207  004462  000137  043274                  JMP     ADJCLK          ;GO TO ADJUST CLOCK ROUTINE
3208
3209  004466                          25$:
3210
3211                                  ;;*****************************************************************
3212                                  ;*TEST 4        RELEASE ALL DRIVES
3213                                  ;*
3214                                  ;*      RELEASE ALL DRIVES.  MAKE SURE NO ERROR
3215                                  ;*      SETS AND STATUS VALID IS RESET.
3216                                  ;*
3217                                  ;*****************************************************************
3218  004466  000004                  TST4:   SCOPE
3219  004470  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3220  004476  104416                          TSSINIT                 ;INITIALIZE SUBSYSTEM
3221  004500  104003                          ERROR   3               ;BAD INIT
3222
3223  004502  013737  001626  001610          MOV     DRVNUM,L.CS2    ;SET DRIVE NUMBER
3224  004510  012737  000101  001600          MOV     #SELDRV,L.CS1   ;SET DRIVE SELECT
3225
3226  004516  104417                          TLOADRK                 ;LOAD RK REGS
3227  004520  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
3228  004522  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3229
3230  004524  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3231  004526  104004                          ERROR   4 ;OR5,6,7      ;REPORT ANY ERRORS
```

```
                        T4      RELEASE ALL DRIVES

3232
3233  004530  012737  000010  001610        MOV     #RLS,L.CS2      ;SET DRIVE RELEASE,STILL SET FOR SELECT
3234
3235  004536  104417                        TLOADRK                 ;LOAD RK REGS
3236  004540  104423                        TWAIT16                 ;WAIT FOR INTERRUPT
3237  004542  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3238
3239  004544  104421                        TCHKOP                  ;CHECK FOR ANY ERRORS
3240  004546  104004                        ERROR   4  ;OR 5, 6, OR 7 ;REPORT ALL ERRORS
3241  004550  032737  100000  001552        BIT     #SVAL,T.DS      ;DID SVAL RESET?
3242  004556  001404                        BEQ     1$              ;YES - SKIP
3243  004560  012737  055565  001400        MOV     #EM49,EM3N      ;MESSAGE (SVAL NOT RESET W/RELEASE)
3244  004566  104003                        ERROR   3
3245  004570                          1$:
3246
3247  004570                          TSTLUP:
3248                                  ;;*****************************************************************
3249                                  ;*TEST 5        NON-STANDARD MESSAGES AND SVAL
3250                                  ;*
3251                                  ;*      PICK ONE OF THE AVAILABLE DRIVES AND GET
3252                                  ;*      NON-STANDARD MESSAGES.  MAKE SURE NO
3253                                  ;*      ERROR OCCURS AND STATUS VALID DOES NOT SET
3254                                  ;*      AND THAT NON-STANDARD MESSAGES CAUSE STATUS
3255                                  ;*      VALID TO RESET.
3256                                  ;*
3257                                  ;;*****************************************************************
3258  004570  000004                  TST5:   SCOPE
3259  004572  012737  000062  001262        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3260  004600  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
3261  004602  104003                        ERROR   3               ;BAD CLEAR MESSAGE
3262  004604  012701  000001                MOV     #1,R1           ;PRESET R1 FOR MESSAGE PAIR 1
3263  004610  013737  001626  001610        MOV     DRVNUM,L.CS2    ;LOAD DRV NUMBER
3264  004616  012737  000101  001600        MOV     #SELDRV,L.CS1   ;LOAD SELECT COMMAND
3265  004624  005037  001616          1$:   CLR     L.MR1           ;LOAD FOR STANDARD STATUS
3266  004630  104417                        TLOADRK                 ;LOAD RK
3267  004632  104423                        TWAIT16                 ;WAIT FOR INTERRUPT
3268  004634  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3269  004636  104421                        TCHKOP                  ;CHECK OPERATION
3270  004640  104004                        ERROR   4 ;5,6 OR 7     ;REPORT ALL ERRORS
3271
3272  004642  032737  100000  001552        BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
3273  004650  001007                        BNE     2$              ;YES-SKIP
3274
3275  004652  012737  056342  001460        MOV     #EMSVAL,EM11N
3276  004660  012737  047640  061010        MOV     #EMSELD,DF011A
3277  004666  104011                        ERROR   11              ;"SVAL NOT SET RESULT OF DRIVE SELECT"
3278
3279  004670  010137  001616          2$:   MOV     R1,L.MR1        ;LOAD MESSAGE PAIR SELECT
3280
3281  004674  104417                        TLOADRK                 ;LOAD RK
3282  004676  104423                        TWAIT16                 ;WAIT FOR INTERRUPT
3283  004700  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3284
3285  004702  104421                        TCHKOP                  ;CHECk OPERATION
3286  004704  104004                        ERROR   4 ;5,6, OR 7    ;REPORT ALL ERRORS
3287
```

```
3288  004706  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID RESET
3289  004714  001407                          BEQ     3$              ;YES-SKIP
3290
3291  004716  012737  056342  001470          MOV     #EMSVAL,EM12N
3292  004724  012737  056357  061010          MOV     #EMNZPR,DF011A
3293  004732  104012                          ERROR   12              ;"SVAL NOT RESET RESULT OF SEL W/ NON-0 PAIR"
3294
3295  004734  022701  000003          3$:     CMP     #3,R1           ;WAS PAIR 3 SELECTED?
3296  004740  001402                          BEQ     4$              ;YES-SKIP
3297  004742  005201                          INC     R1              ;BUMP TO NEXT PAIR
3298  004744  000727                          BR      1$              ;SKIP TO DO IT.
3299  004746                          4$:
3300                                  ;;*****************************************************************
3301                                  ;*TEST 6        WRITING CS2 AND STATUS VALID
3302                                  ;*
3303                                  ;*      SELECT AN AVAILABLE DRIVE.  MAKE SURE STATUS
3304                                  ;*      VALID IS SET.  WRITE COMMAND AND STATUS REGISTER 2.
3305                                  ;*      MAKE SURE STATUS VALID RESETS.
3306                                  ;*
3307                                  ;;*****************************************************************
3308  004746  000004          TST6:   SCOPE
3309  004750  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3310  004756  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3311  004760  104003                          ERROR   3               ;BAD INIT ERROR
3312
3313  004762  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3314  004770  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3315
3316  004776  104417                          TLOADRK                 ;LOAD RK
3317  005000  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
3318  005002  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3319
3320  005004  104421                          TCHKOP                  ;CHECK OPERATION
3321  005006  104004                          ERROR   4 ;5,6, OR 7    ;REPORT ALL ERRORS
3322
3323  005010  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
3324  005016  001007                          BNE     1$              ;YES-SKIP
3325
3326  005020  012737  056342  001460          MOV     #EMSVAL,EM11N
3327  005026  012737  047640  061010          MOV     #EMSELD,DF011A
3328  005034  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV SELECT"
3329
3330  005036  013762  001626  000010  1$:     MOV     DRVNUM,RKCS2(R2)  ;WRITE CS2 TO RESET SVAL
3331
3332  005044  104420                          TGETRK                  ;GET RK REGS.
3333
3334  005046  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAL RESET
3335  005054  001407                          BEQ     2$              ;YES-SKIP
3336
3337  005056  012737  056342  001470          MOV     #EMSVAL,EM12N
3338  005064  012737  056420  061010          MOV     #EMWCS2,DF011A
3339  005072  104012                          ERROR   12              ;"SVAL NOT RESET BY WRITING CS2"
3340  005074                          2$:
3341
```

```
                                .SBTTL  **CONTROLLER ERROR TESTS

                                ;;*********************************************************************
                                ;*TEST 7         DRIVE TYPE ERROR
                                ;*
                                ;*      CREATE A DRIVE TYPE ERROR.  MAKE SURE DRIVE
                                ;*      TYPE ERROR SETS AND STATUS VALID SETS.
                                ;*
                                ;;*********************************************************************
005074  000004                  TST7:   SCOPE
005076  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
005104  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
005106  104003                          ERROR   3               ;BAD INIT ERROR

005110  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
005116  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
005124  052737  002000  001600          BIS     #CDT,L.CS1      ;LOAD DRIVE TYPE

005132  104417                          TLOADRK                 ;LOAD RK
005134  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
005136  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

005140  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
005142  000040                          .WORD   000040          ;DRIVE TYPE ERROR
005144  000000                          .WORD   0
005146  000000                          .WORD   0
005150  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DIFFERENCES (NO ERRORS,
                                                                ;ADDITIONAL ERRORS, DIFFERENT ERRORS)
005152  032737  100000  001552          BIT     #SVAL, T.DS     ;TEST IF SVAL SET
005160  001007                          BNE     1$              ;YES-SKIP

005162  012737  056342  001460          MOV     #EMSVAL,EMIIN
005170  012737  055036  061010          MOV     #EMOTPE,DFOIIA
005176  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV TYPE ERR"
005200                          1$:
                                ;;*********************************************************************
                                ;*TEST 10        STATUS VALID AND PARITY ERROR
                                ;*
                                ;*      ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY.
                                ;*      MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION,
                                ;*      DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT,
                                ;*      AND STATUS VALID SET, ISSUE A CONTROLLER
                                ;*      CLEAR.  MAKE SURE DRIVE INTERRUPT AND ATTENTION
                                ;*      ARE STILL SET.  SELECT DRIVE AGAIN WITH GOOD
                                ;*      PARITY.  MAKE SURE ATTENTION, DRIVE STATUS
                                ;*      CHANGE, DRPAR, CONTROLLLER ERROR, DRIVE INTERRUPT,
                                ;*      AND STATUS VALID ARE SET AND SPAR IS RESET.
                                ;*      ISSUE A CONTROLLER CLEAR.  GET NON-STANDARD MESSAGES
                                ;*      AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION
                                ;*      ARE SET.  CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT
                                ;*      FOR ALL AVAILIABLE DRIVES.
                                ;*
                                ;;*********************************************************************
005200  000004                  TST10:  SCOPE
005202  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
005210  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
```

```
3398  005212  104003                          ERROR   3               ;BAD INIT ERROR
3399
3400  005214  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3401  005222  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3402  005230  012737  000020  001616          MOV     #PAT,L.MR1      ;LOAD EVEN PARITY BIT
3403
3404  005236  104417                          TLOADRK                 ;LOAD RK REGS-SELECT W/EVEN PARITY
3405  005240  104423                          THAT16                  ;WAIT FOR INTERRUPT
3406  005242  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3407
3408  005244  104422                          TCHKWE                  ;CHECK OPERATION FOR EXPECTED ERROR
3409  005246  000011                          DRPARERR!SPARERR        ;DRIVE SELECTED DRIVE BUS PARITY ERROR
3410  005250  000000                          .WORD   0               ;CONTROLLER DETECTED DRIVE BUS PARITY ERROR
3411  005252  000000                          .WORD   0
3412  005254  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DIFFERENCES
3413
3414  005256  012700  000400                  MOV     #BIT8,R0        ;ROUTINE TO DETERMINE WHICH BIT
3415
3416  005262  013701  001626                  MOV     DRVNUM,R1       ;SHOULD BE SET IN ASOF TO INDICATE
3417  005266  001403                          BEQ     3$              ;DRIVE ATTENTION. R0 WILL HAVE THE
3418  005270  006300                  2$:     ASL     R0              ;BIT THAT SHOULD BE SET FOR THE DRIVE
3419  005272  005301                          DEC     R1              ;IN USE
3420  005274  001375                          BNE     2$
3421
3422  005276  030037  001556          3$:     BIT     R0,T.ASOF       ;TEST ATTENTION SET
3423  005302  001007                          BNE     4$              ;YES-SKIP
3424  005304  012737  056526  001460          MOV     #EMDA,EM11N
3425  005312  012737  054753  061010          MOV     #EMDPAR,DF011A
3426  005320  104011                          ERROR   11              ;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3427  005322  032737  040000  001540  4$:     BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT SET
3428  005330  001007                          BNE     5$              ;YES-SKIP
3429  005332  012737  056462  001460          MOV     #EMDI,EM11N
3430  005340  012737  054753  061010          MOV     #EMDPAR,DF011A
3431  005346  104011                          ERROR   11              ;"DRV INT NOT SET RESULT OF DRV PARITY ERR"
3432
3433  005350  032737  040000  001552  5$:     BIT     #DSC,T.DS       ;TEST DRIVE STATUS CHANGE SET
3434  005356  001007                          BNE     6$              ;YES-SKIP
3435  005360  012737  056502  001460          MOV     #EMDSC,EM11N
3436  005366  012737  054753  061010          MOV     #EMDPAR,DF011A
3437  005374  104011                          ERROR   11              ;"DSC NOT SET RESULT OF DRV PARITY ERR"
3438
3439  005376  032737  100000  001552  6$:     BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
3440  005404  001007                          BNE     7$              ;YES-SKIP
3441  005406  012737  056342  001460          MOV     #EMSVAL,EM11N
3442  005414  012737  054753  061010          MOV     #EMDPAR,DF011A
3443
3444  005422  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV PAR ERR"
3445
3446  005424  005037  001616          7$:     CLR     L.MR1           ;CLEAR PAT IN MR1
3447
3448  005430  052737  100000  001600          BIS     #CCLR,L.CS1     ;CLEAR CONTROLLER
3449  005436  104417                          TLOADRK                 ;LOAD RK REGS TO DO CLEAR
3450
3451
3452  005440  104421                          TCHKOP                  ;CHECK NO ERRORS SET
3453  005442  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS STILL SET
```

```
3454
3455  005444  030037  001556          BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET
3456  005450  001007                  BNE     8$              ;YES-SKIP
3457  005452  012737  056526  001510  MOV     #EMDA,EM14N
3458  005460  012737  056546  061010  MOV     #EMCCLR,DF011A
3459  005466  104014                  ERROR   14              ;"DRV ATT RESET RESULT OF CONT CLR"
3460
3461  005470  032737  040000  001540  8$:     BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT STILL SET
3462  005476  001007                  BNE     9$              ;YES-SKIP
3463  005500  012737  056462  001510  MOV     #EMDI,EM14N
3464  005506  012737  056546  061010  MOV     #EMCCLR,DF011A
3465  005514  104014                  ERROR   14              ;"DRV INT RESET RESULT OF CONTROLLER CLR"
3466
3467  005516  012737  000101  001600  9$:     MOV     #SELDRV,L.CS1   ;LOAD SELECT DRIVE
3468
3469  005524  104417                  TLOADRK                 ;LOAD RK REGS
3470  005526  104423                  TWAT16                  ;WAIT FOR INTERRUPT
3471  005530  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3472
3473  005532  104422                  TCHKWE                  ;CHECK THAT ERRORS ARE SET
3474  005534  000010                  .WORD   000010          ;DPAR SET (NO SPAR 1, REPORT
3475  005536  000000                  .WORD   0               ;ANY DISCREPENCIES
3476  005540  000000                  .WORD   0
3477  005542  104004                  ERROR   4 ; OR 5,6,7
3478  005544  030037  001556          BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET.
3479  005550  001007                  BNE     10$             ;YES-SKIP
3480  005552  012737  056526  001460  MOV     #EMDA,EM11N
3481  005560  012737  054753  061010  MOV     #EMDPAR,DF011A
3482  005566  104011                  ERROR   11              ;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3483
3484  005570  032737  040000  001540  10$:    BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT STILL SET
3485  005576  001007                  BNE     11$             ;YES-SKIP
3486  005600  012737  056462  001460  MOV     #EMDI,EM11N
3487  005606  012737  054753  061010  MOV     #EMDPAR,DF011A
3488  005614  104011                  ERROR   11              ;"DRV INT NOT SET RESULT OF DRV  PARITY ERR"
3489
3490  005616  032737  040000  001552  11$:    BIT     #DSC,T.DS       ;TEST STATUS CHANGE STILL SET
3491  005624  001007                  BNE     12$             ;YES-SKIP
3492  005626  012737  056502  001460  MOV     #EMDSC,EM11N
3493  005634  012737  054753  061010  MOV     #EMDPAR,DF011A
3494  005642  104011                  ERROR   11              ;"DSC NOT SET RESULT OF DRV PARITY ERR"
3495
3496  005644  032737  100000  001552  12$:    BIT     #SVAL,T.DS      ;TEST STATUS VALID STILL SET
3497  005652  001007                  BNE     13$             ;YES-SKIP
3498  005654  012737  056342  001460  MOV     #EMSVAL,EM11N
3499  005662  012737  054753  061010  MOV     #EMDPAR,DF011A
3500  005670  104011                  ERROR   11              ;"SVAL NOT SET RESULT OF DRV PARITY ERR"
3501
3502  005672  052737  100000  001600  13$:    BIS     #CCLR,L.CS1     ;CLEAR CONTROLLER
3503
3504  005700  104417                  TLOADRK                 ;LOAD RK REGS (DO CLEAR)
3505
3506
3507  005702  012701  000001          MOV     #1,R1           ;SET COUNTER TO SELECT STATUS PAIR
3508  005706  012737  000101  001600  MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3509
```

```
3510  005714  010137  001616          14$:    MOV     R1,L.MR1        ;LOAD STATUS PAIR SELECTION
3511  005720  104417                          TLOADRK                 ;LOAD RK REGS
3512  005722  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3513  005724  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3514
3515  005726  104421                          TCHKOP                  ;CHECK IF ANY ERRORS SET
3516  005730  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS SET.
3517
3518  005732  030037  001556                  BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET
3519  005736  001007                          BNE     15$             ;YES-SKIP
3520  005740  012737  056526  001410          MOV     #EMDA,EM4N
3521  005746  012737  056357  061010          MOV     #EMNZPR,DF011A
3522  005754  104014                          ERROR   14              ;"ATTENTION RESET RESULT OF NON-0 PAIR SEL"
3523
3524  005756  032737  040000  001540  15$:    BIT     #DI,T.CS1
3525  005764  001007                          BNE     16$
3526  005766  012737  056462  001510          MOV     #EMDI,EM14N
3527  005774  012737  056357  061010          MOV     #EMNZPR,DF011A
3528  006002  104014                          ERROR   14              ;"DRV INT RESET RESULT OF NON-0 PAIR SELECT"
3529
3530  006004  005201                  16$:    INC     R1              ;BUMP PAIR SELECT
3531  006006  022701  000004                  CMP     #4,R1           ;ALL PAIRS DONE?
3532  006012  001340                          BNE     14$             ;NO-LOOP
3533
3534  006014  005037  001616                  CLR     L.MR1           ;CLEAR MR1
3535
3536  006020  012737  000105  001600          MOV     #CLEAR,L.CS1    ;LOAD DRIVE CLEAR
3537
3538  006026  104417                          TLOADRK                 ;DO DRIVE CLEAR
3539  006030  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3540  006032  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3541
3542  006034  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3543  006036  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
3544
3545  006040  012701  000020                  MOV     #20,R1          ;SET COUNT FOR SHORT WAIT
3546  006044  005301                  17$:    DEC     R1              ;TO ALLOW CONTROLLER TIME TO POLL
3547  006046  001376                          BNE     17$             ;DRIVES
3548
3549  006050  104420                          TGETRK                  ;GET RK REGS
3550  006052  030037  001556                  BIT     R0,T.ASOF       ;TEST ATTENTION RESET
3551  006056  001407                          BEQ     18$             ;YES-SKIP
3552  006060  012737  056526  001470          MOV     #EMDA,EM12N
3553  006066  012737  047666  061010          MOV     #EMDCLR,DF011A
3554  006074  104012                          ERROR   12              ;"ATTENTION NOT RESET RESULT OF DRV CLEAR
3555
3556  006076  032737  040000  001540  18$:    BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT RESET
3557  006104  001407                          BEQ     19$             ;YES-SKIP
3558  006106  012737  056462  001470          MOV     #EMDI,EM12N
3559  006114  012737  047666  061010          MOV     #EMDCLR,DF011A
3560  006122  104012                          ERROR   12              ;"DRV INT NOT RESET RESULT OF DRIVE CLR"
3561
3562  006124                          19$:
3563                                  ;;*****************************************************************
3564                                  ;*TEST 11        UNIT FIELD ERROR ON RELEASE
3565                                  ;*
```

```
                                        ;*      ISSUE A SUBSYSTEM CLEAR, SELECT AN AVAILABLE
                                        ;*      DRIVE.  PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE
                                        ;*      A RELEASE COMMAND.  CLOCK THROUGH PHASE ADDRESS 2.
                                        ;*      TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD
                                        ;*      ERROR SETS.
                                        ;*
                                        ;;*****************************************************************
006124  000004                        TST11:  SCOPE
006126  012737  000062  001262                MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
006134  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
006136  104002                                ERROR   2               ;BAD INIT ERROR

006140  013737  001626  001610                MOV     DRVNUM,L.CS2    ;SELECT A DRIVE
006146  012737  000101  001600                MOV     #SELDRV,L.CS1   ;DO DRIVE SELECT

006154  104417                                TLOADRK                 ;LOAD RK
006156  104423                                TWAT16                  ;WAIT FOR INTERRUPT
006160  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

006162  104421                                TCHKOP                  ;CHECK FOR ANY ERRORS
006164  104004                                ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS.

006166  052737  000010  001610                BIS     #RLS,L.CS2      ;LOAD RELEASE
006174  012737  000040  001616                MOV     #DMD,L.MR1      ;SET DIAGNOSTIC MODE

006202  104417                                TLOADRK                 ;LOAD RK

006204  004437  036004                        JSR     R4,MCLOCK       ;CALL MAINT CLOCK
006210     023                                .BYTE   ^D19            ;NUMBER OF PHASES
006211     002                                .BYTE   2               ;NUMBER OF CLOCK XISTIONS

006212  042762  000040  000026                BIC     #DMD,RKMR1(R2)  ;CLEAR DIAG MODE

006220  104423                                TWAT16                  ;WAIT FOR INTERRUPT
006222  104002                                ERROR   2               ;TO SLOW/NOT COMPLETED

006224  104422                                TCHKWE                  ;CHECK OPERATION WITH ERROR
006226  000000                                .WORD   0
006230  000000                                .WORD   0
006232  000004                                .WORD   UFERR           ;UNIT FIELD ERROR
006234  104004                                ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES

006236  104416                                TSSINIT                 ;CLEAR SUBSYSTEM TO INSURE UFE RESETS
006240  104002                                ERROR   2
                                        ;;*****************************************************************
                                        ;*TEST 12        UNIT FIELD ERROR ON SELECT
                                        ;*
                                        ;*      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
                                        ;*      DRIVE.  PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE
                                        ;*      A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE
                                        ;*      SELECTED = 0.  CLOCK THROUGH PHASE ADDRESS 6.
                                        ;*      TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD
                                        ;*      ERROR SETS.
                                        ;*
                                        ;;*****************************************************************
006242  000004                        TST12:  SCOPE
```

```
3622  006244  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3623  006252  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3624  006254  104003                          ERROR   3               ;BAD INIT ERROR
3625
3626  006256  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3627  006264  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3628
3629  006272  104417                          TLOADRK                 ;LOAD RK
3630  006274  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
3631  006276  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3632
3633  006300  104421                          TCHKOP                  ;CHECK FOR ANY ERROR
3634  006302  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
3635
3636  006304  012737  000043  001616          MOV     #DMD!BIT1!BIT0,L.MR1    ;LOAD DIAG MODE & MSG PAIR 3
3637  006312  005037  001610                  CLR     L.CS2           ;LOAD FOR DRIVE 0
3638
3639  006316  104417                          TLOADRK                 ;LOAD RK
3640
3641  006320  004437  036004                  JSR     R4,MCLOCK       ;CALL MAINTENANCE CLOCK
3642  006324     026                          .BYTE   ^D22            ;THROUGH PHASE 6
3643  006325     002                          .BYTE   2               ;PLUS 2 TRANSITIONS
3644
3645  006326  042762  000040  000026          BIC     #DMD,RKMR1(R2)  ;CLEAR DIAG MODE
3646
3647  006334  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
3648  006336  104002                          ERROR   2               ;TO SLOW/NOT COMPLETED ERROR
3649
3650  006340  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
3651  006342  000000                          .WORD   0
3652  006344  000000                          .WORD   0
3653  006346  000004                          .WORD   UFERR           ;UNIT FIELD ERROR SHOULD SET
3654  006350  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
3655
3656                          .SBTTL  **ATTENTION HANDLING BY CONTROLLER
3657
3658                          ;;*****************************************************************
3659                          ;*TEST 13       DOUBLE INTERRUPT
3660                          ;*
3661                          ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.
3662                          ;*      MAKE SURE STATUS VALID IS SET.  CHECK THAT SECOND
3663                          ;*      INTERRUPT OCCURS.  AFTER SECOND INTERRUPT
3664                          ;*      CHECK THAT STATUS VALID IS RESET.  ISSUE SELECT
3665                          ;*      AND MAKE SURE STATUS VALID SETS.  CLEAR DRIVE.
3666                          ;*      CHECK THAT DRIVE STATUS CHANGE SETS
3667                          ;*      (BIT 14 OF DRIVE STATUS
3668                          ;*      REGISTER)
3669                          ;*
3670                          ;;*****************************************************************
3671  006352  000004          TST13:  SCOPE
3672  006354  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3673  006362  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3674  006364  104003                          ERROR   3               ;BAD INIT ERROR
3675
3676  006366  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3677  006374  012737  000113  001600          MOV     #RECAL,L.CS1    ;LOAD RECAL
```

```
3678
3679  006402  104417                          TLOADRK                 ;LOAD RK
3680  006404  104423                          TWAT16                  ;WAIT FOR 1ST INTERRUPT
3681  006406  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3682  006410  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
3683  006414  104420                          TGETRK                  ;GET RK REGS
3684  006416  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAL SET
3685  006424  001010                          BNE     1$              ;YES-SKIP
3686  006426  012737  056342  001460          MOV     #EMSVAL,EM11N
3687  006434  012737  047727  061010          MOV     #EMRCAL,DF011A
3688  006442  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF RECAL"
3689  006444  000463                          BR      50$             ;ABORT TEST
3690
3691  006446  104437                  1$:     TWATBS                  ;WAIT FOR INTERRUPT
3692  006450  000401                          BR      2$              ;NO INTERRUPT RETURN
3693  006452  000404                          BR      3$              ;INTERRUPT RETURN
3694
3695  006454  012737  055635  001370  2$:     MOV     #EM50,EM2N      ;ALTER MESSAGE "NO 2ND INTERRUPT OR IT WAS LATE"
3696  006462  104002                          ERROR   2
3697
3698  006464  104420                  3$:     TGETRK                  ;GET RK REGS
3699  006466  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAC SET NOW
3700  006474  001410                          BEQ     4$              ;NO-SKIP
3701  006476  012737  056342  001470          MOV     #EMSVAL,EM12N
3702  006504  012737  056567  061010          MOV     #EM2INT,DF011A
3703  006512  104012                          ERROR   12              ;"SVAL NOT RESET RESULT OF SECOND TEST"
3704  006514  000437                          BR      50$
3705
3706  006516  032737  040000  001552  4$:     BIT     #DSC,T.DS       ;TEST DSC SET BY ATTENTION
3707  006524  001010                          BNE     5$              ;YES-SKIP
3708  006526  012737  056502  001460          MOV     #EMDSC,EM11N
3709  006534  012737  056567  061010          MOV     #EM2INT,DF011A
3710  006542  104011                          ERROR   11              ;"DSC NOT SET RESULT OF SECOND INTERRUPT"
3711  006544  000423                          BR      50$
3712
3713  006546  012737  000101  001600  5$:     MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3714
3715  006554  104417                          TLOADRK                 ;LOAD RK REGS
3716  006556  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3717  006560  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3718
3719  006562  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3720
3721  006564  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
3722
3723  006566  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAC SET
3724  006574  001007                          BNE     50$             ;YES-SKIP
3725  006576  012737  056342  001460          MOV     #EMSVAL,EM11N
3726  006604  012737  047640  061010          MOV     #EMSELD,DF011A
3727  006612  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV SEL.
3728  006614                          50$:
3729                                  ;;*****************************************************************
3730                                  ;*TEST 14       SINGLE INTERRUPT FROM ATTENTION
3731                                  ;*
3732                                  ;*      DO A SEEK TO CYLINDER 0.  WAIT FOR INTERRUPT FROM
3733                                  ;*      DRIVE ATTENTION.  LOWER PRIORITY AGAIN AND MAKE
```

```
3734                                    ;*      SURE ANOTHER INTERRUPT DOES NOT OCCUR.  CLEAR DRIVE.
3735                                    ;*
3736                                    ;;*********************************************************************
3737  006614  000004                    TST14:  SCOPE
3738  006616  012737  000062  001262            MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3739  006624  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
3740  006626  104003                            ERROR   3               ;BAD INIT ERROR
3741
3742  006630  013737  001626  001610            MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3743  006636  012737  000117  001600            MOV     #SEEK,L.CS1     ;LOAD SEEK (CYL LEFT AT 0.
3744
3745  006644  104417                            TLOADRK                 ;LOAD RK REGS
3746  006646  104423                            TWAT16                  ;WAIT FOR INTERRUPT
3747  006650  104002                            ERROR   2               ;TO SLOW/NOT COMPLETED ERROR
3748
3749
3750  006652  104420                            TGETRK                  ;GET RK REGS
3751
3752  006654  032737  040000  001540            BIT     #DI,T.CS1       ;TEST DI SET
3753  006662  001010                            BNE     2$              ;YES-SKIP
3754  006664  012737  056462  001460            MOV     #EMDI,EM11N
3755  006672  012737  056610  061010            MOV     #EMSKSF,DF011A
3756  006700  104011                            ERROR   11              ;"DI NOT SET RESULT OF SEEK TO SELF"
3757  006702  000417                            BR      50$
3758
3759
3760  006704  012700  000031            2$:     MOV     #25.,R0         ;LOAD AND DECREMENT A COUNT TO
3761  006710  005300                    3$:     DEC     R0              ;ZERO. GIVE CONTROLLER A CHANCE TO
3762  006712  001376                            BNE     3$              ;INTERRUPT AGAIN.  ERROR IF IT DOES.
3763
3764  006714  022737  000001  001662            CMP     #1,INTSET       ;CHECK ONLY ONE INTERRUPT OCCURRED
3765  006722  001407                            BEQ     50$             ;YES-SKIP
3766  006724  012737  056762  001500            MOV     #EMMI,EM13N
3767  006732  012737  056610  061010            MOV     #EMSKSF,DF011A
3768  006740  104013                            ERROR   13              ;"MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"
3769
3770  006742  104421                    50$:    TCHKOP                  ;CHECK FOR ANY ERRORS
3771  006744  104004                            ERROR   4  ;OR 5,6,7    ;REPORT ALL ERRORS
3772                                    ;;*********************************************************************
3773                                    ;*TEST 15       RESET ATTENTIONS WITH UNIBUS INIT
3774                                    ;*
3775                                    ;*      DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
3776                                    ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
3777                                    ;*
3778                                    ;;*********************************************************************
3779  006746  000004                    TST15:  SCOPE
3780  006750  012737  000005  001262            MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
3781  006756  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
3782  006760  104003                            ERROR   3               ;BAD INIT ERROR
3783
3784  006762  013737  001626  001610            MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3785  006770  012737  000117  001600            MOV     #SEEK,L.CS1     ;LOAD SEEK (TO SELF-0)
3786
3787  006776  104417                            TLOADRK                 ;LOAD RK REGS
3788  007000  104423                            TWAT16                  ;WAIT FOR INTERRUPT
3789  007002  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE
```

```
                                
007004  104420                  TGETRK                  ;GET RK REGS
                                
007006  032737  040000  001540  BIT     #DI,T.CS1       ;TEST DI SET
007014  001010                  BNE     1$              ;YES-EXIT
007016  012737  056462  001460  MOV     #EMDI,EM11N
007024  012737  056610  061010  MOV     #EMSKSF,DF011A
007032  104011                  ERROR   11              ;"DI NOT SET RESULT OF SEEK TO SELF
007034  000450                  BR      50$
                                
007036  005037  001662  1$:     CLR     INTSET          ;CLEAR INTERRUPT COUNTER
007042  000005                  RESET                   ;DO UNIBUS RESET
007044  004737  044626          JSR     PC,$TKINT       ;RESET KEYBOARD INTERRUPT
                                
007050  005037  001674          CLR     LCLKTK          ;CLEAR TICK COUNTER
007054  004737  035150  5$:     JSR     PC,MYTIME       ;CALL TIMER
007060  022737  000012  001674  CMP     #10.,LCLKTK     ;COUNT 10 TICKS (MILLISECONDS)?
007066  001372                  BNE     5$              ;NO - LOOP
                                
007070  012762  000100  000000  MOV     #IE,RKCS1(R2)   ;SET IE FOR ANY STRAY INTERRUPTS
007076  004737  034434          JSR     PC,OPTTST       ;SET UP OPTIONS AGAIN
                                
007102  104423                  TWAT16                  ;WAIT 16 MS FOR AN INTERRUPT
007104  000410                  BR      2$              ;NONE IS EXPECTED SO RETURN SHOULD BE
                                                        ;HERE-BR TO CONTINUE TEST.
007106  012737  056625  001500  MOV     #EMUXIT,EM13N   ;INT OCCURRED ON RESET
007114  012737  056667  061010  MOV     #EMRSET,DF011A
007122  104013                  ERROR   13              ;"UNEXECUTED INTERRUPT RESULT OF RESET"
007124  000414                  BR      50$
007126  104420          2$:     TGETRK                  ;GET RK REGS
007130  032737  040000  001540  BIT     #DI,T.CS1       ;TEST DI RESET
007136  001407                  BEQ     50$             ;YES-SKIP
007140  012737  056462  001470  MOV     #EMDI,EM12N
007146  012737  056667  061010  MOV     #EMRSET,DF011A
007154  104012                  ERROR   12              ;"DI NOT RESET RESULT OF RESET"
                                
007156                  50$:
                                
                        .SBTTL  **ILLEGAL DISK ADDRESS ERROR TESTS
                                
                        ;;*****************************************************************
                        ;*TEST 16       ILLEGAL DISK ADDRESS (PART 1)
                        ;*
                        ;*      ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE
                        ;*      ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS.
                        ;*      CLEAR CONTROLLER AND CLEAR DRIVE.  REPEAT FOR HEADS 4-7,
                        ;*      CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
                        ;*      HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.
                        ;*
                        ;;*****************************************************************
007156  000004          TST16:  SCOPE
007160  012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
007166  012701  000003          MOV     #3,R1           ;PRESET FOR SELECTING TRACK 3
                                
007172  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
007174  104003                  ERROR   3
```

```
3846
3847  007176  012737  000113  001600          MOV     #RECAL,L.CS1    ;SET UP TO RECAL
3848  007204  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE
3849
3850  007212  104417                          TLOADRK                 ;LOAD RK REGS
3851
3852  007214  104423                          TWAT16                  ;WAIT FOR 1ST INTERRUPT
3853  007216  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3854
3855  007220  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
3856
3857  007224  104437                          TWAT8S                  ;WAIT FOR INTERRUPT
3858  007226  104002                          ERROR   2
3859
3860  007230  012737  007236  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP ON ERROR
3861
3862  007236  104416                  1$:     TSSINIT                 ;CLEAR SUBSYSTEM
3863  007240  104003                          ERROR   3               ;BAD INIT ERROR
3864
3865  007242  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3866  007250  012737  000117  001600          MOV     #SEEK,L.CS1     ;LOAD SEEK
3867  007256  110137  001607                  MOVB    R1,L.DT         ;LOAD TRACK
3868
3869  007262  104417                          TLOADRK                 ;LOAD RK REGS
3870  007264  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3871  007266  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3872
3873  007270  032701  000001                  BIT     #BIT0,R1        ;TEST IF HEAD ADDRESS HAS BIT 0
3874  007274  001403                          BEQ     2$              ;NO - SKIP
3875  007276  032701  000002                  BIT     #BIT1,R1        ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
3876  007302  001007                          BNE     3$              ;YES-GO CHECK BOTH IDAE AND SKI SET
3877
3878  007304  104422                  2$:     TCHKWE                  ;CHECK OPERATION WITH ERROR
3879  007306  002000                          IDAERR                  ;ILLEGAL DISK ADDRESS ERROR
3880  007310  000000                          0
3881  007312  000000                          0
3882  007314  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL DISCREPANCIES
3883  007316  104415                          SCOP1                   ;LOCAL LOOP ON ERROR
3884  007320  000406                          BR      4$
3885
3886  007322  104422                  3$:     TCHKWE                  ;CHECK OPERATION WITH ERROR
3887  007324  002002                          IDAERR!SKIERR           ;ILLEGAL DISK ADDRESS ERROR
3888  007326  000000                          0                       ;SEEK INCOMPLETE
3889  007330  000000                          0
3890  007332  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ANY DISCREPANCIES
3891  007334  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
3892
3893  007336  005201                  4$:     INC     R1              ;ELSE BUMP TO NEXT ILLEGAL TRACK
3894  007340  022701  000010                  CMP     #8.,R1          ;ALL ILLEGAL TRACKS SELECTED?
3895  007344  001334                          BNE     1$              ;NO-LOOP
3896                          ;;*****************************************************************
3897                          ;*TEST 17        ILLEGAL DISK ADDRESS (PART 2)
3898                          ;*
3899                          ;*      ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE
3900                          ;*      ILLEGAL DISK ADDRESS ERROR SETS.  CLEAR CONTROLLER AND DRIVE
3901                          ;*
```

```
                                ;;*****************************************************************
007346  000004                TST17:  SCOPE
007350  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
007356  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
007360  104003                        ERROR   3               ;BAD INIT ERROR

007362  012737  000113  001600        MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
007370  013737  001626  001610        MOV     DRVNUM,L.CS2    ;LOAD DRIVE

007376  104417                        TLOADRK                 ;LOAD RK REGS

007400  104423                        TWAT16                  ;WAIT FOR 1ST INTERRUPT
007402  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

007404  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG

007410  104437                        TWAT8S                  ;WAIT FOR INTERRUPT
007412  104002                        ERROR   2

007414  012737  007422  001110        MOV     #1$,$LPERR      ;SET LOOP TO BYPASS RECAL

007422  104416                1$:     TSSINIT                 ;CLEAR SUBSYSTEM
007424  104003                        ERROR   3

007426  013737  001626  001610        MOV     DRVNUM, L.CS2   ;LOAD DRIVE NUMBER
007434  012737  000117  001600        MOV     #SEEK,L.CS1     ;LOAD SEEK
007442  012737  001000  001614        MOV     #1000, L.DCYL   ;LOAD ILLEGAL CYLINDER

007450  104417                        TLOADRK                 ;LOAD RK REGS
007452  104423                        TWAT16                  ;WAIT FOR INTERRUPT
007454  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

007456  104422                        TCHKWE                  ;CHECK OPERATION WITH ERROR
007460  002000                        .WORD   IDAERR          ;DISK ADDRESS ERROR
007462  000000                        .WORD   0
007464  000000                        .WORD   0
007466  104004                        ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPANCIES

                        .SBTTL  **WRITE HEADER TESTS

                        ;;*****************************************************************
                        ;*TEST 20        READ BAD SECTOR INFORMATION
                        ;*
                        ;*       ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
                        ;*       TRACK 2 TO GET THE FACTORY DETECTED BAD
                        ;*       SECTOR FILE, 26 SECTOR MODE.
                        ;*
                        ;*       IF AN ERROR OCCURS, READ SECTOR 2, 4, 6, OR 10(8) UNTIL
                        ;*       A SUCCESSFUL READ IS DONE. IF NONE READ SUCCESSFULLY
                        ;*       REMOVE THIS DRIVE FROM TEST. WHEN A READ IS SUCCESSFUL,
                        ;*       TEST THAT THE PACK IS NOT AN ALIGNMENT PACK AND
                        ;*       STORE THE ENTRIES FOR LATER USE.
                        ;*
                        ;*       REPEAT THIS SERIES OF OPERATIONS FOR FACTORY DETECTED
                        ;*       BAD SECTORS 24 SECTOR MODE, SOFTWARE DETECTED
                        ;*       BAD SECTORS 26 SECTOR MODE, AND SOFTWARE DETECTED BAD
```

```
                                ;*      SECTORS 24 SECTOR MODE. IF THE NUMBER OF BAD SECTORS FOR
                                ;*      24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED
                                ;*      FROM TESTING.
                                ;*
                                ;*
                                ;*****************************************************************
007470  000004                TST20:  SCOPE
007472  012737  000001  001262          MOV     #1,$TIMES       ;;DO 1 ITERATION
007500  105037  007540                  CLRB    2$              ;CLEAR SECTOR POINTER
007504  005000                          CLR     R0
007506  005005                          CLR     R5              ;CLEAR R5 FOR BAD SECTOR COUNTING
007512  013703  001640                  MOV     BSF26P,R3       ;SET POINT IN TO STORE BS 26 SECT FORMAT
007520  012737  007522  001110          MOV     #1$,$LPERR      ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
007524  104416                1$:       TSSINIT                 ;CLEAR SUBSYSTEM
007526  104003                          ERROR   3               ;BAD INIT ERROR
007530
007532  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS WITH
007536  000121                          RDDATA                  ; READ DATA
007540  177400                          -400                    ; WORD COUNT
007542  061414                          IBUFF                   ; BUFFER ADDRESS
007544      000         2$:             .BYTE   0               ; SECTOR ADDRESS
007545      002                         .BYTE   2               ; TRACK ADDRESS
007546  000632                          632                     ; CYLINDER ADDRESS
007550
007550  104417                          TLOADRK                 ;LOAD "L" REGS INTO RK
007552  104431                          TWAIT12                 ;WAIT FOR INTERRUPT
007554  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
007556  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
007560  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
007562  104415                          SCOP1                   ;LOOP TO 1$ IF SW 9 SET
007564
007566  105737  001103                  TSTB    $ERFLG          ;TEST FOR ERROR IN OPERATION
007572  001502                          BEQ     7$              ;NO-SKIP
007574  005700                          TST     R0              ;GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
007576  001023                          BNE     3$              ;NO-SKIP
007600  062737  000002  007540          ADD     #2,2$           ;NEXT SECTOR ADDRESS
007606  122737  000012  007540          CMPB    #10.,2$         ;PAST APPLICABLE SECTORS?
007614  001066                          BNE     6$              ;NO-SKIP
007616  012737  055734  001360          MOV     #EM51,EMIN
007624  104001                          ERROR   1               ;"CANNOT READ BS FILES
007626  043737  001630  001354  25$:    BIC     DRVBIT,$DEVM    ;CLEAR DRIVE FROM DRIVE MAP
007634  053737  001630  001666          BIS     DRVBIT,DRVDRP   ;SET DRIVE DROPPED
007642  000137  027764                  JMP     NEWDRV          ;ABORT TEST PASS.
007646
007650  022700  000001          3$:     CMP     #1,R0           ;GETTING A BS SOFT SECTOR 26 SECT FORMAT?
007654  001014                          BNE     4$              ;NO-SKIP
007656  062737  000002  007540          ADD     #2,2$           ;NEXT SECTOR ADDRESS
007664  122737  000026  007540          CMPB    #22.,2$         ;PAST APPLICABLE SECTORS?
007672  001040                          BNE     6$              ;NO-SKIP
007674  012737  055734  001360          MOV     #EM51,EMIN
007702  104001                          ERROR   1               ;"CANNOT READ BS FILES"
007704  000751                          BR      25$
007706
007710  022700  000002          4$:     CMP     #2,R0           ;GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
007714  001014                          BNE     5$              ;NO-SKIP
007716  062737  000002  007540          ADD     #2,2$           ;NEXT SECTOR ADDRESS
```

```
4014  007712  122737  000013  007540          CMPB    #11.,2$         ;PAST APPLICABLE SECTORS?
4015  007720  001021                          BNE     6$              ;NO-SKIP
4016
4017  007722  012737  055734  001360          MOV     #EM51,EMIN
4018  007730  104001                          ERROR   1               ;"CANNOT READ BS FILES"
4019  007732  000732                          BR      25$
4020
4021  007734  062737  000002  007540  5$:     ADD     #2,2$           ;NEXT SECTOR (BS SOFT 24 SECT MODE)
4022  007742  122737  000027  007540          CMPB    #23.,2$         ;PAST APPLICABLE SECTORS?
4023  007750  001005                          BNE     6$              ;NO-SKIP
4024  007752  012737  055734  001360          MOV     #EM51,EMIN
4025  007760  104001                          ERROR   1               ;"CANNOT READ BS FILES
4026  007762  000716                          BR      25$
4027
4028  007764  105037  001103          6$:     CLRB    $ERFLG          ;CLEAR ERROR FLAG
4029  007770  000654                          BR      1$              ;DO NEXT READ
4030
4031  007772  005737  061422          7$:     TST     IBUFF+6         ;CHECK FOR ALIGNMENT PACK
4032  007776  001405                          BEQ     8$              ;NO-SKIP
4033  010000  012737  056027  001360          MOV     #EM52,EMIN
4034  010006  104001                          ERROR   1               ;"ALIGNMENT PACK. DRIVE ABORTING"
4035  010010  000703                          BR      25$
4036
4037  010012  012701  061424          8$:     MOV     #IBUFF+10,R1    ;SET TO START OF BAD SECTOR DATA
4038
4039  010016  022711  177777          9$:     CMP     #-1,(R1)        ;TEST IF WORD ALL ONES (END OF DATA)
4040  010022  001422                          BEQ     11$             ;YES-SKIP
4041  010024  012123                          MOV     (R1)+,(R3)+     ;STORE CYLINDER
4042  010026  012123                          MOV     (R1)+,(R3)+     ;       TRACK AND SECTOR
4043  010030  005205                          INC     R5              ;BUMP ERROR COUNTER
4044  010032  022705  000025                  CMP     #21.,R5         ;DOES IT TOTAL 20 FOR THIS FORMAT?
4045  010036  001367                          BNE     9$              ;NO-TEST AND MORE NEXT ADDRESS
4046  010040  012737  056125  001360          MOV     #EM53,EMIN
4047  010046  104001                          ERROR   1               ;TO MANY BAD SECTORS
4048  010050  043737  001630  001354  10$:     BIC     DRVBIT,$DEVM    ;CLEAR DRIVE FROM TESTING
4049  010056  053737  001630  001666          BIS     DRVBIT,DRVDRP   ;SET DRIVE DROPPED
4050  010064  000137  027764                  JMP     NEWDRV          ;ABORT PASS
4051
4052  010070  005200                  11$:    INC     R0              ;BUMP TO NEXT
4053  010072  022700  000001                  CMP     #1,R0           ;NOW TESTING BS SOFT 26 SECTOR MODE?
4054  010076  001011                          BNE     12$             ;NO-SKIP
4055  010100  012723  177777                  MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
4056  010104  010337  001644                  MOV     R3,BSS26P       ;SET POINTER TO BAD SECTOR SOFTWARE FIELD
4057  010110  112737  000012  007540          MOVB    #12,2$          ;SET TO FIRST SECTOR THIS MODE
4058  010116  000137  007522                  JMP     1$              ;GO READ IT.
4059  010122  022700  000002          12$:    CMP     #2,R0           ;NOW TESTING BS FACT 24 SECTOR MODE?
4060  010126  001014                          BNE     13$             ;NO-SKIP
4061  010130  012723  177777                  MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
4062  010134  112737  000001  007540          MOVB    #1,2$           ;SET TO FIRST SECTOR THIS MODE
4063  010142  010537  001646                  MOV     R5,BS26CT       ;STORE TOTAL BS COUNT 26 SECTOR MODE
4064  010146  005005                          CLR     R5              ;CLEAR COUNTER FOR COUNTING 24 SECT BS
4065  010150  013703  001636                  MOV     BSF24P,R3       ;SET POINTER FOR STORING BS
4066  010154  000137  007522                  JMP     1$              ;GO READ
4067
4068  010160  022700  000003          13$:    CMP     #3,R0           ;NOW TESTING BS SOFT 24 SECTOR MODE
4069  010164  001011                          BNE     14$             ;NO-SKIP
```

```
4070  010166  012723  177777          MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
4071  010172  010337  001642          MOV     R3,BSS24P       ;STORE POINTER TO BSS 24 SECTOR MODE
4072  010176  112737  000013  007540  MOVB    #13,2$          ;GET START OF FIELDS BSS 24 SECT MODE
4073  010204  000137  007522          JMP     1$              ;GO READ IT
4074
4075  010210  012723  177777  14$:    MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
4076  010214  010537  001650          MOV     R5,BS24CT       ;STORE COUNT BSS 24 SECTOR MODE
4077
4078
4079  010220  012700  061042          MOV     #BS26,R0        ;GET START OF BAD SECTOR BUFFER
4080  010224  012703  061252          MOV     #BS24+52.,R3    ;GET END OF BUFFER
4081
4082  010230  022720  000312  27$:    CMP     #312,(R0)+      ;TEST IF ANY SECTORS BAD IN CYL 312
4083  010234  001403                  BEQ     26$             ;YES - GET OUT
4084
4085  010236  020003                  CMP     R0,R3           ;CHECK IF ALL OF BUFFER TESTED
4086  010240  001373                  BNE     27$             ;NO - LOOP
4087  010242  000406                  BR      28$             ;EXIT
4088
4089  010244  012737  057006  001360  26$:    MOV     #DRVABT,EMIN    ;"BAD SECTOR IN AREA FOR TEST"
4090  010252  104001                  ERROR   1
4091  010254  000137  007620          JMP     25$
4092
4093  010260                  28$:
4094
4095                            ;;*****************************************************************
4096                            ;*TEST 21       FORMAT IN 26 SECTOR FORMAT
4097                            ;*
4098                            ;*      FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
4099                            ;*      FORMAT.  VERIFY FORMAT AND THAT DATA LATE DID NOT
4100                            ;*      OCCUR WITH WRITE HEADER OR IN READING DATA
4101                            ;*      BUFFER AFTER READ HEADER.
4102                            ;*
4103                            ;*****************************************************************
4104  010260  000004          TST21:  SCOPE
4105  010262  012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4106  010270  012737  000312  001676  MOV     #312,REFMT      ;SET REFORMAT SWITCH
4107  010276  105037  010421          CLRB    10$             ;CLEAR SECTOR POINTER
4108  010302  105037  010325          CLRB    11$             ;CLEAR TRACK POINTER
4109  010306  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
4110  010310  104003                  ERROR   3               ;BAD INIT ERROR
4111
4112  010312  004437  035530  9$:     JSR     R4,LRLOAD       ;LOAD "L" REGS
4113  010316  000127                  WRHEAD                  ;WRITE HEADER
4114  010320  177676                  -102                    ;WORD CNT FOR 26 SECTOR MODE
4115  010322  063414                  OBUFF                   ;BUFFER
4116  010324     000                  .BYTE   0               ;SECTOR 0
4117  010325     000          11$:    .BYTE   0               ;TRACK 0
4118  010326  000312                  312                     ;CYL 0
4119
4120  010330  004437  041740          JSR     R4,GENCOM       ;GENERATE DATA
4121  010334  000600                  000600                  ;BUILD HEADERS-NO BAD SECTORS
4122  010336  012737  010346  001110  MOV     #111$,$LPERR    ;SET LOCAL LOOP ON ERROR
4123  010344  000402                  BR      1$              ;SKIP INIT
4124  010346                  111$:
4125  010346  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
```

```
4126  010350  104003                          ERROR   3               ;BAD INIT ERROR
4127  010352  104417                  1$:     TLOADRK                 ;LOAD RK REGS
4128  010354  104431                          TWRT112                 ;WAIT FOR INTERRUPT
4129  010356  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4130
4131  010360  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4132  010362  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4133
4134  010364  104415                          SCOP1                   ;INTERNAL LOOP TO 111$
4135
4136  010366  012737  010426  001110          MOV     #112$,$LPERR    ;SET LOCAL LOOP ON ERROR
4137  010374  010203                          MOV     R2,R3           ;BUILD POINTER TO RKDB
4138  010376  062703  000024                  ADD     #RKDB,R3
4139  010402  012701  061414                  MOV     #IBUFF,R1       ;SET POINTER TO BUFFER
4140  010406  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4141  010412  000125                          RDHEAD                  ;READ HEADER
4142  010414  000000                          0                       ;NO WORDS COUNT
4143  010416  000000                          0                       ;NO BUFFER
4144  010420     000                          .BYTE   0               ;SECTOR 0
4145  010421     000                  10$:    .BYTE   0               ;  TRACK 0
4146  010422  000312                          312                     ;  CYL 312
4147
4148  010424  000402                          BR      2$              ;SKIP INIT
4149  010426                          112$:
4150  010426  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4151  010430  104003                          ERROR   3               ;BAD INIT ERROR
4152  010432  104417                  2$:     TLOADRK                 ;LOAD RK REGS
4153  010434  104423                          TWAT16                  ;WAIT FOR INTERRUPT
4154  010436  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4155
4156  010440  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4157  010442  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL ERRORS
4158  010444  012700  000003                  MOV     #3,R0           ;SET COUNT
4159  010450  011321                  5$:     MOV     (R3),(R1)+      ;GET RKDB
4160  010452  104420                          TGETRK                  ;GET RK REGS
4161  010454  032737  100000  001550          BIT     #DLT,T.CS2      ;TEST IF DATA LATE
4162  010462  001410                          BEQ     3$              ;NO-SKIP
4163  010464  012737  054527  054156          MOV     #EMDLT,EM13
4164  010472  012737  056675  061010          MOV     #EMRDB,DF011A
4165  010500  104013                          ERROR   13              ;"DATA LATE SET RESULT OF DB READ
4166  010502  104415                          SCOP1                   ;LOCAL LOOP TO 112$
4167
4168  010504  032737  100000  001540  3$:     BIT     #CERR,T.CS1     ;TEST IF CONT ERROR SET
4169  010512  001410                          BEQ     4$              ;NO-SKIP
4170  010514  012737  056721  001500          MOV     #EMCERR,EM13N
4171  010522  012737  056675  061010          MOV     #EMRDB,DF011A
4172  010530  104013                          ERROR   13              ;"CERR SET RESULT OF READ DB
4173  010532  104415                          SCOP1                   ;LOCAL LOOP TO 112$
4174  010534  005300                  4$:     DEC     R0              ;DEC COUNT
4175  010536  001344                          BNE     5$              ;LOOP IF NOT ZERO
4176  010540  012737  010550  001110          MOV     #117$,$LPERR    ;SET LOCAL LOOP 117$
4177  010546  000402                          BR      7$              ;SKIP INIT
4178  010550                          117$:
4179  010550  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4180  010552  104003                          ERROR   3               ;BAD INIT ERROR
4181  010554  004437  036050          7$:     JSR     R4,RDSTHD       ;GO READ & SEQUENCE HEADERS
```

```
4182  010560  104421                          TCHKOP                  ;CONTROLLER ERROR RETURN
4183  010562  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4184  010564  104013                          ERROR   13              ;"DATA LATE SET RESULT OF DATA BUFFER READ"
4185  010566  104002                          ERROR   2               ;"OPERATION TO SLOW" MESSAGE
4186                                                                  ;OR "HEADER 0 NOT FOUND" MESSAGE
4187
4188  010570  004437  041740                  JSR     R4,GENCOM
4189  010574  100200                          100200                  ;COMPARE IBUF & OBUFF (HEADERS)
4190  010576  000414                          BR      6$              ;GOOD RETURN-NO MISCOMPARES
4191  010600  104015                          ERROR   15              ;REPORT 1ST MSCOMPARES
4192
4193  010602  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
4194  010606  005300                  12$:    DEC     R0              ;DECREMENT IT
4195  010610  001407                          BEQ     6$              ;EXIT IF ZERO
4196  010612  004437  041740                  JSR     R4,GENCOM
4197
4198  010616  040000                          040000                  ;RESUME COMPARE
4199  010620  000403                          BR      6$              ;GOOD RETURN-NO MORE ERRORS
4200  010622  104016                          ERROR   16              ;REPORT NEXT ERROR LINE
4201  010624  000770                          BR      12$             ;LOOP
4202  010626  104415                          SCOP1                   ;LOCAL ERROR LOOP TO 117$
4203
4204  010630  105737  001607          6$:     TSTB    L.DT            ;WAS TRACK 1 JUST DONE?
4205  010634  001010                          BNE     8$              ;YES-SKIP
4206
4207  010636  112737  000001  010325          MOVB    #1,11$          ;CHANGE PARAM TO LOAD "L" WITH
4208  010644  112737  000001  010421          MOVB    #1,10$          ;TRACK 2
4209  010652  000137  010312                  JMP     9$              ;JUMP TO DO ENTIRE TEST ON TRK 1
4210
4211  010656                          8$:
4212
4213                                          .SBTTL  **HEADER RECOGNITION TESTS
4214
4215                                          ;;*****************************************************************
4216                                          ;*TEST 22       BAD SECTOR ERROR
4217                                          ;*
4218                                          ;*      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
4219                                          ;*      SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1
4220                                          ;*      (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
4221                                          ;*      AND ALL OTHER SECTORS GOOD.
4222                                          ;*
4223                                          ;*      ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0,
4224                                          ;*      SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS.  ISSUE A
4225                                          ;*      WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS.
4226                                          ;*      MAKE SURE BAD SECTOR ERROR SET.  ISSUE A WRITE DATA
4227                                          ;*      OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2.  MAKE
4228                                          ;*      SURE NO ERROR SETS.
4229                                          ;*
4230                                          ;;*****************************************************************
4231  010656  000004                  TST22:  SCOPE
4232  010660  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4233  010666  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4234  010674  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4235  010676  104003                          ERROR   3               ;BAD INIT ERROR
4236
4237  010700  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
                                        WRHEAD                  ;WRITE HEADER
                                        -102                    ;WORD COUNT FOR 26 SECTOR MODE
                                        OBUFF                   ;BUFFER ADDRESS
                                        .BYTE   0               ;SECTOR
                                        .BYTE   0               ;TRACK
                                        312                     ;CYLINDER

                                        JSR     R4,GENCOM       ;GENERATE HEADERS
                                        600                     ;WITH NO BS BITS

                                        MOV     #OBUFF+2,R0     ;RESET BIT 15 IN WORD 2 OF
                                        BIC     #BIT15,(R0)+    ;SECTOR 0 HEADER AND BIT 14
                                        BIC     #BIT15,(R0)+    ;IN WORD 2 OF SECTOR 1 HEADER.
                                        TST     (R0)+           ;ALSO CORRECT THE VRC
                                        BIC     #BIT14,(R0)+
                                        BIC     #BIT14,(R0)

                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT112                 ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                                        TCHKOP                  ;CHECK IF ANY ERRORS
                                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
                                        MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR
                                4$:     TSSINIT
                                        ERROR   3
                                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        WRDATA                  ;WRITE DATA
                                        -400                    ;WORD COUNT
                                        OBUFF                   ;BUS ADDRESS
                                5$:     .BYTE   0               ;SECT 0
                                        .BYTE   0               ;TRACK 0
                                        312                     ;CYL 312

                                1$:     TLOADRK                 ;LOAD RK REGS
                                        TWAT32                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                                        CMP     #2,5$           ;JUST READ SECTOR 2?
                                        BEQ     6$              ;YES - SKIP

                                        TCHKWE                  ;CHECK OPERATION WITH ERROR
                                        0
                                        100                     ;EXPECTED BSE
                                        0
                                        ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES

                                        SCOP1                   ;LOCAL ERROR LOOP TO 4$

                                        CMPB    #2,5$           ;WAS SECTOR SET TO 2
                                        BEQ     7$              ;YES-SKIP
                                        INCB    5$              ;BUMP TO NEXT SECTOR
                                        BR      4$              ;LOOP

                                6$:     TCHKOP                  ;CHECK FOR GOOD OPERATION
```

```
                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

                7$:
                ;;*****************************************************************
                ;*TEST 23        HEADER VRC ERROR
                ;*
                ;*      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
                ;*      16 SECTORS WITH BAD HEADER VRC.  ISSUE A WRITE DATA
                ;*      OF EACH OF THE SECTORS WITH A BAD HEADER VRC.  MAKE
                ;*      SURE HEADER VRC ERROR SETS.  ISSUE A WRITE DATA TO
                ;*      A GOOD HEADER AND MAKE SURE NO ERROR OCCURS.
                ;*
                ;;*****************************************************************
011070  000004 TST23:  SCOPE
011072  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
011100  012737  000312  001676        MOV     #312,REFMT      ;SET REFORMAT SWITCH
011106  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
011110  104003                        ERROR   3               ;BAD INIT ERROR

011112  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
011116  000127                        WRHEAD                  ;WRITE HEADER
011120  177676                        -102                    ;WORD COUNT
011122  063414                        OBUFF                   ;BUFF ADD
011124     000                        .BYTE   0               ;SECT
011125     000                        .BYTE   0               ;TRACK
011126  000312                        312                     ;CYL

011130  004437  041740                JSR     R4,GENCOM
011134  000600                        600                     ;BUILD HEADERS NO BSE

011136  012700  063420                MOV     #OBUFF+4,R0     ;GET ADDRESS OF VRC HDR0
011142  012703  000001                MOV     #BIT0,R3        ;SET FOR BIT CHANGE SELECT
011146  030310          1$:           BIT     R3,(R0)         ;CHECK A VRC BIT
011150  001402                        BEQ     2$              ;SKIP IF ZERO
011152  040310                        BIC     R3,(R0)         ;ELSE CLEAR IT
011154  000401                        BR      3$              ;SKIP
011156  050310          2$:           BIS     R3,(R0)         ;IF ZERO SET IT
011160  062700  000006    3$:         ADD     #6,R0           ;BUMP TO NEXT VRC WORD

011164  006303                        ASL     R3              ;SHIFT THE SELECT
011166  001367                        BNE     1$              ;IF BIT NOT SHIFTED OUT-LOOP

011170  104417                        TLOADRK                 ;LOAD RK REGS
011172  104431                        TWAIT12                 ;WAIT FOR INTERRUPT
011174  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
011176  104421                        TCHKOP                  ;CHECK OPERATION COMPLETE
011200  104004                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

011202  012737  011210  001110        MOV     #4$,$LPERR      ;SET LOCAL LOOP
011210  104416          4$:           TSSINIT                 ;CLEAR SUBSYSTEM
011212  104003                        ERROR   3               ;BAD INIT ERROR

011214  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
011220  000123                        WRDATA                  ;WRITE DATA
011222  177400                        -400                    ;WORD COUNT
011224  063414                        OBUFF                   ;BUFFER ADD
```

```
 4350  011226  000000                    5$:     .BYTE   0               ;SECT
 4351  011227  000000                            .BYTE   0               ;TRACK
 4352  011230  000312                            312                     ;CYL
 4353
 4354  011232  104417                            TLOADRK                 ;LOAD RK REG
 4355  011234  104424                            TWAT32                  ;WAIT FOR INTERRUPT
 4356  011236  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 4357
 4358  011240  022737  000020  011226            CMP     #16.,5$         ;WAS THIS WRITE SECTOR 16?
 4359  011246  001415                            BEQ     6$              ;YES-SKIP
 4360                                                                    ;ELSE
 4361  011250  104422                            TCHKWE                  ;CHECK OPERATION WITH ERROR
 4362  011252  000000                            0
 4363  011254  000040                            40                      ;HVRC  EM EXPECTED
 4364  011256  000000                            0
 4365  011260  104004                            ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
 4366
 4367  011262  104415                            SCOP1                   ;LOCAL LOOP TO 4$
 4368
 4369  011264  105237  011226                    INCB    5$              ;BUMP SECTOR IN "L" REG
 4370  011270  022737  000016  011226            CMP     #16,5$          ;IF SECTOR IS 16 OR LESS
 4371  011276  003744                            BLE     4$              ;LOOP
 4372  011300  000402                            BR      7$              ;ELSE EXIT
 4373  011302  104421                    6$:     TCHKOP                  ;CHECK LAST OPERATION NO ERRORS
 4374  011304  104004                            ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
 4375
 4376  011306                            7$:
 4377                                    ;;*****************************************************************
 4378                                    ;*TEST 24       BAD SECTOR ERROR AND HVRC ERROR
 4379                                    ;*
 4380                                    ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
 4381                                    ;*      BOTH A BAD SECTOR ERROR AND HEADER VRC.  ISSUE A WRITE DATA
 4382                                    ;*      TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY HEADER VRC
 4383                                    ;*      ERROR SETS.
 4384                                    ;*
 4385                                    ;*****************************************************************
 4386  011306  000004                    TST24:  SCOPE
 4387  011310  012737  000012  001262            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 4388  011316  012737  000312  001676            MOV     #312,REFMT      ;SET REFORMAT SWITCH
 4389  011324  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
 4390  011326  104003                            ERROR   3               ;BAD INIT ERROR
 4391
 4392  011330  004437  035530                    JSR     R4,LRLOAD       ;LOAD "L" REG
 4393  011334  000127                            WRHEAD                  ;WRITE HEADER
 4394  011336  177676                            -102                    ;WORD CNT FOR 26 SECTOR MODE
 4395  011340  063414                            OBUFF                   ;BUFF ADD
 4396  011342  000000                            .BYTE   0               ;SECTOR
 4397  011343  000000                            .BYTE   0               ;TRACK
 4398  011344  000312                            312                     ;CYLINDER
 4399
 4400  011346  004437  041740                    JSR     R4,GENCOM
 4401  011352  000600                            600                     ;BUILD HEADERS-NO BSE
 4402
 4403  011354  042737  100000  063416            BIC     #BIT15,OBUFF+2  ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.
 4404
 4405  011362  104417                            TLOADRK                 ;LOAD RK REGS
```

```
                                        TWAT112                 ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                                        TCHKOP                  ;CHECK FOR ANY ERRORS
                                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

                                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        WRDATA                  ;WRITE DATA
                                        -400                    ;WORD COUNT
                                        OBUFF                   ;BUFF ADD
                                        .BYTE   0               ;SECTOR
                                        .BYTE   0               ;TRACK
                                        312                     ;CYLINDER

                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT32                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                                        TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERR
                                        0
                                        40                      ;HVRC ERR EXPECTED
                                        0
                                        ERROR   4 ; OR 5,6,7    ;REPORT ALL DISCREPENCIES

                        ;;**************************************************************
                        ;*TEST 25        OPERATION INCOMPLETE
                        ;*
                        ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE
                        ;*      WRONG FORMAT.  ISSUE A WRITE DATA OF 400 TO CYLINER 0,
                        ;*      TRACK 0, SECTOR 21.  MAKE SURE OPI SET.
                        ;*
                        ;**************************************************************
000004                  TST25:  SCOPE
012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
104416                          TSSINIT                 ;CLEAR SUBSYSTEM
104003                          ERROR   3               ;BAD INIT ERROR

004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
000127                          WRHEAD                  ;WRITE HEADER
177676                          -102                    ;WORD COUNT FOR 26 SECT MODE
063414                          OBUFF                   ;BUFF ADD
   000                          .BYTE   0               ;SECTOR
   000                          .BYTE   0               ;TRACK
000312                          312                     ;CYLINDER

004437  041740                  JSR     R4,GENCOM
000600                          600                     ;BUILD HEADERS-NO BSE ERRORS

052737  001000  063614          BIS     #BIT9,OBUFF+200 ;CHANGE FORMAT IN SECTOR 25
052737  001000  063616          BIS     #BIT9,OBUFF+202 ;CORRECT THE VRC

104417                          TLOADRK                 ;LOAD RK REGS
104431                          TWAT112                 ;WAIT FOR INTERRUPT
104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
```

```
4462  011522  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4463  011524  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4464
4465  011526  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4466  011532  000123                          WRDATA                  ;WRITE DATA
4467  011534  177400                          -400                    ;400 WORDS
4468  011536  063414                          OBUFF                   ;BUFF ADD
4469  011540     025                          .BYTE   25              ;SECTOR 25
4470  011541     000                          .BYTE   0               ;TRACK 0
4471  011542  000312                          312                     ;CYL 312
4472
4473  011544  104417                          TLOADRK                 ;LOAD RK REGS
4474  011546  104425                          TWAT48                  ;WAIT FOR INTERRUPT
4475  011550  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
4476
4477  011552  104422                          TCHKWE                  ;CHECK OPERATION EXPECTED ERROR
4478  011554  000000                          0
4479  011556  000020                          20                      ;OPI EXPECTED
4480  011560  000000                          0
4481  011562  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
4482                          ;;*****************************************************************
4483                          ;*TEST 26        OPI WITH HVRC ERROR
4484                          ;*
4485                          ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
4486                          ;*      ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
4487                          ;*      ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312,
4488                          ;*      TRACK 0, SECTOR 17.  THAT BOTH OPERATION INCOMPLETE
4489                          ;*      AND HEADER VRC SET.
4490                          ;*
4491                          ;;*****************************************************************
4492  011564  000004          TST26:  SCOPE
4493  011566  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4494  011574  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4495  011602  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4496  011604  104003                          ERROR   3               ;BAD INIT ERROR
4497
4498  011606  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4499  011612  000127                          WRHEAD                  ;WRITE HEADER
4500  011614  177676                          -102                    ;WORD COUNT FOR 26 SECT MODE
4501  011616  063414                          OBUFF                   ;BUS ADDRESS
4502  011620     000                          .BYTE   0               ;SECTOR
4503  011621     000                          .BYTE   0               ;TRACK
4504  011622  000312                          312                     ;CYLINDER
4505
4506  011624  004437  041740                  JSR     R4,GENCOM
4507  011630  000600                          600                     ;BUILD HEADER- NO BSE ERRORS
4508
4509  011632  012700  063550                  MOV     #OBUFF+134,R0   ;GET ADDRESS 2ND WORD HDR 17(8)
4510  011636  052720  001000                  BIS     #BIT9,(R0)+     ;SET FORMAT 24 SECT PER TRACK
4511  011642  052720  001000                  BIS     #BIT9,(R0)+     ;SET VRC BIT
4512  011646  062700  000004                  ADD     #4,R0           ;BUMP TO HVRC WORD HDR 20(8)
4513  011652  032710  000001                  BIT     #BIT0,(R0)      ;TEST BIT 0
4514  011656  001403                          BEQ     1$              ;RESET-SKIP
4515  011660  042710  000001                  BIC     #BIT0,(R0)      ;CLEAR BIT
4516  011664  000402                          BR      2$
4517  011666  052710  000001          1$:     BIS     #BIT0,(R0)      ;SET BIT
```

```
                                        ;FORCE OPI AND HVRC ERROR
011672  104417                  2$:   TLOADRK                 ;LOAD RK REGS
011674  104431                        TWAIT12                 ;WAIT FOR INTERRUPT
011676  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

011700  104421                        TCHKOP                  ;CHECK FOR ANY ERRORS
011702  104004                        ERROR   4 ; OR 5,6,7    ;YES-REPORT ALL ERRORS

011704  004437  035530              JSR     R4,LRLOAD         ;LOAD "L" REGS
011710  000123                      WRDATA                    ;WRITE DATA
011712  177400                      -400                      ;400 WORDS
011714  063414                      OBUFF                     ;BUFF ADDRESS
011716     017                      .BYTE   17                ;SECT 17
011717     000                      .BYTE   0                 ;TRACK 0
011720  000312                      312                       ;CYLINDER 312

011722  104417                      TLOADRK                   ;LOAD RK REGS
011724  104425                      TWAIT48                   ;WAIT FOR INTERRUPT
011726  104002                      ERROR   2                 ;TO SLOW/NOT COMPLETE

011730  104422                      TCHKWE                    ;CHECK WITH EXPECTED ERROR
011732  000000                      0
011734  000060                      60                        ;HVRC ERR & OPI EXPECTED
011736  000000                      0
011740  104004                      ERROR   4  ;OR 5,6,7
                        ;;**********************************************************************
                        ;*TEST 27       HVRC IGNORE ON NON-ADDRESSED SECTOR
                        ;*
                        ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
                        ;*      ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
                        ;*      AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
                        ;*      END OF THE OPERATION
                        ;*
                        ;**********************************************************************
011742  000004         ;ST27:  SCOPE
011744  012737  000012  001262        MOV     #10.,$TIMES       ;;DO 10. ITERATIONS
011752  012737  000312  001676        MOV     #312,REFMT        ;SET REFORMAT SWITCH

011760  104416                          .NIT                    ;CLEAR SUBSYSTEM
011762  104003                        ROR   3                   ;BAD INIT ERROR

011764  004437  035530              JSR     R4,LRLOAD         ;LOAD "L" REGISTERS
011770  000127                      WRHEAD                    ;WRITE HEADER
011772  177676                      -102                      ;WORD COUNT FOR 26 SECTOR MODE
011774  063414                      OBUFF                     ;BUFF ADD
011776     000                      .BYTE   0                 ;SECTOR
011777     000                      .BYTE   0                 ;TRACK
012000  000312                      312                       ;CYLINDER

012002  004437  041740              JSR     R4,GENCOM
012006  000600                      600                       ;BUILD HEADERS-NO BSE ERRORS

012010  012700  063560              MOV     #OBUFF+144,R0     ;ADDRESS OF HEAD 20 HVRC WORD
012014  012701  000002              MOV     #BIT1,R1          ;BIT 1 CONSTANT
012020  030110                      BIT     R1,(R0)           ;TEST BIT 1 SET
012022  001402                      BEQ     1$                ;RESET-SKIP
```

```
                                        BIC     R1,(R0)         ;ELSE CLEAR BIT 1
                                        BR      2$              ;SKIP
                                1$:     BIS     R1,(R0)         ;SET BIT 1

                                2$:     TLOADRK                 ;LOAD RK REGS
                                        TWAIT12                 ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE

                                        TCHKOP                  ;CHECK FOR ANY ERROR
                                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

012044  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGISTER
012050  000123                          WRDATA                  ;WRITE DATA
012052  177400                          -400                    ;WORD COUNT
012054  063414                          OBUFF                   ;BUFF ADD
012056     021                          .BYTE   21              ;SECTOR
012057     000                          .BYTE   0               ;TRACK
012060  000312                          312                     ;CYLINDER
012062  104417                          TLOADRK                 ;LOAD RK REGS
012064  104424                          TWAIT32                 ;WAIT FOR INTERRUPT
012066  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

012070  104421                          TCHKOP                  ;CHECK FOR ANY ERROR
012072  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS.


                                .SBTTL  **DATA TRANSFER TESTS

                                ;;*****************************************************************
                                ;*TEST 30       WRITE AND READ ONE SECTOR
                                ;*
                                ;*      FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 0
                                ;*      TO AGREED WITH BAD SECTOR INFORMATION.  ISSUE A WRITE DATA
                                ;*      OF ONE SECTOR ON CYLINDER 312, TRACK 0.  READ IT BACK TO
                                ;*      MAKE SURE IT AGREES WITH WHAT IS WRITTEN.
                                ;*
                                ;;*****************************************************************
000004                  TST30:  SCOPE
012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
104416                          TSSINIT                 ;CLEAR SUBSYSTEM
104003                          ERROR   3               ;BAD INIT ERROR

012116  012737  000312  012144          MOV     #312,7$         ;PRESET CYL POINTER
012124  105037  012143                  CLRB    2$              ;CLEAR TRACK POINTER

012130  004437  035530          1$:     JSR     R4,LRLOAD       ;LOAD "L" REG
012134  000127                          WRHEAD                  ;WRITE HEADER
012136  177676                          -102                    ;WORD COUNT FOR 26 SECTOR MODE
012140  063414                          OBUFF                   ;BUFF ADDRESS
012142     000                          .BYTE   0               ;SECTOR
012143     000                  2$:     .BYTE   0               ;TRACK
012144  000312                  7$:     312                     ;CYLINDER

012146  004437  041740                  JSR     R4,GENCOM
012152  001200                          1200                    ;BUILD HDRS-INCLUDE BAD SECTORS
```

```
                                                TLOADRK                 ;LOAD RK REGS
                                                TWAT112                 ;WAIT FOR INTERRUPT
                                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                                                TCHKOP                  ;CHECK FOR ANY ERRORS
                                                ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

012166  022737  000313  012144                CMP     #313,7$         ;TEST IF DONE 313 TK 0
012174  001414                                BEQ     3$              ;YES - SKIP
012176  123727  012143  000002                CMPB    2$,#2           ;DID WE JUST FORMAT TRACK 2
012204  001403                                BEQ     8$              ;YES-SKIP
012206  105237  012143                        INCB    2$              ;BUMP TO NEXT TRACK
012212  000746                                BR      1$              ;GO FORMAT NEXT TRACK

012214  105037  012143          8$:           CLRB    2$              ;CLEAR TRACK POINTER
012220  005237  012144                        INC     7$              ;BUMP CYL TO 313
012224  000741                                BR      1$              ;GO FORMAT 313 TK 0

012226  004437  035530          3$:           JSR     R4,LRLOAD       ;LOAD "L" REGS
012232  000123                                WRDATA                  ;WRITE DATA
012234  177400                                -400                    ;ONE SECTOR WORD COUNT
012236  063414                                OBUFF                   ;BUFF ADDRESS
012240     012                                .BYTE   12              ;SECTOR 12
012241     000                                .BYTE   0               ;TRACK 0
012242  000312                                312                     ;CYLINDER 312

012244  004437  041740                        JSR     R4,GENCOM
012250  000001                                1                       ;BUILD DATA PATTERN 1
012252  000400                                400                     ;400 WORDS LONG
012254  012737  012262  001110                MOV     #4$,$LPERR      ;SET FOR LOCAL LOOP
012262  104417          4$:                   TLOADRK                 ;LOAD RK REGS
012264  104431                                TWAT112                 ;WAIT FOR INTERRUPT
012266  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

012270  104421                                TCHKOP                  ;CHECK FOR ANY ERRORS
012272  104004                                ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

012274  004437  035530                        JSR     R4,LRLOAD       ;LOAD "L" REGS
012300  000121                                RDDATA                  ;READ DATA
012302  177400                                -400                    ;400 WORDS
012304  061414                                IBUFF                   ;BUFF ADD
012306     012                                .BYTE   12              ;SECTOR 12
012307     000                                .BYTE   0               ;TRACK 0
012310  000312                                312                     ;CYL 312

012312  104417                                TLOADRK                 ;LOAD RK
012314  104424                                TWAT32                  ;WAIT FOR INTERRUPT
012316  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE

012320  104421                                TCHKOP                  ;CHECK FOR ANY ERRORS
012322  104004                                ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

012324  004437  041740                        JSR     R4,GENCOM
012330  100001                                100001                  ;GO COMPARE DATA TO PATTERN 1
012332  000400                                400                     ;400 WORDS LONG
```

```
                        000413                          BR      6$              ;GOOD RETURN-NO DATA ERRORS
                        104015                          ERROR   15              ;ERROR RETURN

                        013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
                        005300                  5$:     DEC     R0              ;DEC LIMIT
                        001406                          BEQ     6$              ;EXIT IF 0
                        004437  041740                  JSR     R4,GENCOM
                        040000                          040000                  ;CONTINUE COMPARE
                        000402                          BR      6$              ;EXIT IF NO MORE ERRORS
                        104016                          ERROR   16              ;ELSE REPORT MISCOMPARE
                        000770                          BR      5$              ;LOOP
                        005037  001676          6$:     CLR     REFMT           ;CLEAR REFORMAT SWITCH

                                                ;;*****************************************************************
                                                ;*TEST 31        WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT
                                                ;*
                                                ;*      ISSUE A WRITE DATA OF ONE SECTOR TO CYLINDER 312,
                                                ;*      TRACK 2, SECTOR 12 WITH INHIBIT BUS
                                                ;*      ADDRESS INCREMENT. READ DATA BACK TO MAKE SURE
                                                ;*      EVERY WORD IS THE SAME AND CORRECT.
                                                ;*
                                                ;;*****************************************************************
                        000004                  TST31:  SCOPE
                        012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

                        104416                          TSSINIT                 ;CLEAR SUBSYSTEM
                        104003                          ERROR   3               ;BAD INIT ERROR

                        004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
                        000123                          WRDATA                  ;WRDATA
                        177400                          -400                    ;-400 WORDS
                        063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
                            012                         .BYTE   12              ;SECTOR 12
                            001                         .BYTE   1               ;TRACK 1
                        000312                          312                     ;CYLINDER 312

                        052737  000020  001610          BIS     #BAI,L.CS2      ;SET INCREMENT INHIBIT
                        004437  041740                  JSR     R4,GENCOM       ;BUILD PATTERN
                        000016                          16                      ;PATTERN 16
                        000400                          400                     ;400 WORDS

                        104417                          TLOWORK                 ;LOAD RK REGS
                        104430                          TWAT96                  ;WAIT FOR INTERRUPT
                        104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                        104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
                        104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

                        004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
                        000121                          RDDATA                  ;RDDATA
                        177400                          -400                    ;-400 WORDS
                        061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
                            012                         .BYTE   12              ;SECTOR 12
                            001                         .BYTE   1               ;TRACK 1
                        000312                          312                     ;CYLINDER 312
```

```
4742  012470  012700  000377          MOV     #377,R0         ;SET COUNT TO SET OBUFF TO BE
4743  012474  012701  063416          MOV     #OBUFF+2,R1     ;ALL THE FIRST WORD OF PATTERN
4744  012500  012703  063414          MOV     #OBUFF,R3
4745
4746  012504  011321          1$:     MOV     (R3),(R1)+      ;MOV THE WORD
4747  012506  005300                  DEC     R0
4748  012510  001375                  BNE     1$              ;LOOP UNTIL ALL WORDS SET
4749
4750  012512  104417                  TLOADRK                 ;LOAD RK REGS
4751  012514  104424                  TWAT32                  ;WAIT FOR INTERRUPT
4752  012516  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4753
4754  012520  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4755  012522  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4756
4757  012524  004437  041740          JSR     R4,GENCOM       ;COMPARE THE DATA
4758  012530  100000                  100000
4759  012532  000400                  400
4760  012534  000413                  BR      2$
4761  012536  104015                  ERROR   15
4762  012540  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
4763  012544  005300          64$:    DEC     R0              ;DECREMENT COUNT
4764  012546  001406                  BEQ     65$             ;IF ZERO - EXIT
4765  012550  004437  041740          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4766  012554  040000                  40000
4767  012556  000402                  BR      65$             ;NO MORE ERRORS - EXIT
4768  012560  104016                  ERROR   16              ;REPORT NEXT ERROR
4769  012562  000770                  BR      64$             ;LOOP
4770  012564                  65$:
4771
4772  012564                  2$:
4773                                  ;;*****************************************************************
4774                                  ;*TEST 32       WRITE DATA ADDRESS GREATER THAN 32K
4775                                  ;*
4776                                  ;*      ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
4777                                  ;*      MAKE SURE CORRECT DATA IS ON DISK.
4778                                  ;*
4779                                  ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
4780                                  ;*              OF MEMORY IS PRESENT.
4781                                  ;*
4782                                  ;;*****************************************************************
4783  012564  000004          T32:    SCOPE
4784  012566  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4785  012574  123727  001326  000001          CMPB    $MAMS1,#1       ;TEST IF >32K MEM
4786  012602  002016                  BGE     2$              ;YES-SKIP
4787
4788  012604  032737  000002  001664          BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT ALREADY MADE
4789  012612  001011                  BNE     1$              ;YES -SKIP
4790
4791  012614  104401  051471                  TYPE    ,OPR011         ;"INSUFFICIENT MEMORY DATA TRANSFER WITH
4792  012620  104401  051634                  TYPE    ,OPR012         ;ADDRESS >32K
4793  012624  052737  000002  001664          BIS     #MEMSZB,OPTFLG  ;SET FLAG
4794  012632  104401  051650                  TYPE    ,OPR015         ;BYPASSED"
4795  012636  000470                  1$:     BR      4$              ;EXIT
4796  012640  012737  012646  001110  2$:     MOV     #5$,$LPERR      ;SET LOCAL LOOP ON ERROR ADDRESS
4797  012646                  5$:
```

```
4798  012646  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
4799  012650  104003                        ERROR   3               ;BAD INIT ERROR
4800
4801  012652  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
4802  012656  000123                        WRDATA                  ;WRITE DATA
4803  012660  177400                        -400                    ;400 WORDS
4804  012662  177770                        177770                  ;BUS ADDRESS IN 32K -10 BYTES
4805  012664     016                        .BYTE   16              ;SECTOR 16
4806  012665     000                        .BYTE   0               ;TRACK 0
4807  012666  000312                        312                     ;CYLINDER 312
4808  012670  004437  041740                JSR     R4,GENCOM       ;GENERATE DATA
4809  012674  010010                        10010                   ;PATTERN 10, MEM. MANAGEMENT FOR DEST.
4810  012676  001777                        1777                    ;RELOCATION ARGUMENT
4811  012700  000400                        400                     ;400 WORDS
4812
4813  012702  104417                        TLOADRK                 ;LOAD RK REGS
4814  012704  104430                        TWAT96                  ;WAIT FOR INTERRUPT
4815  012706  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4816
4817  012710  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4818  012712  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4819  012714  104415                        SCOP1                   ;LOCAL LOOP ON ERROR TO 5$
4820
4821  012716  004437  041740                JSR     R4,GENCOM       ;CLEAR IBUFF TO 1'S.
4822  012722  002007                        2007
4823  012724  001000                        1000
4824
4825  012726  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
4826  012732  000121                        RDDATA                  ;RDDATA
4827  012734  177400                        -400                    ;-400 WORDS
4828  012736  061414                        IBUFF                   ;IBUFF IS BUFF ADDRESS
4829  012740     016                        .BYTE   16              ;SECTOR 16
4830  012741     000                        .BYTE   0               ;TRACK 0
4831  012742  000312                        312                     ;CYLINDER 312
4832  012744  104417                        TLOADRK                 ;LOAD RK REGS
4833  012746  104424                        TWAT32                  ;WAIT FOR INTERRUPT
4834  012750  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4835  012752  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4836  012754  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4837  012756  004437  041740                JSR     R4,GENCOM       ;COMPARE
4838  012762  110000                        110000                  ;MEMORY MANAGEMENT FOR DESTINATION
4839  012764  001777                        1777                    ;RELOCATION ARGUMENT
4840  012766  000400                        400                     ;400 WORDS
4841  012770  000413                        BR      4$              ;NO ERROR-SKIP
4842  012772  104015                        ERROR   15              ;REPORT FIRST MISCOMPARE
4843  012774  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
4844  013000  005300                6$:     DEC     R0              ;DECREMENT COUNT
4845  013002  001406                        BEQ     65$             ;IF ZERO - EXIT
4846  013004  004437  041740                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4847  013010  050000                        50000
4848  013012  000402                        BR      65$             ;NO MORE ERRORS - EXIT
4849  013014  104016                        ERROR   16              ;REPORT NEXT ERROR
4850  013016  000770                        BR      64$             ;LOOP
4851  013020                        65$:
4852
4853  013020                        4$:
```

```
                                ;;*****************************************************************
                                ;*TEST 33        READ DATA ADDRESS GREATER THAN 32K
                                ;*
                                ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 177770.
                                ;*      CHECK MEMORY FOR CORRECT TRANSFER.
                                ;*
                                ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
                                ;*              OF MEMORY IS PRESENT.
                                ;*
                                ;*****************************************************************
013020  000004                  TST33:  SCOPE
013022  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
013030  123727  001326  000001          CMPB    $MAMS1,#1       ;CHECK IF >32K MEMORY
013036  002001                          BGE     2$              ;YES-SKIP

013040  000462                  1$:     BR      5$              ;EXIT

013042  012737  013050  001110  2$:     MOV     #3$,$LPERR      ;SET LOCAL ERROR LOOP

013050                          3$:
013050  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
013052  104003                          ERROR   3               ;BAD INIT ERROR
013054  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
013060  000123                          WRDATA                  ;WRDATA
013062  177400                          -400                    ;-400 WORDS
013064  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
013066     017                          .BYTE   17              ;SECTOR 17
013067     000                          .BYTE   0               ;TRACK 0
013070  000312                          312                     ;CYLINDER 312
013072  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA IN OBUFF
013076  000011                          11                      ;PATTERN 11
013100  000400                          400                     ;400 WORDS

013102  104417                          TLOADRK                 ;LOAD RK REGS
013104  104430                          TWAT96                  ;WAIT FOR INTERRUPT
013106  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

013110  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
013112  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
013114  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REG
013120  000121                          RDDATA                  ;READ DATA
013122  177400                          -400                    ;400 WORDS
013124  177770                          177770                  ;ACCROSS 32K BOUNDARY
013126     017                          .BYTE   17              ;SECTOR 17
013127     000                          .BYTE   0               ;TRACK 0
013130  000312                          312                     ;CYL 312

013132  104417                          TLOADRK                 ;LOAD RK REGS
013134  104424                          TWAT32                  ;WAIT FOR INTERRUPT
013136  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
013140  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
013142  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
013144  004437  041740                  JSR     R4,GENCOM       ;COMPARE DATA
013150  120000                          120000                  ;MEMORY MANAGEMENT WITH SOURCE
013152  001777                          1777                    ;RELOCATION ARGUMENT
013154  000400                          400                     ;COMPARE 400 WORDS
```

```
4910  013156  000413                          BR      5$              ;NO MISCOMPARE-EXIT
4911  013160  104015                          ERROR   15              ;REPORT FIRST MISCOMPARE
4912  013162  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
4913  013166  005300                  64$:    DEC     R0              ;DECREMENT COUNT
4914  013170  001406                          BEQ     65$             ;IF ZERO - EXIT
4915  013172  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4916  013176  060000                          60000
4917  013200  000402                          BR      65$             ;NO MORE ERRORS - EXIT
4918  013202  104016                          ERROR   16              ;REPORT NEXT ERROR
4919  013204  000770                          BR      64$             ;LOOP
4920  013206                          65$:
4921  013206                          5$:
4922                                  ;;*****************************************************************
4923                                  ;*TEST 34       WRITE DATA ADDRESS GREATER THAN 64K
4924                                  ;*
4925                                  ;*      ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.
4926                                  ;*      MAKE SURE CORRECT DATA IS ON DISK.
4927                                  ;*
4928                                  ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
4929                                  ;*              OF MEMORY IS PRESENT.
4930                                  ;*
4931                                  ;;*****************************************************************
4932  013206  000004                  T$T34:  SCOPE
4933  013210  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4934  013216  123727  001326  000002          CMPB    $MAMS1,#2       ;CHECK IF >64K MEMORY
4935  013224  002016                          BGE     2$              ;YES-SKIP
4936  013226  032737  000002  001664  6$:     BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
4937  013234  001011                          BNE     1$              ;NO-SKIP
4938
4939  013236  104401  051471                  TYPE    ,OPR011         ;"INSUFFICIENT MEMORY-DATA XFER WITH
4940  013242  104401  051640                  TYPE    ,OPR013         ;ADDRESS >64K
4941  013246  104401  051650                  TYPE    ,OPR015         ;BYPASSED"
4942  013252  052737  000002  001664          BIS     #MEMSZB,OPTFLG  ;SET FLAG
4943  013260  000467                  1$:     BR      5$
4944
4945  013262  012737  013270  001110  2$:     MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
4946
4947  013270                          3$:
4948  013270  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4949  013272  104003                          ERROR   3               ;BAD INIT ERROR
4950  013274  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA, PATTERN 11
4951  013300  010011                          10011                   ;MEM MANAGEMENT ON DESTINATION
4952  013302  003777                          3777                    ;RELOCATION ARGUMENT
4953  013304  000400                          400                     ;400 WORDS
4954
4955  013306  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4956  013312  000523                          WRDATA!BA16             ;WRITE DATA AND SET BA16
4957  013314  177400                          -400                    ;400 WORDS
4958  013316  177770                          177770                  ;ACCROSS 64K BOUNDARY
4959  013320     020                          .BYTE   20              ;SECTOR 20
4960  013321     000                          .BYTE   0               ;TRACK 0
4961  013322  000312                          312                     ;CYLINDER 312
4962
4963  013324  104417                          TLOADRK                 ;LOAD RK REGS
4964  013326  104430                          TWAT96                  ;WAIT FOR INTERRUPT
4965  013330  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
                                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
                                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
013334  104004
013336  004437  041740                  JSR     R4,GENCOM       ;CLEAR IBUFF TO 1'S
013342  002007                          2007
013344  001000                          1000

013346  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
013352  000121                          RDDATA                  ;RDDATA
013354  177400                          -400                    ;-400 WORDS
013356  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
013360     020                          .BYTE   20              ;SECTOR 20
013361     000                          .BYTE   0               ;TRACK 0
013362  000312                          312                     ;CYLINDER 312
013364  104417                          TLOADRK                 ;LOAD RK REGS
013366  104424                          TWAT32                  ;WAIT FOR INTERRUPT
013370  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

013372  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
013374  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
013376  004437  041740                  JSR     R4,GENCOM       ;CHECK DATA
013402  110000                          110000                  ;MEMORY MANAGEMENT WITH DESTINATION
013404  003777                          3777                    ;RELOCATION ARGUMENT
013406  000400                          400                     ;400 WORDS
013410  000413                          BR      5$              ;NO MISCOMPARES-SKIP
013412  104015                          ERROR   15              ;REPORT FIRST ERROR

013414  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
013420  005300                  64$:    DEC     R0              ;DECREMENT COUNT
013422  001406                          BEQ     65$             ;IF ZERO - EXIT
013424  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
013430  050000                          50000
013432  000402                          BR      65$             ;NO MORE ERRORS - EXIT
013434  104016                          ERROR   16              ;REPORT NEXT ERROR
013436  000770                          BR      64$             ;LOOP
013440                          65$:
013440                          5$:
                                ;;*****************************************************************
                                ;*TEST 35       READ DATA ADDRESS GREATER THAN 64K
                                ;*
                                ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
                                ;*      CHECK MEMORY FOR CORRECT TRANSFER.
                                ;*
                                ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
                                ;*              OF MEMORY IS PRESENT.
                                ;*
                                ;;*****************************************************************
013440  000004                  T35:    SCOPE
013442  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
013450  123727  001326  000002          CMPB    $MAMS1,#2       ;CHECK IF >64K MEMORY
013456  002001                          BGE     2$              ;YES-SKIP
013460  000462                  1$:     BR      5$              ;EXIT

013462  012737  000032  001110  2$:     MOV     #32,$LPERR      ;SET LOCAL LOOP ON ERROR

013470                          3$:
```

```
5022  013470  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
5023  013472  104003                  ERROR   3               ;BAD INIT ERROR
5024  013474  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5025  013500  000123                  WRDATA                  ;WRDATA
5026  013502  177400                  -400                    ;-400 WORDS
5027  013504  063414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5028  013506     021                  .BYTE   21              ;SECTOR 21
5029  013507     000                  .BYTE   0               ;TRACK 0
5030  013510  000312                  312                     ;CYLINDER 312
5031  013512  004437  041740          JSR     R4,GENCOM       ;GENERATE DATA
5032  013516  000012                  12                      ;PATTERN 12
5033  013520  000400                  400                     ;400 WORDS
5034
5035  013522  104417                  TLOADRK                 ;LOAD RK REGS
5036  013524  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5037  013526  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5038
5039  013530  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5040  013532  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5041  013534  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5042  013540  000521                  RDDATA!BA16             ;READ DATA AND SET BA16
5043  013542  177400                  -400                    ;400 WORDS
5044  013544  177770                  177770                  ;ACCROSS 64K BOUNDARY
5045  013546     021                  .BYTE   21              ;FROM SECTOR 21
5046  013547     000                  .BYTE   0               ;TRACK 0
5047  013550  000312                  312                     ;CYLINDER 312
5048
5049  013552  104417                  TLOADRK                 ;LOAD RK REGS
5050  013554  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5051  013556  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5052
5053  013560  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5054  013562  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5055  013564  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
5056  013570  120000                  120000                  ;MEM MANAGEMENT WITH SOURCE
5057  013572  003777                  3777                    ;RELOCATION ARGUMENT
5058  013574  000400                  400                     ;400 WORDS
5059  013576  000413                  BR      5$              ;NO MISCOMPARES-SKIP
5060  013600  104015                  ERROR   15              ;REPORT FIRST ERROR
5061
5062  013602  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5063  013606  005300          64$:    DEC     R0              ;DECREMENT COUNT
5064  013610  001406                  BEQ     65$             ;IF ZERO - EXIT
5065  013612  004437  041740          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5066  013616  060000                  60000
5067  013620  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5068  013622  104016                  ERROR   16              ;REPORT NEXT ERROR
5069  013624  000770                  BR      64$             ;LOOP
5070  013626                  65$:
5071
5072  013626                  5$:
5073                          ;;*****************************************************************
5074                          ;*TEST 36       WRITE DATA ADDRESS GREATER THAN 96K
5075                          ;*
5076                          ;*      ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.
5077                          ;*      MAKE SURE CORRECT DATA IS ON DISK.
```

```
5078                                        ;*
5079                                        ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
5080                                        ;*              OF MEMORY IS PRESENT.
5081                                        ;*
5082                                        ;**************************************************************************
5083  013626  000004                        TST36:  SCOPE
5084  013630  012737  000012  001262                MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5085  013636  123727  001326  000003                CMPB    $MAMS1,#3       ;CHECK IF >96K MEMORY
5086  013644  002016                                BGE     3$              ;YES-SKIP
5087  013646  032737  000002  001664  1$:           BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
5088  013654  001011                                BNE     2$              ;NO-SKIP
5089
5090  013656  104401  051471                        TYPE    ,OPR011         ;"INSUFFICIENT MEMORY-DATA TRANSFET WITH
5091  013662  104401  051644                        TYPE    ,OPR014         ;ADDRESS >96K BYPASSED"
5092  013666  104401  051650                        TYPE    ,OPR015
5093  013672  052737  000002  001664                BIS     #MEMSZB,OPTFLG  ;SET REPORT FLAG
5094  013700  000463                        2$:     BR      6$
5095
5096  013702  012737  013710  001110  3$:           MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR
5097
5098  013710                                4$:
5099  013710  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
5100  013712  104003                                ERROR   3               ;BAD INIT ERROR
5101  013714  004437  035530                        JSR     R4,LRLOAD       ;LOAD "L" REG
5102  013720  001123                                WRDATA!BA17             ;WRITE DATA AND BA17
5103  013722  177400                                -400                    ;400 WORDS FROM
5104  013724  177770                                177770                  ;ACCROSS 96K BOUNDARY
5105  013726     022                                .BYTE   22              ;TO SECTOR 22
5106  013727     000                                .BYTE   0               ;TRACK 0
5107  013730  000312                                312                     ;CYL 312
5108  013732  004437  041740                        JSR     R4,GENCOM       ;GENERATE DATA
5109  013736  010013                                10013                   ;PATTERN 13 MEM MAN WITH DEST.
5110  013740  005777                                5777                    ;RELOCATION ARGUMENT
5111  013742  000400                                400                     ;400 WORDS
5112
5113  013744  104417                                TLOADRK                 ;LOAD RK REGS
5114  013746  104430                                TWAT96                  ;WAIT FOR INTERRUPT
5115  013750  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5116
5117  013752  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5118  013754  104004                                ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5119
5120  013756  004437  035530                        JSR     R4,LRLOAD       ;LOAD "L" REGS
5121  013762  000121                                RDDATA                  ;RDDATA
5122  013764  177400                                -400                    ;-400 WORDS
5123  013766  061414                                IBUFF                   ;IBUFF IS BUFF ADDRESS
5124  013770     022                                .BYTE   22              ;SECTOR 22
5125  013771     000                                .BYTE   0               ;TRACK 0
5126  013772  000312                                312                     ;CYLINDER 312
5127
5128  013774  104417                                TLOADRK                 ;LOAD RK REGS
5129  013776  104424                                TWAT32                  ;WAIT FOR INTERRUPT
5130  014000  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5131
5132  014002  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5133  014004  104004                                ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
```

```
5134
5135  014006  004437  041740                  JSR     R4,GENCOM       ;COMPARE DATA
5136  014012  110000                          110000                  ;MEM MANAGEMENT WITH DESTINATION
5137  014014  005777                          5777                    ;RELOCATION ARGUMENT
5138  014016  000400                          400                     ;400 WORDS
5139  014020  000413                          BR      6$              ;NO MISCOMPARES-BRANCH
5140  014022  104015                          ERROR   15              ;REPORT 1ST ERROR
5141
5142  014024  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5143  014030  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5144  014032  001406                          BEQ     65$             ;IF ZERO - EXIT
5145  014034  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5146  014040  050000                          50000
5147  014042  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5148  014044  104016                          ERROR   16              ;REPORT NEXT ERROR
5149  014046  000770                          BR      64$             ;LOOP
5150  014050                          65$:
5151
5152  014050                          6$:
5153                                  ;;*****************************************************************
5154                                  ;*TEST 37       READ DATA ADDRESS GREATER THAN 96K
5155                                  ;*
5156                                  ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
5157                                  ;*      CHECK MEMORY FOR CORRECT TRANSFER.
5158                                  ;*
5159                                  ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
5160                                  ;*              OF MEMORY IS PRESENT.
5161                                  ;*
5162                                  ;;*****************************************************************
5163  014050  000004                  TST37:  SCOPE
5164  014052  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5165  014060  123727  001326  000003          CMPB    $MAMS1,#3       ;CHECK IF >96K MEMORY
5166  014066  002001                          BGE     3$              ;YES-SKIP
5167  014070  000462                  2$:     BR      6$
5168
5169  014072  012737  014100  001110  3$:     MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR
5170
5171  014100                          4$:
5172  014100  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5173  014102  104003                          ERROR   3               ;BAD INIT ERROR
5174
5175  014104  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5176  014110  000123                          WRDATA                  ;WRDATA
5177  014112  177400                          -400                    ;-400 WORDS
5178  014114  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5179  014116      005                         .BYTE   5               ;SECTOR 5
5180  014117      000                         .BYTE   0               ;TRACK 0
5181  014120  000312                          312                     ;CYLINDER 312
5182  014122  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
5183  014126  000014                          14                      ;PATTERN 14
5184  014130  000400                          400                     ;400 WORDS
5185
5186  014132  104417                          TLOADRK                 ;LOAD RK REGS
5187  014134  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5188  014136  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5189
```

```
5190  014140  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5191  014142  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5192  014144  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5193  014150  001121                  RDDATA!BA17             ;READ DATA WITH BA17 SET
5194  014152  177400                  -400                    ;400 WORDS
5195  014154  177770                  177770                  ;ACCROSS 96K BOUNDARY
5196  014156     005                  .BYTE   5               ;FROM SECTOR 5
5197  014157     000                  .BYTE   0               ;TRACK 0
5198  014160  000312                  312                     ;CYL 312
5199
5200  014162  104417                  TLOADRK                 ;LOAD RK REGS
5201  014164  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5202  014166  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5203
5204  014170  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5205  014172  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5206  014174  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
5207  014200  120000                  120000                  ;MEM MANAGEMENT WITH SOURCE
5208  014202  005777                  5777                    ;RELOCATION ARGUMENT
5209  014204  000400                  400                     ;400 WORDS
5210  014206  000413                  BR      6$              ;NO MISCOMPARES-SKIP
5211  014210  104015                  ERROR   15              ;REPORT FIRST ERROR
5212
5213  014212  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5214  014216  005300          64$:    DEC     R0              ;DECREMENT COUNT
5215  014220  001406                  BEQ     65$             ;IF ZERO - EXIT
5216  014222  004437  041740          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5217  014226  060000                  60000
5218  014230  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5219  014232  104016                  ERROR   16              ;REPORT NEXT ERROR
5220  014234  000770                  BR      64$             ;LOOP
5221  014236                  65$:
5222
5223  014236                  6$:
5224                                  ;;*****************************************************************
5225                                  ;*TEST 40       PARTIAL SECTOR WRITE DATA
5226                                  ;*
5227                                  ;*      ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312,
5228                                  ;*      HEAD 0, SECTOR 0.  MAKE SURE THE SECTOR WAS
5229                                  ;*      ZERO FILLED CORRECTLY.
5230                                  ;*
5231                                  ;;*****************************************************************
5232  014236  000004          TST40:  SCOPE
5233  014240  012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5234  014246  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
5235  014250  104003                  ERROR   3               ;BAD INIT ERROR
5236
5237  014252  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REG
5238  014256  000123                  WRDATA                  ;WRITE DATA
5239  014260  177675                  -103                    ;WORD COUNT PARTIAL SECTOR
5240  014262  063414                  OBUFF                   ;BUFF ADDRESS
5241  014264     007                  .BYTE   7               ;SECTOR 7
5242  014265     000                  .BYTE   0               ;TRACK 0
5243  014266  000312                  312                     ;CYLINDER 312
5244
5245  014270  004437  041740          JSR     R4,GENCOM       ;GENERATE DATA
```

```
5246  014274  000003                  3                       ;PATTERN 3
5247  014276  000400                  400                     ;400 WORDS
5248
5249  014300  104417                  TLOADRK                 ;LOAD RK REGS
5250  014302  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5251  014304  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5252
5253  014306  104421                  TCHKOP                  ;CHECK FOR ANY ERROR
5254  014310  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERROR
5255
5256  014312  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5257
5258  014316  000121                  RDDATA                  ;READ DATA
5259  014320  177400                  -400                    ;ONE FULL SECTOR
5260  014322  061414                  IBUFF                   ;BUFF ADDRESS
5261  014324     007                  .BYTE   7               ;SECTOR 7
5262  014325     000                  .BYTE   0               ;TRACK 0
5263  014326  000312                  312                     ;CYLINDER 312
5264
5265  014330  004437  041740          JSR     R4,GENCOM
5266  014334  002007                  2007                    ;CLEAR IBUFF TO ALL ONES
5267  014336  000400                  400
5268
5269  014340  104417                  TLOADRK                 ;LOAD RK REGS
5270  014342  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5271  014344  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE
5272
5273  014346  104421                  TCHKOP                  ;CHECK FOR ANY ERRORS
5274  014350  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
5275
5276  014352  012701  063622          MOV     #OBUFF+206,R1   ;CLEAR THE LAST 205 WORDS
5277  014356  012700  000275          MOV     #275,R0         ;OF THE OUTPUT BUFFER TO ZERO
5278  014362  005021          1$:     CLR     (R1)+           ;TO VERIFY THE PARTIAL SECTOR
5279  014364  005300                  DEC     R0              ;WRITE 0 FILLED THE SECTOR
5280  014366  001375                  BNE     1$
5281  014370  004437  041740          JSR     R4,GENCOM
5282  014374  100000                  100000                  ;COMPARE OBUFF & IBUFF.
5283  014376  000400                  400                     ;ALL 400 WORDS
5284  014400  000413                  BR      3$              ;NO ERRORS-EXIT
5285  014402  104015                  ERROR   15              ;REPORT FIRST COMPARE ERROR
5286
5287  014404  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5288  014410  005300          2$:     DEC     R0              ;DECREMENT IT
5289  014412  001406                  BEQ     3$              ;IF ZERO-EXIT
5290  014414  004437  041740          JSR     R4,GENCOM
5291  014420  040000                  40000                   ;CONTINUE COMPARE
5292  014422  000402                  BR      3$              ;NO MORE ERRORS-EXIT
5293  014424  104016                  ERROR   16              ;REPORT NEXT COMPARE ERROR
5294  014426  000770                  BR      2$              ;LOOP
5295
5296  014430                  3$:
5297                          ;;*****************************************************************
5298                          ;*TEST 41        PARTIAL SECTOR READ DATA
5299                          ;*
5300                          ;*      WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A
5301                          ;*      KNOWN CONFIGURATION.  ISSUE A READ DATA OF
```

```
5302                                    ;*      103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0.
5303                                    ;*      MAKE SURE ONLY 103 WORDS GET TRANSFERRED
5304                                    ;*      TO MEMORY.
5305                                    ;*
5306                                    ;;**********************************************************************
5307  014430  000004                    TST41:  SCOPE
5308  014432  012737  000012  001262            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5309  014440  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
5310  014442  104003                            ERROR   3               ;BAD INIT ERROR
5311
5312  014444  004437  035530                    JSR     R4,LRLOAD       ;LOAD "L" REGS
5313  014450  000123                            WRDATA                  ;WRDATA
5314  014452  177400                            -400                    ;-400 WORDS
5315  014454  063414                            OBUFF                   ;OBUFF IS BUFF ADDRESS
5316  014456     017                            .BYTE   17              ;SECTOR 17
5317  014457     000                            .BYTE   0               ;TRACK 0
5318  014460  000312                            312                     ;CYLINDER 312
5319  014462  004437  041740                    JSR     R4,GENCOM       ;GENERATE DATA
5320  014466  000004                            4                       ;PATTERN 4
5321  014470  000400                            400                     ;400 WORDS
5322
5323  014472  104417                            TLOADRK                 ;LOAD RK REGS
5324  014474  104430                            TWAT96                  ;WAIT FOR INTERRUPT
5325  014476  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5326
5327  014500  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5328  014502  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5329
5330  014504  004437  035530                    JSR     R4,LRLOAD       ;LOAD "L" REGS
5331  014510  000121                            RDDATA                  ;RDDATA
5332  014512  177675                            -103                    ;-103 WORDS
5333  014514  061414                            IBUFF                   ;IBUFF IS BUFF ADDRESS
5334  014516     017                            .BYTE   17              ;SECTOR 17
5335  014517     000                            .BYTE   0               ;TRACK 0
5336  014520  000312                            312                     ;CYLINDER 312
5337  014522  004437  041740                    JSR     R4,GENCOM
5338  014526  002007                            2007                    ;CLEAR IBUFF
5339  014530  000400                            400
5340
5341  014532  104417                            TLOADRK                 ;LOAD RK REGS
5342  014534  104424                            TWAT32                  ;WAIT FOR INTERRUPT
5343  014536  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5344  014540  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5345  014542  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5346
5347  014544  012700  063622                    MOV     #OBUFF+206,R0   ;AFTER THE LAST 205 WORDS OF
5348  014550  012701  000275                    MOV     #275,R1         ;THE OUTPUT BUFFER TO ALL ONES.
5349  014554  012720  177777          1$:       MOV     #-1,(R0)+       ;THESE SHOULD ALL BE ONES IN
5350  014560  005301                            DEC     R1              ;IBUFF BECAUSE THE PARTIAL
5351  014562  001374                            BNE     1$              ;READ FILLED ONLY 103 WORDS.
5352  014564  004437  041740                    JSR     R4,GENCOM       ;GO COMPARE IBUFF & OBUFF
5353  014570  100000                            100000
5354  014572  000400                            400                     ;ALL 400 WORDS
5355  014574  000413                            BR      3$              ;NO ERRORS-EXIT
5356  014576  104015                            ERROR   15              ;REPORT FIRST COMPARE ERROR
5357
```

```
5358  014600  013700  001634              MOV     ERRLMT,R0       ;GET ERROR LIMIT
5359  014604  005300              64$:    DEC     R0              ;DECREMENT COUNT
5360  014606  001406                      BEQ     65$             ;IF ZERO - EXIT
5361  014610  004437  041740              JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5362  014614  040000                      40000
5363  014616  000402                      BR      65$             ;NO MORE ERRORS - EXIT
5364  014620  104016                      ERROR   16              ;REPORT NEXT ERROR
5365  014622  000770                      BR      64$             ;LOOP
5366  014624                      65$:
5367
5368  014624                      3$:
5369                              ;;*****************************************************************
5370                              ;*TEST 42       WRITE DATA WITH NON-EXISTENT MEMORY
5371                              ;*
5372                              ;*      ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000.
5373                              ;*      MAKE SURE NON-EXISTENT MEMORY SETS.
5374                              ;*
5375                              ;;*****************************************************************
5376  014624  000004              TST42:  SCOPE
5377  014626  012737  000012  001262      MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5378  014634  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
5379  014636  104003                      ERROR   3               ;BAD INIT ERROR
5380
5381  014640  004437  035530              JSR     R4,LRLOAD       ;LOAD "L" REG
5382  014644  001523                      BA16!BA17!WRDATA        ;BA16 & 17 SET WITH WRITE DATA
5383  014646  177777                      -1                      ;WORD COUNT OF 1
5384  014650  176000                      176000                  ;BUFF ADDRESS=IO PAGE BASE
5385  014652     013                      .BYTE   13              ;SECT 13
5386  014653     000                      .BYTE   0               ;TRACK 0
5387  014654  000312                      312                     ;CYLINDER 312
5388
5389  014656  104417                      TLOADRK                 ;LOAD RK REGS
5390  014660  104430                      TWAT96                  ;WAIT FOR INTERRUPT
5391  014662  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5392  014664  104422                      TCHKWE                  ;CHECK OPERATION WITH ERROR
5393  014666  000000                      0
5394  014670  000000                      0
5395  014672  000040                      NEMERR                  ;NON-EXISTENT MEMORY ERROR
5396  014674  104004                      ERROR   4 ;OR5,6,7      ;REPORT ANY DISCREPENCIES
5397  014676  012737  054047  001450      MOV     #EM11A,EM1ON    ;SET UP ERROR MESSAGE
5398  014704  012737  050240  060770      MOV     #OPER42,DF01OA  ;WITH SUPPORT MESSAGE
5399  014712  113700  001541              MOVB    T.CS1+1,R0      ;GET UPPER CS1
5400  014716  042700  177774              BIC     #177774,R0      ;CLEAR UNUSED BITS
5401  014722  022700  000003              CMP     #3,R0           ;TEST IF BOTH UPPER BUS BITS SET
5402  014726  001406                      BEQ     1$              ;YES - SKIP
5403  014730  010037  001204              MOV     R0,$REG11       ;SET UP FOR ERROR REPORT
5404  014734  012737  000003  001202      MOV     #3,$REG10
5405  014742  104010                      ERROR   10
5406  014744  022737  176002  001544  1$: CMP     #176002,T.BA    ;TEST IF BUSS ADDRESS LOW OKAY
5407  014752  001412                      BEQ     2$              ;YES - SKIP
5408  014754  012737  054021  001450      MOV     #EM11,EM1ON     ;SET UP MESSGAE
5409  014762  012737  176002  001202      MOV     #176002,$REG10  ;STORE VALUE FOR REPORT
5410  014770  013737  001544  001204      MOV     T.BA,$REG11
5411  014776  104010                      ERROR   10
5412
5413  015000  005737  001542          2$: TST     T.WC            ;TEST IF WORD COUNT CORRECT
```

```
                        T42     WRITE DATA WITH NON-EXISTENT MEMORY

015004  001411                          BEQ     3$              ;YES - SKIP
015006  012737  053774  001450          MOV     #EM10,EMION     ;SET UP MESSAGE
015014  005037  001202                  CLR     $REG10
015020  013737  001542  001204          MOV     T.WC,$REG11
015026  104010                          ERROR 10

015030                          3$:

                        ;;*****************************************************************
                        ;*TEST 43       READ DATA WITH NON-EXISTENT MEMORY
                        ;*
                        ;*      ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000.
                        ;*      MAKE SURE NON-EXISTENT MEMORY SETS.
                        ;*
                        ;*****************************************************************
015030  000004         TST43:  SCOPE
015032  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
015040  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
015042  104003                          ERROR   3               ;BAD INIT ERROR

015044  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REG
015050  001521                          BA16!BA17!RDDATA        ;BA16 & 17 WITH READ DATA
015052  177777                          -1                      ;WORD COUNT OF 1
015054  176000                          176000                  ;BUFF ADDRESS=IO PAGE BASE
015056     013                          .BYTE   13              ;SECTOR 13
015057     000                          .BYTE   0               ;TRACK 0
015060  000312                          312                     ;CYL 312

015062  104417                          TLOADRK                 ;LOAD RK REGS
015064  104430                          TWAT96                  ;WAIT FOR INTERRUPT
015066  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
015070  104422                          TCHKWE                  ;CHECK OPERATION WITH ERRORS
015072  000000                          0
015074  000000                          0
015076  000040                          NEMERR                  ;NON-EXISTENT MEMORY ERROR
015100  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
015102  012737  054047  001450          MOV     #EM11A,EMION    ;SET MESSAGE
015110  012737  050314  060770          MOV     #OPER43,DFO10A  ;SET SUPPORT MESSAGE
015116  113700  001541                  MOVB    T.CS1+1,R0              ;GET UPPER CS1
015122  042700  177774                  BIC     #177774,R0      ;CLEAR UNWANTED BITS
015126  022700  000003                  CMP     #3,R0           ;TEST BOTH BUS 16 & 17 SET
015132  001406                          BEQ     1$              ;YES - SKIP
015134  012737  000003  001202          MOV     #3,$REG10       ;SET VALUES FOR REPORT
015142  010037  001204                  MOV     R0,$REG11
015146  104010                          ERROR   10

015150  022737  176002  001544  1$:     CMP     #176002,T.BA    ;TEST IF BUS ADDRESS CORRECT
015156  001412                          BEQ     2$              ;YES - SKIP
015160  012737  054021  001450          MOV     #EM11,EMION     ;SET MESSAGE
015166  012737  176002  001202          MOV     #176002,$REG10  ;SET VALUES FOR REPORT
015174  013737  001544  001204          MOV     T.BA,$REG11
015202  104010                          ERROR   10

015204  005737  001542          2$:     TST     T.WC            ;TEST IF WORD COUNT CORRECT
015210  001411                          BEQ     3$              ;YES - SKIP
015212  012737  053774  001450          MOV     #EM10,EMION     ;SET MESSAGE
```

```
5470  015220  005037  001202                CLR     $REG10          ;SET VALUES
5471  015224  013737  001542  001204        MOV     T.WC,$REG11
5472  015232  104010                        ERROR   10
5473
5474  015234                        3$:
5475
5476                                ;;*********************************************************************
5477                                ;*TEST 44       UNIBUS PARITY ERROR
5478                                ;*
5479                                ;*      INITIALIZE A MEMORY LOCATION WITH BAD
5480                                ;*      PARITY. ISSUE A WRITE DATA OF 400 WORDS
5481                                ;*      STARTING AT A LOCATION 112 WORDS BEFORE
5482                                ;*      THE LOCATION WITH BAD PARITY.  MAKE SURE
5483                                ;*      THAT UNIBUS PARITY ERROR SETS.
5484                                ;*
5485                                ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF
5486                                ;*              MEMORY PARITY OPTION EXISTS FOR
5487                                ;*              BUFFER.
5488                                ;*
5489                                ;*********************************************************************
5490  015234  000004                TST44:  SCOPE
5491  015236  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5492  015244  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
5493  015246  104003                        ERROR   3               ;BAD INIT ERROR
5494
5495  015250  032737  000100  001664        BIT     #PARPRE,OPTFLG  ;TEST IF PARITY OPTION PRESENT
5496  015256  001013                        BNE     1$              ;YES-SKIP
5497  015260  032737  000004  001664        BIT     #MEMPYB,OPTFLG  ;TEST IF PARITY OPTION REPORTED
5498  015266  001005                        BNE     25$             ;YES-SKIP TO EXIT
5499  015270  104401  051274                TYPE    ,OPR010         ;PRINT BYPASS MESSAGE
5500  015274  052737  000004  001664        BIS     #MEMPYB,OPTFLG  ;SET OPTION REPORTED BIT
5501  015302  000137  016050        25$:    JMP     2$              ;SKIP TO EXIT
5502
5503  015306                        1$:
5504  015306  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
5505  015312  000123                        WRDATA                  ;WRDATA
5506  015314  177400                        -400                    ;-400 WORDS
5507  015316  063414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
5508  015320     010                        .BYTE   10              ;SECTOR 10
5509  015321     000                        .BYTE   0               ;TRACK 0
5510  015322  ( 1312                        312                     ;CYLINDER 312
5511
5512  015324  032737  002000  001664        BIT     #CP1170,OPTFLG  ;TEST IF 11/70
5513  015332  001002                        BNE     3$              ;YES - SKIP
5514  015334  005077  164332                CLR     @CSRPTR         ;CLEAR PARITY IE
5515
5516  015340  004437  041740        3$:     JSR     R4,GENCOM       ;GENERATE DATA
5517  015344  000007                        7                       ;PATTERN 7
5518  015346  000400                        400                     ;400 WORDS
5519
5520  015350  012746  000340                MOV     #PR7,-(SP)      ;PUT PRIORITY 7 ON STACK
5521  015354  012746  015362                MOV     #10$,-(SP)      ;PUT ADDRESS ON STACK
5522  015360  000002                        RTI                     ;SET PRI
5523  015362                        10$:
5524
5525  015362  012737  100200  063640        MOV     #100200,OBUFF+224 ;SET WORD IN BUFFER
```

```
5526  015370  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
5527  015376  001011                          BNE     5$              ;YES - SKIP
5528  015400  012777  000004  164264          MOV     #BIT2,@CSRPTR   ;SET WRITE WRONG PARITY BIT
5529  015406  012737  100200  063640          MOV     #100200,OBUFF+224  ;SET BAD PARITY IN MEMORY
5530  015414  012777  000001  164250          MOV     #BIT0,@CSRPTR   ;CLEAR CONTROL BIT, SET IE BIT
5531
5532  015422  013746  001622          5$:     MOV     RKPRI,-(SP)     ;SET OLD PRIORITY
5533  015426  012746  015434                  MOV     #11$,-(SP)      ;SET ADDRESS
5534  015432  000002                          RTI                     ;RESTORE PRI
5535  015434  013704  001672          11$:    MOV     CSRPTR,R4       ;SET R4 WITH CSR POINTER
5536  015440  005000                          CLR     R0              ;SET R0 FOR COUNTER
5537  015442  012701  001662                  MOV     #INTSET,R1      ;SET R1 FOR POINTER TO INTERRUPT FLAG
5538  015446  012777  061400  164144          MOV     #SPCHLR,@RKVEC  ;SET UP INTERRUPT VECTORS FOR
5539  015454  012777  061410  164234          MOV     #SPCPAR,@MMVECA ; RK611 AND PARITY ERROR
5540  015462  012737  016050  001264          MOV     #2$,$ESCAPE     ;SET UP ESCAPE FOR ERROR
5541  015470  104417                          TLOADRK                 ;LOAD RK REGS
5542  015472  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
5543  015500  001434                          BEQ     45$             ;NO - SKIP
5544  015502  012737  000016  177746          MOV     #16,177746      ;SET TO DISABLE CACHE AND UNIBUS ERROR
5545  015510  012777  000000  164172          MOV     #0,@KWLADD      ;TURN OFF CLOCK INTERRUPTS
5546  015516  012777  000000  163420          MOV     #0,@STKS        ;TURN OFF KEYBOARD
5547  015524  012737  170000  177750          MOV     #170000,177750  ;SET FOR ERROR FORCE
5548
5549                                  ;**********************
5550  015532  105711                  40$:    TSTB    (R1)            ;LOOP TO WAIT FOR INTERRUPT OR ABORT
5551  015534  003005                          BGT     43$             ;WAIT. THE CODE BETWEEN THE STARS IS SET
5552  015536  005300                          DEC     R0              ;SET UP SO ALL BYTES HAVE PARITY OF 1.
5553  015540  100774                          BMI     40$             ;IF THIS CODE IS CHANGED, REMEMBER ALL
5554  015542  000240                          NOP                     ;BYTES MUST HAVE AN EVEN NUMBER OF
5555  015544  003372                          BGT     40$             ;BITS.
5556  015546  000240                          NOP
5557  015550  005014                  43$:    CLR     (R4)            ;CLEAR ERROR FORCE
5558                                  ;**************************
5559
5560  015552  005037  177746                  CLR     177746          ;ENABLE CACHE
5561  015556  012777  000100  164124          MOV     #BIT6,@KWLADD   ;TURN ON CLOCK INTERRUPTS
5562  015564  012777  000100  163352          MOV     #100,@STKS      ;TURN ON KEYBOARD INTERRUPTS
5563  015572  104430                  45$:    TWAT96                  ;WAIT FOR INTERRUPT
5564  015574  000414                          BR      46$             ;TO SLOW/NOT COMPLETE ERROR
5565  015576  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
5566  015604  001024                          BNE     48$             ;YES - SKIP
5567  015606  005077  164060                  CLR     @CSRPTR         ;ELSE CLEAR CSR
5568  015612  005037  063640                  CLR     OBUFF+224       ;CLEAR THE BAD PARITY WORD
5569  015616  012777  000001  164046          MOV     #1,@CSRPTR      ;SET PARITY DETECT AGAIN
5570  015624  000414                          BR      48$             ;SKIP
5571
5572  015626  032737  002000  001664  46$:    BIT     #CP1170,OPTFLG  ;TEST IF 11/70
5573  015634  001007                          BNE     47$             ;YES - SKIP
5574  015636  005077  164030                  CLR     @CSRPTR         ;CLEAR CSR
5575  015642  005037  063640                  CLR     OBUFF+224       ;CLEAR BAD PARITY WORD
5576  015646  012777  000001  164016          MOV     #1,@CSRPTR      ;SET UP PARITY DETECT AGAIN
5577  015654  104002                  47$:    ERROR   2               ;REPORT TO SLOW ERROR
5578  015656                          48$:
5579  015656  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
5580  015660  000000                          0
5581  015662  000000                          0
```

```
5582  015664  000100                          UPERR                           ;UNIBUS PARITY ERROR
5583  015666  104004                          ERROR   4; OR 5,6,7             ;REPORT ALL DISCREPANCIES
5584
5585  015670  005037  001264                  CLR     $ESCAPE                 ;CLEAR ESCAPE
5586  015674  012737  050367  060770          MOV     #OPER44,DF01DA          ;SET MESSAGES FOR REPORT
5587  015702  012737  054021  001450          MOV     #EM11,EM1ON
5588  015710  023727  001544  063642          CMP     T.BA,#OBUFF+226         ;CHECK IF BA IN RANGE
5589  015716  103010                          BHIS    14$                     ;NOT TO LOW - SKIP
5590  015720  012737  063642  001202          MOV     #OBUFF+226,$REG10 ;SET VALUES FOR REPORT
5591  015726  013737  001544  001204          MOV     T.BA,$REG11
5592  015734  104010                          ERROR   10
5593  015736  000413                          BR      16$
5594
5595  015740  023727  001544  063646  14$:    CMP     T.BA,#OBUFF+232 ;CHECK IF BA IN RANGE
5596  015746  101407                          BLOS    16$                     ;YES - SKIP
5597  015750  012737  063646  001202          MOV     #OBUFF+232,$REG10 ;SET VALUES
5598  015756  012737  001544  001204          MOV     #T.BA,$REG11
5599  015764  104010                          ERROR   10
5600
5601  015766  012737  053774  001450  16$:    MOV     #EM10,EM1ON             ;SET MESSAGE
5602  015774  023727  001542  177513          CMP     T.WC,#-265              ;CHECK IF WORD COUNT WITHIN RANGE
5603  016002  103007                          BHIS    20$                     ;YES - SKIP
5604  016004  012737  177513  001202          MOV     #-265,$REG10    ;SET VALUES
5605  016012  013737  001542  001204          MOV     T.WC,$REG11
5606  016020  104010                          ERROR   10
5607
5608  016022  023727  001542  177515  20$:    CMP     T.WC,#-263              ;STILL CHECKING IF WC IN RANGE
5609  016030  101407                          BLOS    2$                      ;YES - SK
5610  016032  012737  177515  001202          MOV     #-263,$REG10    ;SET VALUES
5611  016040  013737  001542  001204          MOV     T.WC,$REG11
5612  016046  104010                          ERROR   10
5613
5614  016050  012777  034354  163542  2$:     MOV     #INTHLR,@RKVEC  ;RESET INT. VECTORS FOR RK06
5615  016056  012777  034364  163632          MOV     #PERHLR,@MMVECA ;    AND PARITY ERRORS
5616  016064  012777  000100  163616          MOV     #BIT6,@KWLADD   ;INSURE CLOCK IS ENABLED
5617  016072  012777  000100  163044          MOV     #100,@STKS              ;INSURE KEYBOARD IS ENABLED
5618
5619
5620                                  .SBTTL  **MULTIPLE SECTOR OPERATIONS
5621
5622                                  ;;*****************************************************************
5623                                  ;*TEST 45       TWO SECTOR WRITE DATA (PART 1)
5624                                  ;*
5625                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5626                                  ;*      TRACK 0, SECTOR 0.  READ DATA BACK ONE SECTOR
5627                                  ;*      AT A TIME AND MAKE SURE IT IS CORRECT.
5628                                  ;*
5629                                  ;;*****************************************************************
5630  016100  000004                  TST45:  SCOPE
5631  016102  012737  000012  001262          MOV     #10.,$TIMES             ;;DO 10. ITERATIONS
5632  016110  104416                          TSSINIT                         ;CLEAR SUBSYSTEM
5633  016112  104003                          ERROR   3                       ;BAD INIT ERROR
5634
5635  016114  004437  035530                  JSR     R4,LRLOAD               ;LOAD "L" REGS
5636  016120  000123                          WRDATA                          ;WRDATA
5637  016122  177000                          -1000                           ;-1000 WORDS
```

```
5638  016124  063414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5639  016126     000                  .BYTE   0               ;SECTOR 0
5640  016127     000                  .BYTE   0               ;TRACK 0
5641  016130  000312                  312                     ;CYLINDER 312
5642
5643  016132  004437  041740          JSR     R4,GENCOM       ;GENERATE DATA
5644  016136  000015                  15                      ;PATTERN 15
5645  016140  001000                  1000                    ;1000 WORDS
5646
5647  016142  104417                  TLOADRK                 ;LOAD RK REGS
5648  016144  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5649  016146  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5650
5651  016150  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5652  016152  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5653
5654  016154  004437  041740          JSR     R4,GENCOM       ;CLEAR IBUFF
5655  016160  002007                  2007                    ;TO ALL 1'S
5656  016162  001000                  1000
5657
5658  016164  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5659  016170  000121                  RDDATA                  ;RDDATA
5660  016172  177400                  -400                    ;-400 WORDS
5661  016174  061414                  IBUFF                   ;IBUFF IS BUFF ADDRESS
5662  016176     000                  .BYTE   0               ;SECTOR 0
5663  016177     000                  .BYTE   0               ;TRACK 0
5664  016200  000312                  312                     ;CYLINDER 312
5665
5666  016202  104417                  TLOADRK                 ;LOAD RK REGS
5667  016204  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5668  016206  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5669
5670  016210  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5671  016212  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5672
5673  016214  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5674  016220  000121                  RDDATA                  ;RDDATA
5675  016222  177400                  -400                    ;-400 WORDS
5676  016224  062414                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5677  016226     001                  .BYTE   1               ;SECTOR 1
5678  016227     000                  .BYTE   0               ;TRACK 0
5679  016230  000312                  312                     ;CYLINDER 312
5680
5681  016232  104417                  TLOADRK                 ;LOAD RK REGS
5682  016234  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5683  016236  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5684
5685  016240  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5686  016242  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5687
5688  016244  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
5689  016250  100000                  100000                  ;
5690  016252  001000                  1000                    ;1000 WORDS
5691  016254  000413                  BR      2$              ;NO MISCOMPARES-EXIT
5692  016256  104015                  ERROR   15              ;REPORT FIRST ERROR
5693
```

```
5694  016260  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5695  016264  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5696  016266  001406                          BEQ     65$             ;IF ZERO - EXIT
5697  016270  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5698  016274  040000                          40000
5699  016276  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5700  016300  104016                          ERROR   16              ;REPORT NEXT ERROR
5701  016302  000770                          BR      64$             ;LOOP
5702  016304                          65$:
5703
5704  016304                          2$:
5705                                  ;*****************************************************************
5706                                  ;*TEST 46       TWO SECTOR WRITE DATA (PART 2)
5707                                  ;*
5708                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5709                                  ;*      TRACK 0, SECTOR 23.  READ DATA BACK ONE SECTOR
5710                                  ;*      AT A TIME AND MAKE SURE A MID-TRANSFER
5711                                  ;*      SEEK DID NOT TAKE PLACE.
5712                                  ;*
5713                                  ;*****************************************************************
5714  016304  000004                  TST46:  SCOPE
5715  016306  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5716  016314  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5717  016316  104003                          ERROR   3               ;BAD INIT ERROR
5718
5719  016320  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5720  016324  000123                          WRDATA                  ;WRDATA
5721  016326  177000                          -1000                   ;-1000 WORDS
5722  016330  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5723  016332     023                          .BYTE   23              ;SECTOR 23
5724  016333     000                          .BYTE   0               ;TRACK 0
5725  016334  000312                          312                     ;CYLINDER 312
5726
5727  016336  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
5728  016342  000016                          16                      ;PATTERN 16
5729  016344  001000                          1000                    ;1000 WORDS
5730
5731  016346  104417                          TLOADRK                 ;LOAD RK REGS
5732  016350  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5733  016352  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5734
5735  016354  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5736  016356  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5737
5738                                  ;       IF THE TRACK ADDRESS AT THE END OF OPERATION IS IN ERROR
5739                                  ;       THE CONTROLLER DID A MID-TRANSFER SEEK AS THOUGH IT
5740                                  ;       WERE IN 24(8) SECTORS PER TRACK MODE.
5741
5742  016360  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5743  016364  000121                          RDDATA                  ;RDDATA
5744  016366  177400                          -400                    ;-400 WORDS
5745  016370  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
5746  016372     023                          .BYTE   23              ;SECTOR 23
5747  016373     000                          .BYTE   0               ;TRACK 0
5748  016374  000312                          312                     ;CYLINDER 312
5749
```

```
5750  016376  004437  041740          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
5751  016402  002007                  2007
5752  016404  001000                  1000
5753
5754  016406  104417                  TLOADRK                 ;LOAD RK REGS
5755  016410  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5756  016412  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5757
5758  016414  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5759  016416  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5760
5761  016420  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5762  016424  000121                  RDDATA                  ;RDDATA
5763  016426  177400                  -400                    ;-400 WORDS
5764  016430  062414                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5765  016432     024                  .BYTE   24              ;SECTOR 24
5766  016433     000                  .BYTE   0               ;TRACK 0
5767  016434  000312                  312                     ;CYLINDER 312
5768
5769  016436  104417                  TLOADRK                 ;LOAD RK REGS
5770  016440  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5771  016442  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5772
5773  016444  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5774  016446  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5775
5776  016450  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
5777  016454  100000                  100000
5778  016456  001000                  1000                    ;1000 WORDS
5779  016460  000413                  BR      1$              ;NO ERRORS-SKIP
5780  016462  104015                  ERROR   15              ;REPORT FIRST ERROR
5781
5782  016464  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5783  016470  005300          64$:    DEC     R0              ;DECREMENT COUNT
5784  016472  001406                  BEQ     65$             ;IF ZERO - EXIT
5785  016474  004437  041740          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5786  016500  040000                  40000
5787  016502  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5788  016504  104016                  ERROR   16              ;REPORT NEXT ERROR
5789  016506  000770                  BR      64$             ;LOOP
5790  016510                  65$:
5791  016510                  1$:
5792                          ;;*****************************************************************
5793                          ;*TEST 47       TWO SECTOR WRITE DATA (PART 3)
5794                          ;*
5795                          ;*      ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312,
5796                          ;*      TRACK 0, SECTOR 10.  READ DATA BACK ONE SECTOR AT
5797                          ;*      A TIME AND CHECK ZERO FILL OF SECOND SECTOR.
5798                          ;*
5799                          ;;*****************************************************************
5800  016510  000004          TST47:  SCOPE
5801  016512  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5802  016520  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
5803  016522  104003                  ERROR   3               ;BAD INIT ERROR
5804
5805  016524  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
5806  016530  000123                  WRDATA                  ;WRDATA
5807  016532  177377                  -401                    ;-401 WORDS
5808  016534  063414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5809  016536     010                  .BYTE   10              ;SECTOR 10
5810  016537     000                  .BYTE   0               ;TRACK 0
5811  016540  000312                  312                     ;CYLINDER 312
5812
5813  016542  004437  041740          JSR     R4,GENCOM       ;GENERATE DATA
5814  016546  000002                  2                       ;PATTERN 2
5815  016550  001000                  1000                    ;1000 WORDS
5816
5817  016552  104417                  TLOADRK                 ;LOAD RK REGS
5818  016554  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5819  016556  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5820
5821  016560  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5822  016562  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5823                          ;       CLEAR LAST 377 WORDS OF OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
5824  016564  012700  064416          MOV     #OBUFF+1002,R0  ;GET STARTING ADDRESS TO BE CLEARED
5825  016570  012701  000377          MOV     #377,R1         ;NUMBER OF WORDS
5826  016574  005020          1$:     CLR     (R0)+           ;CLEAR WORD
5827  016576  005301                  DEC     R1              ;DEC COUNT
5828  016600  001375                  BNE     1$              ;LOOP UNTIL COUNT ZERO
5829  016602  004437  041740          JSR     R4,GENCOM       ;CLEAR IBUFF TO ONES
5830  016606  002007                  2007
5831  016610  001000                  1000
5832
5833  016612  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5834  016616  000121                  RDDATA                  ;RDDATA
5835  016620  177400                  -400                    ;-400 WORDS
5836  016622  061414                  IBUFF                   ;IBUFF IS BUFF ADDRESS
5837  016624     010                  .BYTE   10              ;SECTOR 10
5838  016625     000                  .BYTE   0               ;TRACK 0
5839  016626  000312                  312                     ;CYLINDER 312
5840
5841  016630  104417                  TLOADRK                 ;LOAD RK REGS
5842  016632  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5843  016634  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5844
5845  016636  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5846  016640  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5847  016642  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5848  016646  000121                  RDDATA                  ;RDDATA
5849  016650  177400                  -400                    ;-400 WORDS
5850  016652  062414                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5851  016654     011                  .BYTE   11              ;SECTOR 11
5852  016655     000                  .BYTE   0               ;TRACK 0
5853  016656  000312                  312                     ;CYLINDER 312
5854
5855  016660  104417                  TLOADRK                 ;LOAD RK REGS
5856  016662  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5857  016664  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5858
5859  016666  004437  041740          JSR     R4,GENCOM       ;DATA COMPARE
5860  016672  100000                  100000
5861  016674  001000                  1000                    ;1000 WORDS
```

```
5862  016676  000413                          BR      2$              ;NO ERROR-SKIP
5863  016700  104015                          ERROR   15              ;REPORT FIRST ERROR
5864
5865  016702  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5866  016706  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5867  016710  001406                          BEQ     65$             ;IF ZERO - EXIT
5868  016712  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5869  016716  040000                          40000
5870  016720  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5871  016722  104016                          ERROR   16              ;REPORT NEXT ERROR
5872  016724  000770                          BR      64$             ;LOOP
5873  016726                          65$:
5874
5875  016726                          2$:
5876                                  ;:**********************************************************************
5877                                  ;*TEST 50       MID-TRANSFER SEEK ON WRITE (PART 1)
5878                                  ;*
5879                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5880                                  ;*      TRACK 0, SECTOR 25.  READ DATA BACK ONE SECTOR
5881                                  ;*      AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5882                                  ;*      DID TAKE PLACE.
5883                                  ;*
5884                                  ;:**********************************************************************
5885  016726  000004                  TST50:  SCOPE
5886  016730  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5887  016736  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5888  016740  104003                          ERROR   3               ;BAD INIT ERROR
5889
5890  016742  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5891  016746  000123                          WRDATA                  ;WRDATA
5892  016750  177000                          -1000                   ;-1000 WORDS
5893  016752  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5894  016754     025                          .BYTE   25              ;SECTOR 25
5895  016755     000                          .BYTE   0               ;TRACK 0
5896  016756  000312                          312                     ;CYLINDER 312
5897
5898  016760  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
5899  016764  000003                          3                       ;PATTERN 3
5900  016766  001000                          1000                    ;1000 WORDS
5901
5902  016770  104417                          TLOADRK                 ;LOAD RK REGS
5903  016772  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5904  016774  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5905
5906  016776  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5907  017000  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5908                                  ;       A TRACK ERROR PRINTED OUT AT THE END OF THE OPERATION INDICATES A
5909                                  ;       MID-TRANSFER HEAD SWITCH DID NOT OCCUR.
5910  017002  004437  041740                  JSR     R4,GENCOM
5911  017006  002007                          2007
5912  017010  001000                          1000
5913
5914  017012  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5915  017016  000121                          RDDATA                  ;RDDATA
5916  017020  177400                          -400                    ;-400 WORDS
5917  017022  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
```

```
5918  017024  000025                  .BYTE   25              ;SECTOR 25
5919  017025  000000                  .BYTE   0               ;TRACK 0
5920  017026  000312                  312                     ;CYLINDER 312
5921
5922  017030  104417                  TLOADRK                 ;LOAD RK REGS
5923  017032  104425                  TWAT48                  ;WAIT FOR INTERRUPT
5924  017034  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5925
5926  017036  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5927  017040  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5928
5929  017042  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5930  017046  000121                  RDDATA                  ;RDDATA
5931  017050  177400                  -400                    ;-400 WORDS
5932  017052  062414                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5933  017054  000000                  .BYTE   0               ;SECTOR 0
5934  017055  000001                  .BYTE   1               ;TRACK 1
5935  017056  000312                  312                     ;CYLINDER 312
5936
5937  017060  104417                  TLOADRK                 ;LOAD RK REGS
5938  017062  104425                  TWAT48                  ;WAIT FOR INTERRUPT
5939  017064  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5940
5941  017066  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5942  017070  104004                  ERROR   4  ;OR 5, 6  7, 10 ;REPORT ALL ERRORS
5943
5944  017072  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
5945  017076  100000                  100000
5946  017100  001000                  1000                    ;1000 WORDS
5947  017102  000413                  BR      1$              ;NO ERRORS-SKIP
5948  017104  104015                  ERROR   15              ;REPORT FIRST ERROR
5949
5950  017106  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5951  017112  005300          64$:    DEC     R0              ;DECREMENT COUNT
5952  017114  001406                  BEQ     65$             ;IF ZERO - EXIT
5953  017116  004437  041740          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5954  017122  040000                  40000
5955  017124  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5956  017126  104016                  ERROR   16              ;REPORT NEXT ERROR
5957  017130  000770                  BR      64$             ;LOOP
5958  017132                  65$:
5959
5960  017132                  1$:
5961                                  ;;*****************************************************************
5962                                  ;*TEST 51       MID-TRANSFER SEEK ON WRITE (PART 2)
5963                                  ;*
5964                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5965                                  ;*      TRACK 2, SECTOR 25.  READ DATA BACK ONE SECTOR
5966                                  ;*      AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5967                                  ;*      DID TAKE PLACE.
5968                                  ;*
5969                                  ;;*****************************************************************
5970  017132  000004          TST51:  SCOPE
5971  017134  012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5972  017142  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
5973  017144  104003                  ERROR   3               ;BAD INIT ERROR
```

```
5974
5975  017146  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
5976  017152  000123                  WRDATA                  ;WRDATA
5977  017154  177000                  -1000                   ;-1000 WORDS
5978  017156  063414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5979  017160     025                  .BYTE   25              ;SECTOR 25
5980  017161     002                  .BYTE   2               ;TRACK 2
5981  017162  000312                  312                     ;CYLINDER 312
5982
5983  017164  004437  041740          JSR     R4,GENCOM       ;GENERATE DATA
5984  017170  000004                  4                       ;PATTERN 4
5985  017172  001000                  1000                    ;1000 WORDS
5986
5987  017174  104417                  TLOADRK                 ;LOAD RK REGS
5988  017176  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5989  017200  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5990                          ;       A CYLINDER ERROR REPORTED AT THE END OF THE OPERATION INDICATES A
5991                          ;       MID-TRANSFER SEEK DID NOT OCCUR.
5992  017202  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5993  017204  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5994
5995  017206  004437  041740          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
5996  017212  002007                  2007
5997  017214  001000                  1000
5998
5999  017216  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
6000  017222  000121                  RDDATA                  ;RDDATA
6001  017224  177400                  -400                    ;-400 WORDS
6002  017226  061414                  IBUFF                   ;IBUFF IS BUFF ADDRESS
6003  017230     025                  .BYTE   25              ;SECTOR 25
6004  017231     002                  .BYTE   2               ;TRACK 2
6005  017232  000312                  312                     ;CYLINDER 312
6006
6007  017234  104417                  TLOADRK                 ;LOAD RK REGS
6008  017236  104425                  TWAT48                  ;WAIT FOR INTERRUPT
6009  017240  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6010
6011  017242  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6012  017244  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6013
6014  017246  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
6015  017252  000121                  RDDATA                  ;RDDATA
6016  017254  177400                  -400                    ;-400 WORDS
6017  017256  062414                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
6018  017260     000                  .BYTE   0               ;SECTOR 0
6019  017261     000                  .BYTE   0               ;TRACK 0
6020  017262  000313                  313                     ;CYLINDER 313
6021
6022  017264  104417                  TLOADRK                 ;LOAD RK REGS
6023  017266  104425                  TWAT48                  ;WAIT FOR INTERRUPT
6024  017270  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6025
6026  017272  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6027  017274  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6028
6029  017276  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
```

```
                        100000
                        1000                    ;1000 WORDS
                        BR      1$              ;NO MISCOMPARES-SKIP
                        ERROR   15              ;REPORT 1ST ERROR

                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
64$:                    DEC     R0              ;DECREMENT COUNT
                        BEQ     65$             ;IF ZERO - EXIT
                        JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
                        40000
                        BR      65$             ;NO MORE ERRORS - EXIT
                        ERROR   16              ;REPORT NEXT ERROR
                        BR      64$             ;LOOP
65$:

1$:
;;*****************************************************************
;*TEST 52        TWO SECTOR READ DATA (PART 1)
;*
;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
;*      TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS
;*      READ.
;*
;*      NOTE:   TWO SECTOR WRITE DATA (PART 1) MUST BE
;*              EXECUTED BEFORE THIS TEST.
;*
;;*****************************************************************
TST52:  SCOPE
        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
        TSSINIT                 ;CLEAR SUBSYSTEM
        ERROR   3               ;BAD INIT ERROR

        ;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA (PART 1)

;       GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA PART 1
        JSR     R4,GENCOM       ;GENERATE DATA
        15                      ;PATTERN 15
        1000                    ;1000 WORDS

        JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
        2007
        1000

        JSR     R4,LRLOAD       ;LOAD "L" REGS
        RDDATA                  ;RDDATA
        -1000                   ;-1000 WORDS
        IBUFF                   ;IBUFF IS BUFF ADDRESS
        .BYTE   0               ;SECTOR 0
        .BYTE   0               ;TRACK 0
        312                     ;CYLINDER 312

        TLOADRK                 ;LOAD RK REGS
        TWAT96                  ;WAIT FOR INTERRUPT
        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
```

```
                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
                        JSR     R4,GENCOM       ;COMPARE DATA
                        100000
                        1000                    ;1000 WORDS
                        BR      1$              ;NO MISCOMPARES-SKIP
                        ERROR   15
                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
                64$:    DEC     R0              ;DECREMENT COUNT
                        BEQ     65$             ;IF ZERO - EXIT
                        JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
                        40000
                        BR      65$             ;NO MORE ERRORS - EXIT
                        ERROR   16              ;REPORT NEXT ERROR
                        BR      64$             ;LOOP
                65$:

                1$:
                ;*****************************************************************
                ;*TEST 53        TWO SECTOR READ DATA (PART 2)
                ;*
                ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
                ;*      TRACK 0, SECTOR 23.  VERIFY THAT CORRECT DATA IS
                ;*      READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.
                ;*
                ;*      NOTE:   TWO SECTOR WRITE DATA (PART 2) MUST BE
                ;*              EXECUTED BEFORE THIS TEST.
                ;*
                ;*****************************************************************
                TST53:  SCOPE
                        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
                        TSSINIT                 ;CLEAR SUBSYSTEM
                        ERROR   3               ;BAD INIT ERROR

                        ;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 2)

                        JSR     R4,GENCOM       ;GENERATE DATA
                        16                      ;PATTERN 16
                        1000                    ;1000 WORDS

                        JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
                        2007
                        1000
                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                        RDDATA                  ;RDDATA
                        -1000                   ;-1000 WORDS
                        IBUFF                   ;IBUFF IS BUFF ADDRESS
                        .BYTE   23              ;SECTOR 23
                        .BYTE   0               ;TRACK 0
                        312                     ;CYLINDER 312

                        TLOADRK                 ;LOAD RK REGS
                        TWAT96                  ;WAIT FOR INTERRUPT
                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

                        JSR     R4,GENCOM       ;COMPARE DATA
                        100000
                        1000                    ;1000 WORDS
                        BR      1$              ;NO MISCOMPARES-SKIP
                        ERROR   15              ;REPORT 1ST ERROR

                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
                64$:    DEC     R0              ;DECREMENT COUNT
                        BEQ     65$             ;IF ZERO - EXIT
                        JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
                        40000
                        BR      65$             ;NO MORE ERRORS - EXIT
                        ERROR   16              ;REPORT NEXT ERROR
                        BR      64$             ;LOOP
                65$:

                1$:
                ;*****************************************************************
                ;*TEST 54        TWO SECTOR READ DATA (PART 3)
                ;*
                ;*      ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312,
                ;*      TRACK 0, SECTOR 10.  VERIFY THAT ALL 401 WORDS
                ;*      ARE PLACED IN MEMORY.
                ;*
                ;*      NOTE:   TWO SECTOR WRITE DATA (PART 3) MUST BE
                ;*              EXECUTED BEFORE THIS TEST.
                ;*
                ;*****************************************************************
                TST54:  SCOPE
                        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
                        TSSINIT                 ;CLEAR SUBSYSTEM
                        ERROR   3               ;BAD INIT ERROR

                ;       GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 3)

                        JSR     R4,GENCOM       ;GENERATE DATA
                        2                       ;PATTERN 2
                        401                     ;401 WORDS

                        JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
                        2007
                        1000

                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                        RDDATA                  ;RDDATA
                        -401                    ;-401 WORDS
                        IBUFF                   ;IBUFF IS BUFF ADDRESS
                        .BYTE   10              ;SECTOR 10
                        .BYTE   0               ;TRACK 0
                        312                     ;CYLINDER 312

                        TLOADRK                 ;LOAD RK REGS
                        TWAT96                  ;WAIT FOR INTERRUPT
```

```
                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

017672  004437  041740  JSR     R4,GENCOM       ;COMPARE DATA
017676  100000          100000
017700  000401          401                     ;401 WORDS
017702  000413          BR      1$              ;NO MISCOMPARES-SKIP
017704  104015          ERROR   15              ;PRINT FIRST ERROR

017706  013700  001634  MOV     ERRLMT,R0       ;GET ERROR LIMIT
017712  005300  64$:    DEC     R0              ;DECREMENT COUNT
017714  001406          BEQ     65$             ;IF ZERO - EXIT
017716  004437  041740  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
017722  040000          40000
017724  000402          BR      65$             ;NO MORE ERRORS - EXIT
017726  104016          ERROR   16              ;REPORT NEXT ERROR
017730  000770          BR      64$             ;LOOP
017732          65$:
017732          1$:
                ;;*****************************************************************
                ;*TEST 55        MID-TRANSFER SEEK ON READ (PART 1)
                ;*
                ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
                ;*      TRACK 0, SECTOR 25.  VERIFY THAT CORRECT DATA IS
                ;*      READ AND A MID-TRANSFER SEEK DOES OCCUR.
                ;*
                ;*      NOTE:   MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE
                ;*              EXECUTED BEFORE THIS TEST.
                ;*
                ;*****************************************************************
000004  TST55:  SCOPE
012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
104416          TSSINIT                         ;CLEAR SUBSYSTEM
104003          ERROR   3                       ;BAD INIT ERROR

                ;       GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
004437  041740  JSR     R4,GENCOM       ;GENERATE DATA
000003          3                       ;PATTERN 3
001000          1000                    ;1000 WORDS

004437  041740  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
002007          2007
001000          1000

004437  035530  JSR     R4,LRLOAD       ;LOAD "L" REGS
000121          RDDATA                  ;RDDATA
177000          -1000                   ;-1000 WORDS
061414          IBUFF                   ;IBUFF IS BUFF ADDRESS
000025          .BYTE   25              ;SECTOR 25
000000          .BYTE   0               ;TRACK 0
000312          312                     ;CYLINDER 312

104417          TLOADRK                 ;LOAD RK REGS
104430          TWAT96                  ;WAIT FOR INTERRUPT
```

```
6254  020010  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6255
6256  020012  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6257  020014  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6258
6259  020016  004437  041740                  JSR     R4,GENCOM       ;COMPARE DATA
6260  020022  100000                          100000
6261  020024  001000                          1000                    ;1000 WORDS
6262  020026  000413                          BR      1$              ;NO MISCOMPARES-SKIP
6263  020030  104015                          ERROR   15              ;PRINT FIRST ERROR
6264
6265  020032  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6266  020036  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6267  020040  001406                          BEQ     65$             ;IF ZERO - EXIT
6268  020042  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6269  020046  040000                          40000
6270  020050  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6271  020052  104016                          ERROR   16              ;REPORT NEXT ERROR
6272  020054  000770                          BR      64$             ;LOOP
6273  020056                          65$:
6274  020056                          1$:
6275                                  ;;*****************************************************************
6276                                  ;*TEST 56       MID-TRANSFER SEEK ON READ (PART 2)
6277                                  ;*
6278                                  ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6279                                  ;*      TRACK 2, SECTOR 25.  VERIFY THAT CORRECT DATA IS
6280                                  ;*      READ AND A MID-TRANSFER SEEK DOES OCCUR.
6281                                  ;*
6282                                  ;*      NOTE:   MID-TRANSFER SEEK ON WRITE (PART 2) MUST BE
6283                                  ;*              EXECUTED BEFORE THIS TEST.
6284                                  ;*
6285                                  ;;*****************************************************************
6286  020056  000004                  T$T56:  SCOPE
6287  020060  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6288  020066  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6289  020070  104003                          ERROR   3               ;BAD INIT ERROR
6290
6291                                  ;       GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)
6292  020072  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
6293  020076  000004                          4                       ;PATTERN 4
6294  020100  001000                          1000                    ;1000 WORDS
6295
6296  020102  004437  041740                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6297  020106  002007                          2007
6298  020110  001000                          1000
6299
6300  020112  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6301  020116  000121                          RDDATA                  ;RDDATA
6302  020120  177000                          -1000                   ;-1000 WORDS
6303  020122  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
6304  020124     025                          .BYTE   25              ;SECTOR 25
6305  020125     002                          .BYTE   2               ;TRACK 2
6306  020126  000312                          312                     ;CYLINDER 312
6307
6308  020130  104417                          TLOADRK                 ;LOAD RK REGS
6309  020132  104430                          TWAT96                  ;WAIT FOR INTERRUPT
```

```
6310  020134  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6311
6312  020136  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6313  020140  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6314
6315  020142  004437  041740                  JSR     R4,GENCOM       ;COMPARE DATA
6316  020146  100000                          100000
6317  020150  001000                          1000                    ;1000 WORDS
6318  020152  000413                          BR      1$              ;NO MISCOMPARES-SKIP
6319  020154  104015                          ERROR   15              ;REPORT FIRST ERROR
6320
6321  020156  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6322  020162  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6323  020164  001406                          BEQ     65$             ;IF ZERO - EXIT
6324  020166  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6325  020172  040000                          40000
6326  020174  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6327  020176  104016                          ERROR   16              ;REPORT NEXT ERROR
6328  020200  000770                          BR      64$             ;LOOP
6329  020202                          65$:
6330  020202                          1$:
6331                                  ;;*****************************************************************
6332                                  ;*TEST 57       CYLINDER ADDRESS OVERFLOW (PART 1)
6333                                  ;*
6334                                  ;*      ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
6335                                  ;*      TRACK 2, SECTOR 25.  MAKE SURE CYLINDER ADDRESS
6336                                  ;*      OVERFLOW ERROR DOES NOT OCCUR.
6337                                  ;*
6338                                  ;;*****************************************************************
6339  020202  000004                  TST57:  SCOPE
6340  020204  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6341  020212  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6342  020214  104003                          ERROR   3               ;BAD INIT ERROR
6343
6344  020216  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6345  020222  000121                          RDDATA                  ;RDDATA
6346  020224  177400                          -400                    ;-400 WORDS
6347  020226  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
6348  020230     025                          .BYTE   25              ;SECTOR 25
6349  020231     002                          .BYTE   2               ;TRACK 2
6350  020232  000632                          632                     ;CYLINDER 632
6351
6352  020234  104417                          TLOADRK                 ;LOAD RK REGS
6353  020236  104434                          TWAT159                 ;WAIT FOR INTERRUPT
6354  020240  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6355
6356  020242  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6357  020244  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6358                                  ;;*****************************************************************
6359                                  ;*TEST 60       CYLINDER ADDRESS OVERFLOW (PART 2)
6360                                  ;*
6361                                  ;*      ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632,
6362                                  ;*      TRACK 2, SECTOR 25.  MAKE SURE CYLINDER ADDRESS
6363                                  ;*      OVERFLOW ERROR DOES OCCUR.
6364                                  ;*
6365                                  ;;*****************************************************************
```

```
                                TST60:  SCOPE
020246  000004
020250  012737  000012  001262            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
020256  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
020260  104003                            ERROR   3               ;BAD INIT ERROR

020262  004437  035530                    JSR     R4,LRLOAD       ;LOAD "L" REGS
020266  000121                            RDDATA                  ;RDDATA
020270  177377                            -401                    ;-401 WORDS
020272  061414                            IBUFF                   ;IBUFF IS BUFF ADDRESS
020274     025                            .BYTE   25              ;SECTOR 25
020275     002                            .BYTE   2               ;TRACK 2
020276  000632                            632                     ;CYLINDER 632

020300  104417                            TLOADRK                 ;LOAD RK REGS
020302  104434                            TWAT159                 ;WAIT FOR INTERRUPT
020304  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

020306  104422                            TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
020310  001000                            COERR                   ;CYLINDER OVERFLOW
020312  000000                            0
020314  000000                            0
020316  104004                            ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES


                                .SBTTL  **18 BIT DATA TRANSFER TESTS

                                ;;*****************************************************************
                                ;*TEST 61       FORMAT IN 24 SECTOR FORMAT
                                ;*
                                ;*      FORMAT CYLINDER 312, TRACK 0, AND TRACK 1 FOR 24 SECTOR
                                ;*      FORMAT.  VERIFY FORMAT AND THAT DATA LATE DID NOT
                                ;*      OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER
                                ;*      AFTER READ HEADER.
                                ;*
                                ;*****************************************************************
020320  000004                TST61:  SCOPE
020322  012737  000012  001262            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
020330  012737  000312  001676            MOV     #312,REFMT      ;SET REFORMAT SWITCH
020336  012737  020346  001110            MOV     #1$,$LPERR      ;SET LOCAL LOOP ON ERROR
020344  005001                            CLR     R1              ;CLEAR R1 FOR TRACK COUNTER
020346                          1$:
020346  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
020350  104003                            ERROR   3               ;BAD INIT ERROR
020352  012737  010127  001600            MOV     #WRHEAD!CFMT,L.CS1  ;SET UP FOR WRITE HEADER 24(8) SECTOR MODE
020360  013737  001626  001610            MOV     DRVNUM,L.CS2    ;SET DRIVE NUMBER
020366  012737  000074  001602            MOV     #74,L.WC        ;SET WORD COUNT
020374  110137  001607                    MOVB    R1,L.DT         ;LOAD TRACK ADDRESS
020400  012737  063414  001604            MOV     #OBUFF,L.BA     ;       SET BUS ADDRESS
020406  012737  000312  001614            MOV     #312,L.DCYL     ;     CYLINDER ADDRESS

020414  004437  041740                    JSR     R4,GENCOM       ;GENERATE HEADER
020420  001200                            1200                    ;INCLUDE BAD SECTOR BITS

020422  104417                            TLOADRK                 ;LOAD RK REGS
020424  104434                            TWAT159                 ;WAIT FOR INTERRUPT
020426  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
6422
6423  020430  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6424  020432  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
6425
6426  020434  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
6427
6428  020436  005701                          TST     R1              ;R1 POINTING TO TRACK 0
6429  020440  001002                          BNE     2$              ;NO-SKIP
6430  020442  005201                          INC     R1              ;BUMP TO TRACK 1
6431  020444  000740                          BR      1$              ;LOOP
6432  020446  012737  020456  001110  2$:     MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
6433  020454  005001                          CLR     R1              ;CLEAR TRACK POINTER
6434  020456                          3$:
6435  020456  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6436  020460  104003                          ERROR   3               ;BAD INIT ERROR
6437  020462  012737  010125  001600          MOV     #RDHEAD!CFMT,L.CS1  ;LOAD READ 24(8) SECTOR FORMAT
6438  020470  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
6439  020476  110137  001607                  MOVB    R1,L.DT         ;LOAD TRACK
6440  020502  012737  000312  001614          MOV     #312,L.DCYL     ;LOAD CYLINDER
6441
6442  020510  004437  036050                  JSR     R4,RDSTHD       ;GO READ STANDARD HEADER
6443  020514  104421                          TCHKOP                  ;RETURN IF CERR W/O DATA LATE SET
6444  020516  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL OTHER ERRORS
6445  020520  104013                          ERROR   13              ;REPORT DATA LATE
6446  020522  104002                          ERROR   2               ;REPORT "OPERATION TO SLOW" OR "HEADER
6447                                                                  ;O NOT FOUND" MESSAGE
6448
6449  020524  104415                          SCOP1                   ;LOCAL LOOP TO 3$ ON ERROR
6450  020526  004437  041740                  JSR     R4,GENCOM       ;GENERATE & COMPARE HEADERS
6451  020532  101200                          101200                  ;INCLUDING BAD SECTOR LISTS
6452  020534  000413                          BR      4$              ;NO MISCOMPARES-SKIP
6453  020536  104015                          ERROR   15              ;REPORT FIRST MISCOMPARE
6454
6455  020540  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6456  020544  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6457  020546  001406                          BEQ     65$             ;IF ZERO - EXIT
6458  020550  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6459  020554  040000                          40000
6460  020556  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6461  020560  104016                          ERROR   16              ;REPORT NEXT ERROR
6462  020562  000770                          BR      64$             ;LOOP
6463  020564                          65$:
6464
6465  020564  104415                  4$:     SCOP1                   ;LOCAL LOOP TO 3$
6466  020566  005701                          TST     R1              ;POINTING TO TRACK 1
6467  020570  001002                          BNE     5$              ;NO-EXIT
6468  020572  005201                          INC     R1              ;BUMP TO TRACK 1
6469  020574  000730                          BR      3$              ;LOOP
6470
6471  020576                          5$:
6472                                  ;;*****************************************************************
6473                                  ;*TEST 62       24 SECTOR FORMAT DATA TRANSFER (PART 1)
6474                                  ;*
6475                                  ;*      ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
6476                                  ;*      TO CYLINDER 312, TRACK 0, SECTOR 0.  READ SECTOR BACK
6477                                  ;*      AND MAKE SURE IT IS CORRECT.
```

```
                                        ;*
                                        ;**********************************************************************
                                TST62:  SCOPE
020600  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
020606  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
020614  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
020616  104003                          ERROR   3               ;BAD INIT ERROR

020620  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
020624  000013                          13                      ;PATTERN 13
020626  000400                          400                     ;400 WORDS

020630  004437  041740                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
020634  002007                          2007
020636  000400                          400

020640  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
020644  010123                          WRDATA!CFMT                     ;WRDATA!CFMT
020646  177400                          -400                    ;-400 WORDS
020650  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
020652     000                          .BYTE   0               ;SECTOR 0
020653     000                          .BYTE   0               ;TRACK 0
020654  000312                          312                     ;CYLINDER 312

020656  104417                          TLOADRK                 ;LOAD RK REGS
020660  104430                          TWAT96                  ;WAIT FOR INTERRUPT
020662  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

020664  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
020666  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

020670  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
020674  010121                          RDDATA!CFMT                     ;RDDATA!CFMT
020676  177400                          -400                    ;-400 WORDS
020700  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
020702     000                          .BYTE   0               ;SECTOR 0
020703     000                          .BYTE   0               ;TRACK 0
020704  000312                          312                     ;CYLINDER 312

020706  104417                          TLOADRK                 ;LOAD RK REGS
020710  104424                          TWAT32                  ;WAIT FOR INTERRUPT
020712  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

020714  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
020716  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

020720  004437  041740                  JSR     R4,GENCOM       ;COMPARE DATA
020724  100000                          100000
020726  000400                          400                     ;400 WORDS
020730  000413                          BR      1$              ;NO MISCOMPARES-SKIP
020732  104015                          ERROR   15              ;REPORT 1ST ERROR

020734  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
020740  005300                  64$:    DEC     R0              ;DECREMENT COUNT
020742  001406                          BEQ     65$             ;IF ZERO - EXIT
020744  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
```

```
                        040000                          40000
                        000402                          BR      65$             ;NO MORE ERRORS - EXIT
                        104016                          ERROR   16              ;REPORT NEXT ERROR
                        000770                          BR      64$             ;LOOP
020760                                          65$:
                                                1$:
020760                                          ;*****************************************************************
                                                ;*TEST 63       24 SECTOR FORMAT DATA TRANSFER (PART 2)
                                                ;*
                                                ;*      LOAD A LOCATION WITH BAD PARITY.  ISSUE A WRITE DATA OF
                                                ;*      400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0,
                                                ;*      SECTOR 0 WITH BUFFER BEGINNING 112 WORDS BEFORE WORD
                                                ;*      WITH BAD PARITY.  MAKE SURE UNIBUS PARITY ERROR DOES NOT SET.
                                                ;*      READ SECTOR BACK AND MAKE SURE IT IS CORRECT.
                                                ;*
                                                ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
                                                ;*              EXISTS FOR SPECIFIED LOCATION.
                                                ;*
                                                ;*****************************************************************
020760  000004                                  TST63:  SCOPE
020762  012737  000012  001262                          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
020770  012737  000312  001676                          MOV     #312,REFMT      ;SET REFORMAT SWITCH
020776  032737  000100  001664                          BIT     #PARPRE,OPTFLG  ;PARITY OPTION PRESENT?
021004  001556                                          BEQ     4$              ;YES-SKIP
                                                
021006  004437  041740                        1$:     JSR     R4,GENCOM       ;GENERATE DATA
021012  000007                                          7                       ;PATTERN 7
021014  000400                                          400                     ;400 WORDS
                                                
021016  032737  002000  001664                          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
021024  001023                                          BNE     11$             ;YES - SKIP
021026  012746  000340                                  MOV     #PR7,-(SP)      ;SET PRIORITY TO 7
021032  012746  021040                                  MOV     #10$,-(SP)      ;SET ADDRESS
021036  000002                                          RTI
021040                                          10$:
                                                
021040  012777  000004  160624                          MOV     #BIT2,@CSRPTR   ;SET WRONG PARITY WRITE
021046  012737  100200  063640                          MOV     #100200,OBUFF+224       ;WRITE WITH BAD PARITY
021054  012777  000001  160610                          MOV     #BIT0,@CSRPTR   ;CLEAR WRONG PARITY, SET IE
                                                
021062  013746  001622                        6$:     MOV     RKPRI,-(SP)     ;RESTORE PRIORITY
021066  012746  021074                                  MOV     #11$,-(SP)
021072  000002                                          RTI
021074                                          11$:
021074  104416                                          TSSINIT                 ;CLEAR SUBSYSTEM
021076  104003                                          ERROR   3               ;BAD INIT ERROR
021100  004437  035530                                  JSR     R4,LRLOAD       ;LOAD "L" REGS
021104  010123                                          WRDATA!CFMT                     ;WRDATA!CFMT
021106  177400                                          -400                    ;-400 WORDS
021110  063414                                          OBUFF                   ;OBUFF IS BUFF ADDRESS
021112     000                                          .BYTE   0               ;SECTOR 0
021113     000                                          .BYTE   0               ;TRACK 0
021114  000312                                          312                     ;CYLINDER 312
                                                
021116  013704  001672                                  MOV     CSRPTR,R4       ;LOAD R4 WITH CSR POINTER
```

```
6590  021122  005000                          CLR     R0              ;CLEAR R0 FOR COUNTING
6591  021124  012701  001662                  MOV     #INTSET,R1      ;LOAD R1 WITH POINTER INTERRUPT FLAG
6592  021130  012777  061400  160462          MOV     #SPCHLR,@RKVEC  ;SET INT VECTOR FOR RK611
6593  021136  012777  061410  160552          MOV     #SPCPAR,@MMVECA ;   AND PARITY ERROR
6594  021144  012737  021342  001264          MOV     #45$,$ESCAPE    ;SET ESCAPE FOR ERROR
6595  021152  104417                          TLOADRK                 ;LOAD RK REGS
6596  021154  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
6597  021162  001433                          BEQ     45$             ;NO - SKIP
6598  021164  012737  000016  177746          MOV     #16,177746      ;SET TO DISABLE CACHE
6599  021172  012777  000000  160510          MOV     #0,@KWLADD      ;DISABLE CLOCK INTERRUPTS
6600  021200  012777  000000  157736          MOV     #0,@$TKS        ;TURN OFF KEYBOARD INTERRUPTS
6601  021206  012714  170000                  MOV     #170000,(R4)    ;SET TO FORCE PARITY ERROR
6602
6603                                  ;**********************
6604  021212  105711                 40$:    TSTB    (R1)            ;LOOP TO WAIT FOR INTERRUPT. THE
6605  021214  003005                          BGT     43$             ;CODE BETWEEN THE STARS IS SETUP SO ALL
6606  021216  005300                          DEC     R0              ;BYTES HAVE PARITY OF 1, HENCE NO PARITY
6607  021220  100774                          BMI     40$             ;TRAPS ON AN 11/70. IF THIS CODE IS CHA-
6608  021222  000240                          NOP                     ;NGED, REMEMBER ALL BYTES MUST HAVE AN
6609  021224  003372                          BGT     40$             ;EVEN NUMBER OF BITS.
6610  021226  000240                          NOP
6611  021230  005014                 43$:    CLR     (R4)            ;CLEAR ERROR FORCE
6612                                  ;**********************
6613
6614  021232  005037  177746                  CLR     177746          ;ENABLE CACHE
6615  021236  012777  000100  160444          MOV     #BIT6,@KWLADD   ;ENABLE CLOCK INTERRUPTS
6616  021244  012777  000100  157672          MOV     #100,@$TKS      ;TURN ON KEYBOARD INTERRUPTS
6617  021252  104430                 45$:    TWAT96                  ;WAIT FOR INTERRUPT
6618  021254  000414                          BR      46$             ;TO SLOW/NOT COMPLETE ERROR
6619  021256  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
6620  021264  001024                          BNE     48$             ;YES - SKIP
6621  021266  005077  160400                  CLR     @CSRPTR         ;CLEAR PARITY DETECT
6622  021272  005037  063640                  CLR     OBUFF+224       ;CLEAR BAD PARITY WORD
6623  021276  012777  000001  160366          MOV     #1,@CSRPTR      ;ENABLE PARITY DETECT
6624  021304  000414                          BR      48$             ;SKIP
6625
6626  021306  032737  002000  001664  46$:    BIT     #CP1170,OPTFLG  ;TEST IF 11/70
6627  021314  001007                          BNE     47$             ;YES - SKIP
6628  021316  005077  160350                  CLR     @CSRPTR         ;CLEAR PARITY DETECT
6629  021322  005037  063640                  CLR     OBUFF+224       ;CLEAR BAD PARITY WORD
6630  021326  012777  000001  160336          MOV     #1,@CSRPTR      ;ENABLE PARITY DETECT
6631  021334  104002                 47$:    ERROR   2               ;REPORT TO SLOW ERROR
6632
6633  021336                         48$:
6634  021336  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6635  021340  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6636                                  ;       A UNIBUS PARITY ERROR REPORTED AT THIS
6637                                  ;       TIME INDICATES CONTROLLER IS NOT
6638                                  ;       PROPERLY TRANSFERRING DATA IN 18 BIT MODE
6639
6640
6641  021342  012777  034354  160250  45$:    MOV     #INTHLR,@RKVEC  ;RESET INT VECTOR FOR RK611
6642  021350  012777  034364  160340          MOV     #PERHLR,@MMVECA ;   AND PARITY ERROR
6643  021356  012777  000100  160324          MOV     #BIT6,@KWLADD   ;ENABLE CLOCK
6644  021364  012777  000100  157552          MOV     #100,@$TKS      ;ENABLE KEYBOARD INTERRUPTS
6645
```

```
6646
6647                                    ;;*****************************************************************
6648                                    ;*TEST 64       24 SECTOR FORMAT DATA TRANSFER (PART 3)
6649                                    ;*
6650                                    ;*      ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT
6651                                    ;*      TO CYLINDER 312, TRACK 0, SECTOR 23.  READ SECTOR BACK
6652                                    ;*      AND MAKE SURE IT IS CORRECT.  MAKE SURE THAT MID-TRANSFER
6653                                    ;*      SEEK HAS TAKEN PLACE.
6654                                    ;*
6655                                    ;;*****************************************************************
6656  021372  000004                    TST64:  SCOPE
6657  021374  012737  000012  001262            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6658  021402  012737  000312  001676            MOV     #312,REFMT      ;SET REFORMAT SWITCH
6659  021410  004737  034434                    JSR     PC,OPTTST       ;SET UP OPTIONS
6660  021414  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
6661  021416  104003                            ERROR   3               ;BAD INIT ERROR
6662
6663  021420  004437  041740                    JSR     R4,GENCOM       ;GENERATE DATA
6664  021424  000015                            15                      ;PATTERN 15
6665  021426  001000                            1000                    ;1000 WORDS
6666
6667  021430  004437  041740                    JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6668  021434  002007                            2007
6669  021436  001000                            1000
6670
6671  021440  004437  035530                    JSR     R4,LRLOAD       ;LOAD "L" REGS
6672  021444  010123                            WRDATA!CFMT                     ;WRDATA!CFMT
6673  021446  177000                            -1000                   ;-1000 WORDS
6674  021450  063414                            OBUFF                   ;OBUFF IS BUFF ADDRESS
6675  021452     023                            .BYTE   23              ;SECTOR 23
6676  021453     000                            .BYTE   0               ;TRACK 0
6677  021454  000312                            312                     ;CYLINDER 312
6678
6679  021456  104417                            TLOADRK                 ;LOAD RK REGS
6680  021460  104430                            TWAT96                  ;WAIT FOR INTERRUPT
6681  021462  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6682
6683  021464  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6684  021466  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6685
6686  021470  004437  035530                    JSR     R4,LRLOAD       ;LOAD "L" REGS
6687  021474  010121                            RDDATA!CFMT                     ;RDDATA!CFMT
6688  021476  177000                            -1000                   ;-1000 WORDS
6689  021500  061414                            IBUFF                   ;IBUFF IS BUFF ADDRESS
6690  021502     023                            .BYTE   23              ;SECTOR 23
6691  021503     000                            .BYTE   0               ;TRACK 0
6692  021504  000312                            312                     ;CYLINDER 312
6693
6694  021506  104417                            TLOADRK                 ;LOAD RK REGS
6695  021510  104426                            TWAT64                  ;WAIT FOR INTERRUPT
6696  021512  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6697
6698  021514  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6699  021516  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6700
6701  021520  004437  041740                    JSR     R4,GENCOM       ;COMPARE DATA
```

```
                                100000
                                1000                    ;1000 WORDS
                                BR      1$              ;NO MISCOMPARES-SKIP
                                ERROR   15              ;REPORT FIRST ERROR

                                MOV     ERRLMT,R0       ;GET ERROR LIMIT
                        64$:    DEC     R0              ;DECREMENT COUNT
                                BEQ     65$             ;IF ZERO - EXIT
                                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
                                40000
                                BR      65$             ;NO MORE ERRORS - EXIT
                                ERROR   16              ;REPORT NEXT ERROR
                                BR      64$             ;LOOP
                        65$:

                        1$:

                        .SBTTL  **SPECIAL DATA TRANSFER TESTS

                        ;;*****************************************************************
                        ;*TEST 65       MULTI-SECTOR DATA TRANSFER AND BSE
                        ;*
                        ;*      FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH
                        ;*      SECTOR 1 MARKED BAD.  ISSUE A WRITE DATA OF 1000 WORDS
                        ;*      TO CYLINDER 312, TRACK 0, SECTOR 0.  MAKE SURE BAD SECTOR
                        ;*      ERROR SETS AND RKDA IS CORRECT.  READ SECTOR 0 AND
                        ;*      MAKE SURE IT IS CORRECT.
                        ;*
                        ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0,
                        ;*      SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS AND THE
                        ;*      PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.
                        ;*
                        ;*****************************************************************
000004          TST65:  SCOPE
012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
104416                          TSSINIT                 ;CLEAR SUBSYSTEM
104003                          ERROR   3               ;BAD INIT ERROR

004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
000127                          WRHEAD                  ;WRHEAD
177676                          -102                    ;-102 WORDS
063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
   000                          .BYTE   0               ;SECTOR 0
   000                          .BYTE   0               ;TRACK 0
000312                          312                     ;CYLINDER 312

004437  041740                  JSR     R4,GENCOM       ;BUILD HEADERS
000600                          600

042737  040000  063424          BIC     #BIT14,OBUFF+10 ;MARK SECTOR 1 BAD
042737  040000  063426          BIC     #BIT14,OBUFF+12 ;CORRECT HURC

104417                          TLOADRK                 ;LOAD RK REGS
104431                          TWAIT12                 ;WAIT FOR INTERRUPT
104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
                                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
021650  104421                                  ERROR   4  ;OR 5, 6, 7   ;REPORT ALL ERRORS

021652  104004                                  JSR     R4,GENCOM       ;GENERATE DATA
021654  004437  041740                          16                      ;PATTERN 16
021660  000016                                  1000                    ;1000 WORDS
021662  001000

021664  004437  041740                          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
021670  002007                                  2007
021672  001000                                  1000

021674  004437  035530                          JSR     R4,LRLOAD       ;LOAD "L" REGS
021700  000123                                  WRDATA                  ;WRDATA
021702  177000                                  -1000                   ;-1000 WORDS
021704  063414                                  OBUFF                   ;OBUFF IS BUFF ADDRESS
021706     000                                  .BYTE   0               ;SECTOR 0
021707     000                                  .BYTE   0               ;TRACK 0
021710  000312                                  312                     ;CYLINDER 312

021712  104417                                  TLOADRK                 ;LOAD RK REGS
021714  104424                                  TWAT32                  ;WAIT FOR INTERRUPT
021716  104002                                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

021720  104422                                  TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERR
021722  000000                                  0
021724  000100                                  BSERR                   ;BAD SECTOR ERROR
021726  000000                                  0
021730  104004                                  ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
021732  005037  047462                          CLR     GRP4ER          ;CLEAR GROUP 4 ERRORS
021736  004437  037322                          JSR     R4,CHKCTS       ;CHECK CYL, TRK, SECT CORRECT AFTER ABORTED WRITE
021742  032737  000020  047462                  BIT     #TRKERR,GRP4ER  ;TRK IN ERROR?
021750  001416                                  BEQ     1$              ;NO-SKIP
021752  012737  054156  001450                  MOV     #EM13,EM1ON     ;"TRACK ADDRESS INCORRECT"
021760  013737  047436  001202                  MOV     EXPTRK,$REG10   ;EXPECTED VALUE
021766  013737  047450  001204                  MOV     REALTRK,$REG11  ;REAL VALUE
021774  012737  050147  060770                  MOV     #OPER37,DF010A  ;"AFTER WRITE DATA TERMINATED WITH BSE"
022002  104010                                  ERROR   10
022004  000527                                  BR      5$              ;EXIT

022006  032737  000040  047462  1$:             BIT     #SECERR,GRP4ER  ;SECTOR IN ERROR?
022014  001422                                  BEQ     3$              ;NO-SKIP
022016  012737  054206  001450                  MOV     #EM14,EM1ON     ;"SECTOR ADDRESS INCORRECT"
022024  012737  050147  060770                  MOV     #OPER37,DF010A  ;"AFTER WRITE DATA ABORTED WITH BSE"
022032  013737  047434  001202                  MOV     EXPSEC,$REG10   ;EXPECTED VALUE
022040  013737  047452  001204                  MOV     REALSEC,$REG11  ;REAL VALUE
022046  104010                                  ERROR   10
022050  000505                                  BR      5$              ;EXIT
022052  104415                                  SCOP1                   ;LOCAL LOOP TO BEGINNING OF TEST
022054  012737  022062  001110                  MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
022062                                  3$:
022062  104416                                  TSSINIT                 ;CLEAR SUBSYSTEM
022064  104003                                  ERROR   3               ;BAD INIT ERROR
022066  004437  035530                          JSR     R4,LRLOAD       ;LOAD "L" REGS
022072  000121                                  RDDATA                  ;RDDATA
022074  177400                                  -400                    ;-400 WORDS
```

```
6814  022076  061414                  IBUFF                   ;IBUFF IS BUFF ADDRESS
6815  022100     000                  .BYTE   0               ;SECTOR 0
6816  022101     000                  .BYTE   0               ;TRACK 0
6817  022102  000312                  312                     ;CYLINDER 312
6818
6819  022104  104417                  TLOADRK                 ;LOAD RK REGS
6820  022106  104424                  TWAT32                  ;WAIT FOR INTERRUPT
6821  022110  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6822
6823  022112  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6824  022114  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6825
6826  022116  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA
6827  022122  100000                  100000
6828  022124  000400                  400                     ;400 WORDS
6829  022126  000413                  BR      4$              ;NO MISCOMPARES-EXIT
6830  022130  104015                  ERROR   15              ;REPORT FIRST ERROR
6831
6832  022132  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
6833  022136  005300          64$:    DEC     R0              ;DECREMENT COUNT
6834  022140  001406                  BEQ     65$             ;IF ZERO - EXIT
6835  022142  004437  041740          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6836  022146  040000                  40000
6837  022150  000402                  BR      65$             ;NO MORE ERRORS - EXIT
6838  022152  104016                  ERROR   16              ;REPORT NEXT ERROR
6839  022154  000770                  BR      64$             ;LOOP
6840  022156                  65$:
6841
6842  022156  004437  041740    4$:   JSR     R4,GENCOM       ;CLEAR IBUFF
6843  022162  002007                  2007
6844  022164  001000                  1000
6845
6846  022166  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
6847  022172  000121                  RDDATA                  ;RDDATA
6848  022174  177000                  -1000                   ;-1000 WORDS
6849  022176  061414                  IBUFF                   ;IBUFF IS BUFF ADDRESS
6850  022200     000                  .BYTE   0               ;SECTOR 0
6851  022201     000                  .BYTE   0               ;TRACK 0
6852  022202  000312                  312                     ;CYLINDER 312
6853
6854  022204  104417                  TLOADRK                 ;LOAD RK REGS
6855  022206  104424                  TWAT32                  ;WAIT FOR INTERRUPT
6856  022210  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6857
6858  022212  104422                  TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
6859  022214  000000                  0
6860  022216  000100                  BSERR                   ;BAD SECTOR ERROR
6861  022220  000000                  0
6862  022222  104004                  ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
6863
6864  022224  004437  041740          JSR     R4,GENCOM       ;COMPARE DATA AGAIN
6865  022230  100000                  100000
6866  022232  000400                  400                     ;400 WORDS
6867  022234  000413                  BR      5$              ;NO MISCOMPARES
6868  022236  104015                  ERROR   15              ;REPORT FIRST ERROR
6869
```

```
6870  022240  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
6871  022244  005300                  66$:    DEC     R0              ;DECREMENT COUNT
6872  022246  001406                          BEQ     67$             ;IF ZERO - EXIT
6873  022250  004437  041740                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6874  022254  040000                          40000
6875  022256  000402                          BR      67$             ;NO MORE ERRORS - EXIT
6876  022260  104016                          ERROR   16              ;REPORT NEXT ERROR
6877  022262  000770                          BR      66$             ;LOOP
6878  022264                          67$:
6879
6880  022264                          5$:
6881                                  ;;*****************************************************************
6882                                  ;*TEST 66       FORMAT TFST
6883                                  ;*
6884                                  ;*      FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR FORMAT.
6885                                  ;*      MAKE SURE NO ERRORS SET.  READ SECTORS 0-25 AND MAKE
6886                                  ;*      SURE DATA CHECK DOES NOT OCCUR.
6887                                  ;*
6888                                  ;;*****************************************************************
6889  022264  000004                  TST66:  SCOPE
6890  022266  012737  000001  001262          MOV     #1,$TIMES       ;;DO 1 ITERATION
6891  022274  005000                          CLR     R0              ;CLEAR TRACK COUNTER
6892  022276  012737  022304  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP
6893
6894  022304                          1$:
6895  022304  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6896  022306  104003                          ERROR   3               ;BAD INIT ERROR
6897
6898  022310  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
6899  022316  012737  000127  001600          MOV     #WRHEAD,L.CS1   ;LOAD WRITE HEADER
6900  022324  110037  001607                  MOVB    R0,L.DT         ;LOAD DESIRED TRACK FROM TRACK COUNTER
6901  022330  012737  063414  001604          MOV     #OBUFF,L.BA     ;LOAD BUS ADDRESS
6902  022336  012737  177676  001602          MOV     #-102,L.WC      ;       WORD COUNT
6903  022344  012737  000312  001614          MOV     #312,L.DCYL     ;       CYLINDER
6904
6905  022352  004437  041740                  JSR     R4,GENCOM       ;BUILD HEADER
6906  022356  001200                          1200                    ;WITH BSE FLAGGED
6907
6908  022360  104417                          TLOADRK                 ;LOAD RK REGS
6909  022362  104431                          TWAT112                 ;WAIT FOR INTERRUPT
6910  022364  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6911
6912  022366  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6913  022370  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
6914
6915  022372  104415                          SCOP1                   ;LOCAL LOOP TO 1$
6916
6917  022374  005700                          TST     R0              ;R0 AT ZERO?
6918  022376  001002                          BNE     2$              ;NO-EXIT
6919  022400  005200                          INC     R0              ;BUMP COUNTER
6920  022402  000740                          BR      1$              ;LOOP
6921  022404  005000                  2$:     CLR     R0              ;CLEAR SECTOR COUNTER
6922  022406  012737  022414  001110          MOV     #113$,$LPERR    ;SET LOCAL LOOP ON ERROR
6923
6924  022414                          113$:
6925  022414  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
```

```
                                        ERROR   3               ;BAD INIT ERROR
                                3$:
                                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        RDDATA                  ;RDDATA
                                        -400                    ;-400 WORDS
                                        IBUFF                   ;IBUFF IS BUFF ADDRESS
                                        .BYTE   0               ;SECTOR 0
                                        .BYTE   0               ;TRACK 0
                                        312                     ;CYLINDER 312

                                        MOVB    R0,L.DS         ;LOAD SECTOR COUNTER INTO DESIRED SECTOR

                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT32                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

                                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
                                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

                                        SCOP1                   ;LOCAL LOOP TO 3$ ON ERROR

                                        CMP     #24,R0          ;LAST SECTOR READ?
                                        BEQ     4$              ;YES-EXIT
                                        INC     R0              ;BUMP SECTOR COUNTER
                                        BR      3$              ;LOOP

                                4$:     CLR     REFMT           ;CLEAR REFORMAT SWITCH

                                .SBTTL  **WRITE CHECK TESTS

                                ;;*****************************************************************
                                ;*TEST 67       WRITE-CHECK WITH NO ERROR
                                ;*
                                ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN.
                                ;*      DO A WRITE-CHECK OF 400 WORDS.  MAKE SURE NO
                                ;*      ERROR OCCURS.
                                ;*
                                ;;*****************************************************************
                                TST67:  SCOPE
                                        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
                                        TSSINIT                 ;CLEAR SUBSYSTEM
                                        ERROR   3               ;BAD INIT ERROR

                                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        WRDATA                  ;WRDATA
                                        -400                    ;-400 WORDS
                                        OBUFF                   ;OBUFF IS BUFF ADDRESS
                                        .BYTE   0               ;SECTOR 0
                                        .BYTE   0               ;TRACK 0
                                        312                     ;CYLINDER 312

                                        JSR     R4,GENCOM       ;GENERATE DATA
                                        2                       ;PATTERN 2
                                        400                     ;400 WORDS

                                        TLOADRK                 ;LOAD RK REGS
```

```
 6982  022540  104430                        TWAT96                  ;WAIT FOR INTERRUPT
 6983  022542  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6984
 6985  022544  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 6986  022546  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 6987
 6988  022550  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
 6989  022554  000131                        WRTCHK                  ;WRTCHK
 6990  022556  177400                        -400                    ;-400 WORDS
 6991  022560  063414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
 6992  022562     000                        .BYTE   0               ;SECTOR 0
 6993  022563     000                        .BYTE   0               ;TRACK 0
 6994  022564  000312                        312                     ;CYLINDER 312
 6995
 6996  022566  104417                        TLOADRK                 ;LOAD RK REGS
 6997  022570  104424                        TWAT32                  ;WAIT FOR INTERRUPT
 6998  022572  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6999
 7000  022574  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7001  022576  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 7002                          ;;*****************************************************************
 7003                          ;*TEST 70      WRITE CHECK ERROR (PART 1)
 7004                          ;*
 7005                          ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES.
 7006                          ;*      WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME
 7007                          ;*      DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING
 7008                          ;*      CONFIGURATIONS:
 7009                          ;*
 7010                          ;*      000001  000020  000400  010000
 7011                          ;*      000002  000040  001000  020000
 7012                          ;*      000004  000100  002000  040000
 7013                          ;*      000010  000200  004000  100000
 7014                          ;*
 7015                          ;*      MAKE SURE WRITE CHECK ERROR SET FOR EACH
 7016                          ;*      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
 7017                          ;*      AND WORD COUNT IS CORRECT.
 7018                          ;*
 7019                          ;;*****************************************************************
 7020  022600  000004          ST70:   SCOPE
 7021  022602  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 7022  022610  012700  000001                MOV     #BIT0,R0        ;SET LO ORDER BIT IN R0 FOR
 7023                                                                ;CAUSING WRITE CHECK ERROR
 7024  022614  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
 7025  022616  104003                        ERROR   3               ;BAD INIT ERROR
 7026  022620  004437  041740                JSR     R4,GENCOM       ;GENERATE DATA, ALL 0'S
 7027  022624  000001                        1
 7028  022626  000400                        400
 7029
 7030  022630  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
 7031  022634  000123                        WRDATA                  ;WRDATA
 7032  022636  177400                        -400                    ;-400 WORDS
 7033  022640  063414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
 7034  022642     000                        .BYTE   0               ;SECTOR 0
 7035  022643     000                        .BYTE   0               ;TRACK 0
 7036  022644  000312                        312                     ;CYLINDER 312
 7037
```

```
7038  022646  104417                          TLOADRK                 ;LOAD RK REGS
7039  022650  104430                          TWAT96                  ;WAIT FOR INTERRUPT
7040  022652  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7041
7042  022654  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7043  022656  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7044
7045  022660  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7046  022664  000131                          WRTCHK                  ;WRTCHK
7047  022666  177400                          -400                    ;-400 WORDS
7048  022670  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7049  022672     000                          .BYTE   0               ;SECTOR 0
7050  022673     000                          .BYTE   0               ;TRACK 0
7051  022674  000312                          312                     ;CYLINDER 312
7052
7053  022676  104417                          TLOADRK                 ;LOAD RK REGS
7054  022700  104424                          TWAT32                  ;WAIT FOR INTERRUPT
7055  022702  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7056
7057  022704  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7058  022706  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7059
7060  022710  104415                          SCOP1                   ;LOCAL LOOP ON WRITE CHECK
7061  022712  012737  022720  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP
7062  022720  010037  063634          1$:     MOV     R0,OBUFF+220    ;CAUSE ERROR BIT IN BUFFER
7063  022724  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7064  022726  104003                          ERROR   3               ;BAD INIT ERROR
7065  022730  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7066  022734  000131                          WRTCHK                  ;WRTCHK
7067  022736  177400                          -400                    ;-400 WORDS
7068  022740  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7069  022742     000                          .BYTE   0               ;SECTOR 0
7070  022743     000                          .BYTE   0               ;TRACK 0
7071  022744  000312                          312                     ;CYLINDER 312
7072
7073  022746  104417                          TLOADRK                 ;LOAD RK REGS
7074  022750  104424                          TWAT32                  ;WAIT FOR INTERRUPT
7075  022752  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7076
7077  022754  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7078  022756  000000                          0
7079  022760  000004                          WCKERR                  ;WRITE CHECK ERROR
7080  022762  000000                          0
7081  022764  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
7082
7083  022766  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
7084
7085                          ;               NOTE:   THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF THREE
7086                          ;               VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE
7087                          ;               WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE
7088                          ;               NPR CYCLE, WC AND BA WILL BE TWO PAST WHERE THE ERROR ACTUALLY
7089                          ;               OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST
7090                          ;               NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS ARE ONE PAST
7091                          ;               THE ACTUAL WORD WHERE THE ERROR WAS. IN SOME CASES, WC AND BA
7092                          ;               CAN BE THREE PAST WHERE THE ERROR WAS AND STILL BE ACCEPTABLE.
7093
```

```
7094
7095  022770  023727  001544  063636          CMP     T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
7096  022776  001416                          BEQ     2$              ;YES-SKIP
7097  023000  101040                          BHI     6$              ;IF TO HI - SKIP
7098  023002  012737  054021  001450          MOV     #EM11,EM10N     ;"INCORRECT BA"
7099  023010  012737  063636  001202          MOV     #OBUFF+222,$REG10  ;GOOD VALUE
7100  023016  013737  001544  001204          MOV     T.BA,$REG11     ;BAD VALUE
7101  023024  012737  050203  060770          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7102  023032  104010                          ERROR   10
7103
7104  023034  023727  001542  177511  2$:     CMP     T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7105  023042  001461                          BEQ     3$              ;YES-SKIP
7106  023044  101037                          BHI     7$              ;IF HIGHER SKIP
7107  023046  012737  053774  001450          MOV     #EM10,EM10N     ;"INCORRECT WC"
7108  023054  012737  050203  060770          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7109  023062  012737  177511  001202          MOV     #-267,$REG10    ;GOOD VALUE
7110  023070  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7111  023076  104010                          ERROR   10
7112  023100  000442                          BR      3$              ;EXIT
7113
7114  023102  023727  001544  063642  6$:     CMP     T.BA,#OBUFF+226 ;TEST IF BA AT HI SIDE
7115  023110  101415                          BLOS    7$              ;YES - SKIP
7116  023112  012737  054021  001450          MOV     #EM11,EM10N     ;SET MESSAGE
7117  023120  012737  063642  001202          MOV     #OBUFF+226,$REG10 ;GOOD VALUE
7118  023126  013737  001544  001204          MOV     T.BA,$REG11     ;ERROR VALUE
7119  023134  012737  050203  060770          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7120  023142  104010                          ERROR   10
7121
7122  023144  023727  001542  177513  7$:     CMP     T.WC,#-265      ;TEST IF WORD COUNT AT HI SIDE
7123  023152  101415                          BLOS    3$              ;YES - SKIP
7124  023154  012737  053774  001450          MOV     #EM10,EM10N     ;SET MESSAGE
7125  023162  012737  050203  060770          MOV     #OPER41,DF010A  ;"WC ABORTED WITH WCE"
7126  023170  012737  177513  001202          MOV     #-265,$REG10    ;GOOD VALUE
7127  023176  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7128  023204  104010                          ERROR   10
7129
7130  023206  104415                  3$:     SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
7131
7132  023210  032700  100000                  BIT     #BIT15,R0       ;BIT 15 SET?
7133  023214  001002                          BNE     4$              ;YES-EXIT
7134  023216  006300                          ASL     R0              ;SHIFT ERROR BIT
7135  023220  000637                          BR      1$              ;LOOP
7136
7137  023222                          4$:
7138                                  ;*****************************************************************
7139                                  ;*TEST 71       WRITE CHECK ERROR (PART 2)
7140                                  ;*
7141                                  ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
7142                                  ;*      IN ALL WORDS.  WRITE CHECK CYLINDER 312, TRACK 0,
7143                                  ;*      SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
7144                                  ;*      ONE OF THE FOLLWOING CONFIGURATIONS:
7145                                  ;*
7146                                  ;*      177776  177757  177377  167777
7147                                  ;*      177775  177737  176777  157777
7148                                  ;*      177773  177677  175777  137777
7149                                  ;*      177767  177577  173777  077777
```

```
                                        ;*
                                        ;*      MAKE SURE WRITE CHECK ERROR SET FOR EACH
                                        ;*      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
                                        ;*      AND WORD COUNT IS CORRECT.
                                        ;*
                                        ;**************************************************************
023222  000004                  TST71:  SCOPE
023224  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
023232  012700  177776                  MOV     #177776,R0      ;LOAD R0 FOR CAUSING WRITE CHECK ERROR

023236  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
023240  104003                          ERROR   3               ;BAD INIT ERROR
023242  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
023246  000007                          7                       ;ALL 1'S
023250  000400                          400                     ;400 WORDS

023252  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
023256  000123                          WRDATA                  ;WRDATA
023260  177400                          -400                    ;-400 WORDS
023262  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
023264     000                          .BYTE   0               ;SECTOR 0
023265     000                          .BYTE   0               ;TRACK 0
023266  000312                          312                     ;CYLINDER 312

023270  104417                          TLOADRK                 ;LOAD RK REGS
023272  104430                          TWAT96                  ;WAIT FOR INTERRUPT
023274  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

023276  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
023300  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

023302  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
023306  000131                          WRTCHK                  ;WRTCHK
023310  177400                          -400                    ;-400 WORDS
023312  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
023314     000                          .BYTE   0               ;SECTOR 0
023315     000                          .BYTE   0               ;TRACK 0
023316  000312                          312                     ;CYLINDER 312

023320  104417                          TLOADRK                 ;LOAD RK REGS
023322  104424                          TWAT32                  ;WAIT FOR INTERRUPT
023324  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

023326  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
023330  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

023332  104415                          SCOP1           ;LOCAL LOOP TO START OF TEST
023334  012737  023342  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP

023342  010037  063634          1$:     MOV     R0,OBUFF+220    ;PUT WORD I' 'BUFF TO CAUSE WCE
023346  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
023350  104003                          ERROR   3               ;BAD INIT ERROR

023352  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
023356  000131                          WRTCHK                  ;WRTCHK
023360  177400                          -400                    ;-400 WORDS
```

```
7206  023362  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7207  023364     000                          .BYTE   0               ;SECTOR 0
7208  023365     000                          .BYTE   0               ;TRACK 0
7209  023366  000312                          312                     ;CYLINDER 312
7210
7211  023370  104417                          TLOADRK                 ;LOAD RK REGS
7212  023372  104424                          TWAT32                  ;WAIT FOR INTERRUPT
7213  023374  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7214
7215  023376  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7216  023400  000000                          0
7217  023402  000004                          WCKERR                  ;WRITE CHECK ERROR
7218  023404  000000                          0
7219  023406  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
7220
7221  023410  104415                          SCOP1                   ;LOCAL LOOP TO 1$
7222
7223                                ;         NOTE:   THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF THREE
7224                                ;         VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE
7225                                ;         WCE OCCURRED, IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE
7226                                ;         NPR CYCLE, WC AND BA WILL BE TWO PAST WHERE THE ERROR ACTUALLY
7227                                ;         OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST
7228                                ;         NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS ARE ONE PAST
7229                                ;         THE ACTUAL WORD WHERE THE ERROR WAS. IN SOME CASES, WC AND BA
7230                                ;         CAN BE THREE PAST WHERE THE ERROR WAS AND STILL BE ACCEPTABLE.
7231
7232
7233  023412  023727  001544  063536          CMP     T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
7234  023420  001416                          BEQ     2$              ;YES-SKIP
7235  023422  101040                          BHI     6$              ;IF TO HI - SKIP
7236  023424  012737  054021  001450          MOV     #EM11,EM10N     ;"INCORRECT BA"
7237  023432  012737  063636  001202          MOV     #OBUFF+222,$REG10  ;GOOD VALUE
7238  023440  013737  001544  001204          MOV     T.BA,$REG11     ;BAD VALUE
7239  023446  012737  050203  060770          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7240  023454  104010                          ERROR   10
7241
7242  023456  023727  001542  177511  2$:     CMP     T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7243  023464  001461                          BEQ     3$              ;YES-SKIP
7244  023466  101037                          BHI     7$              ;IF HIGHER - SKIP
7245  023470  012737  053774  001450          MOV     #EM10,EM10N     ;"INCORRECT WC"
7246  023476  012737  050203  060770          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7247  023504  012737  177511  001202          MOV     #-267,$REG10    ;GOOD VALUE
7248  023512  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7249  023520  104010                          ERROR   10
7250  023522  000442                          BR      3$              ;EXIT
7251
7252  023524  023727  001544  063642  6$:     CMP     T.BA,#OBUFF+226 ;TEST IF BA AT HI SIDE
7253  023532  101415                          BLOS    7$              ;YES - SKIP
7254  023534  012737  054021  001450          MOV     #EM11,EM10N     ;SET MESSAGE
7255  023542  012737  063642  001202          MOV     #OBUFF+226,$REG10 ;GOOD VALUE
7256  023550  013737  001544  001204          MOV     T.BA,$REG11     ;ERROR VALUE
7257  023556  012737  050203  060770          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7258  023564  104010                          ERROR   10
7259
7260  023566  023727  001542  177513  7$:     CMP     T.WC,#-265      ;TEST IF WORD COUNT AT HI SIDE
7261  023574  101415                          BLOS    3$              ;YES - SKIP
```

```
7262  023576  012737  053774  001450          MOV     #EM10,EMION     ;SET MESSAGE
7263  023604  012737  050203  060770          MOV     #OPER41,DF010A  ;"WC ABORTED WITH WCE"
7264  023612  012737  177513  001202          MOV     #-265,$REG10    ;GOOD VALUE
7265  023620  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7266  023626  104010                          ERROR   10
7267
7268  023630  104415                  3$:     SCOP1
7269
7270  023632  032700  100000                  BIT     #BIT15,R0       ;BIT 15 SET? (ALL PATTERNS TESTED)
7271  023636  001002                          BNE     4$              ;YES-EXIT
7272  023640  006300                          ASL     R0              ;SHIFT FOR NEXT TEST
7273  023642  000637                          BR      1$              ;LOOP
7274
7275  023644                          4$:
7276                                  ;;*****************************************************************
7277                                  ;*TEST 72       WRITE CHECK OF PARTIAL SECTOR
7278                                  ;*
7279                                  ;*      WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN
7280                                  ;*      CONFIGURATIONS.  ISSUE A WRITE CHECK COMMAND OF
7281                                  ;*      110 WORDS MAKING SURE THE 111TH WORD IS
7282                                  ;*      DIFFERENT THAN DATA ON DISK.  MAKE SURE
7283                                  ;*      WRITE CHECK ERROR DOES NOT SET.
7284                                  ;*
7285                                  ;;*****************************************************************
7286  023644  000004                  TST72:  SCOPE
7287  023646  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
7288  023654  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7289  023656  104003                          ERROR   3               ;BAD INIT ERROR
7290
7291  023660  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
7292  023664  000007                          7                       ;ALL 1'S
7293  023666  000400                          400                     ;400 WORDS
7294
7295  023670  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7296  023674  000123                          WRDATA                  ;WRDATA
7297  023676  177400                          -400                    ;-400 WORDS
7298  023700  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7299  023702     000                          .BYTE   0               ;SECTOR 0
7300  023703     000                          .BYTE   0               ;TRACK 0
7301  023704  000312                          312                     ;CYLINDER 312
7302
7303  023706  104417                          TLOADRK                 ;LOAD RK REGS
7304  023710  104430                          TWAT96                  ;WAIT FOR INTERRUPT
7305  023712  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7306
7307  023714  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7308  023716  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7309
7310  023720  005037  063636                  CLR     OBUFF+222
7311
7312  023724  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7313  023730  000131                          WRTCHK                  ;WRTCHK
7314  023732  177670                          -110                    ;-110 WORDS
7315  023734  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7316  023736     000                          .BYTE   0               ;SECTOR 0
7317  023737     000                          .BYTE   0               ;TRACK 0
```

```
023740  000312                                   312                          ;CYLINDER 312

023742  104417                                   TLOADRK                      ;LOAD RK REGS
023744  104424                                   TWAT32                       ;WAIT FOR INTERRUPT
023746  104002                                   ERROR   2                    ;TO SLOW/NOT COMPLETE ERROR

023750  104421                                   TCHKOP                       ;CHECK OPERATION FOR ANY ERRORS
023752  104004                                   ERROR   4  ;OR 5, 6, 7, 10  ;REPORT ALL ERRORS

                                         .SBTTL  **MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT

                                         ;;*************************************************************
                                         ;*TEST 73        MAXIMUM DATA TRANSFER (PART 1)
                                         ;*
                                         ;*       IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF
                                         ;*       THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO
                                         ;*       INSURE THE FORMAT IS CORRECT.
                                         ;*
                                         ;*       ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH
                                         ;*       ONE SECTOR WRITES.  ISSUE A SEEK TO CYLINDER 0, TRACK 0.
                                         ;*       ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 200000 WORDS
                                         ;*       TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE CONTROLLER
                                         ;*       TIME OUT IS NOT SET.  CHECK CYLINDER ADDRESS,
                                         ;*       DISK ADDRESS, BUS ADDRESS AND WORD COUNT.  READ
                                         ;*       EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.
                                         ;*
                                         ;*       NOTE:   THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
                                         ;*               IN THE FIRST 256 SECTORS ON THE PACK.
                                         ;*
                                         ;;*************************************************************
023754  000004                           TST73:  SCOPE
023756  012737  000002  001262                   MOV     #2,$TIMES            ;;DO 2 ITERATIONS
023764  032737  000400  001664                   BIT     #FPFMT,OPTFLG        ;TEST IF FIRST PASS SWITCH SET
023772  001043                                   BNE     24$                  ;YES - SKIP FORMAT

023774  105037  024017                           CLRB    21$                  ;CLEAR ADDRESS POINTERS
024000  005037  024020                           CLR     22$

024004  004437  035530                   20$:    JSR     R4,LRLOAD            ;LOAD "L" REGISTERS
024010  000127                                   WRHEAD                       ;WRITE HEADER
024012  177676                                   -102                         ;102 WORDS
024014  063414                                   OBUFF                        ;OBUFF IS BUFF ADDRESS
024016     000                                   .BYTE   0                    ;SECTOR 0
024017     000                           21$:    .BYTE   0                    ;TRACK ADDRESS (VARIABLE)
024020  000000                           22$:    0                            ;CYLINDER 0 (VARIABLE)

024022  004437  041740                           JSR     R4,GENCOM            ;GO GENERATE HEADERS
024026  001200                                   1200                         ;WITH BAD SECTOR ERRORS

024030  104417                                   TLOADRK                      ;LOAD RK REGS
024032  104431                                   TWAT112                      ;WAIT FOR INTERRUPT
024034  104002                                   ERROR   2                    ;TO SLOW/NOT COMPLETE ERROR

024036  104421                                   TCHKOP                       ;CHECK OPERATION FOR ANY ERRORS
024040  104004                                   ERROR   4  ;OR 5, 6, 7       ;REPORT ALL ERRORS
```

```
7374  024042  123727  024017  000002          CMPB    21$,#2          ;TEST IF LAST TRACK
7375  024050  001403                          BEQ     23$             ;YES - SKIP
7376  024052  105237  024017                  INCB    21$             ;ELSE BUMP TRACK
7377  024056  000752                          BR      20$             ;LOOP
7378
7379  024060  105037  024017          23$:    CLRB    21$             ;CLEAR TRACK POINTER
7380  024064  023727  024020  000003          CMP     22$,#3          ;TEST IF LAST CYLINDER WRITTEN
7381  024072  001403                          BEQ     24$             ;YES - SKIP
7382  024074  005237  024020                  INC     22$             ;ELSE BUMP CYLINDER
7383  024100  000741                          BR      20$             ;LOOP
7384
7385  024102  013737  024114  001110  24$:    MOV     1$,$LPERR       ;SET LOCAL LOOP ON ERROR
7386  024110  012703  000400                  MOV     #400,R3         ;SET COUNT FOR SECTOR CLEARING
7387  024114                          1$:
7388  024114  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7389  024116  104003                          ERROR   3               ;BAD INIT ERROR
7390  024120  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7391  024124  000123                          WRDATA                  ;WRDATA
7392  024126  177400                          -400                    ;-400 WORDS
7393  024130  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7394  024132     000                          .BYTE   0               ;SECTOR 0
7395  024133     000                          .BYTE   0               ;TRACK 0
7396  024134  000000                          0                       ;CYLINDER 0
7397
7398  024136  104417                          TLOADRK                 ;LOAD RK REGS
7399  024140  104434                          TWAT159                 ;WAIT FOR INTERRUPT
7400  024142  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7401
7402  024144  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7403  024146  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7404
7405  024150  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
7406  024152  005303                          DEC     R3              ;DECREMENT COUNT
7407  024 54  012762  063414  000004  2$:     MOV     #OBUFF,RKBA(R2) ;SET BA
7408  024162  012762  177400  000002          MOV     #-400,RKWC(R2)  ;AND WC AGAIN
7409  024170  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
7410  024174  013762  001626  000010          MOV     DRVNUM,RKCS2(R2)        ;SET DRIVE NUMBER
7411  024202  012762  000123  000000          MOV     #WRDATA,RKCS1(R2) ;DO WRITE DATA
7412
7413  024210  104425                          TWAT48                  ;WAIT FOR INTERRUPT
7414  024212  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7415
7416  024214  032762  000200  000014          BIT     #BSE,RKER(R2)   ;BAD SECTOR ERROR?
7417  024222  001415                          BEQ     3$              ;NO-SKIP
7418  024224  032737  000200  001664          BIT     #BSERPT,OPTFLG  ;TEST IF BSE TO MANY HAS BEEN REPORTED
7419  024232  001007                          BNE     5$              ;YES - SKIP
7420  024234  052737  000200  001664          BIS     #BSERPT,OPTFLG  ;SET FLAG
7421  024242  012737  052445  001360          MOV     #OPR017,EMIN    ;SET MESSAGE
7422  024250  104001                          ERROR   1               ;"FIRST 256 SECTOR NOT BSE FREE"
7423  024252  000137  024606          5$:     JMP     14$             ;GO TO EXIT
7424                                  ;
                                      ;CHECK FOR ANY OTHER ERRORS
7425  024256                          3$:
7426  024256  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7427  024260  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7428  024262  104415                          SCOP1                   ;LOCAL LOOP TO 1$ (RESTART SECTOR CLEAR)
7429
```

```
7430  024264  005303                          DEC     R3              ;DECREMENT COUNT
7431  024266  001332                          BNE     2$              ;LOOP IF NOT ZERO
7432
7433  024270  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7434  024274  000117                          SEEK                    ;SEEK
7435  024276  000000                          0                       ;0 WORDS
7436  024300  000000                          0                       ;0 IS BUFF ADDRESS
7437  024302     000                          .BYTE   0               ;SECTOR 0
7438  024303     000                          .BYTE   0               ;TRACK 0
7439  024304  000000                          0                       ;CYLINDER 0
7440
7441  024306  104417                          TLOADRK                 ;LOAD RK REGS
7442  024310  104423                          TWAT16                  ;WAIT FOR INTERRUPT
7443  024312  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7444  024314  005037  001662                  CLR     INTSET          ;CLEAR FIRST INTERRUPT
7445  024320  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7446  024322  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7447  024324  104427                          TWAT80                  ;WAIT FOR SECOND INTERRUPT
7448  024326  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7449  024330  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA
7450  024334  004006                          4006                    ;PATTERN 6, 1ST WORD REPEATED
7451  024336  000400                          400                     ;400 WORDS
7452
7453  024340                          4$:
7454  024340  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7455  024342  104003                          ERROR   3               ;BAD INIT ERROR
7456  024344  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7457  024350  000123                          WRDATA                  ;WRDATA
7458  024352  000000                          0                       ;0 WORDS
7459  024354  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7460  024356     000                          .BYTE   0               ;SECTOR 0
7461  024357     000                          .BYTE   0               ;TRACK 0
7462  024360  000000                          0                       ;CYLINDER 0
7463
7464  024362  052737  000020  001610          BIS     #BAI,L.CS2
7465
7466  024370  104417                          TLOADRK                 ;LOAD RK REGS
7467  024372  104437                          TWAT8S                  ;WAIT FOR SECOND INTERRUPT
7468  024374  104002                          ERROR   2               ;ELSE REPORT TO SLOW/NOT COMPLETE ERROR
7469
7470  024376                          7$:
7471  024376  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7472  024400  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7473
7474  024402  104415                          SCOP1                   ;INTERNAL LOOP ON ERROR TO 4$
7475  024404  012703  000400                  MOV     #400,R3         ;SET COUNTER FOR READ-COMPARE LOOP
7476  024410  005037  024452                  CLR     10$             ;CLEAR SECTOR AND TRACK VALUES
7477  024414  005037  024454                  CLR     12$             ;CLEAR CYL VALUE
7478  024420  013737  024434  001110          MOV     8$,$LPERR       ;SET LOCAL LOOP ON ERROR
7479  024426  042737  000020  001610          BIC     #BAI,L.CS2      ;CLEAR BAI
7480
7481  024434                          8$:
7482  024434  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7483  024436  104003                          ERROR   3               ;BAD INIT ERROR
7484  024440  004437  035530          9$:     JSR     R4,LRLOAD       ;LOAD RK REGS
7485  024444  000121                          RDDATA                  ;READ DATA
```

```
7486  024446  177400                        -400                    ;400 WORDS
7487  024450  061414                        IBUFF                   ;INTO IBUFF
7488  024452  000                   10$:    .BYTE   0               ;SECTOR (VARIABLE)
7489  024453  000                   11$:    .BYTE   0               ;TRACK  (VARIABLE)
7490  024454  000000                12$:    0                       ;CYL    (VARIABLE)
7491
7492  024456  104417                        TLOADRK                 ;LOAD RK REGS
7493  024460  104425                        TWAT48                  ;WAIT FOR INTERRUPT
7494  024462  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7495
7496  024464  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7497  024466  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7498
7499  024470  104415                        SCOP1                   ;LOCAL LOOP ON ERROR TO 8$
7500
7501  024472  004437  041740                JSR     R4,GENCOM       ;COMPARE DATA
7502  024476  100000                        100000
7503  024500  000400                        400                     ;400 WORDS
7504  024502  000413                        BR      13$             ;NO MISCOMPARE-EXIT LOOP
7505  024504  104015                        ERROR   15              ;REPORT FIRST ERROR
7506
7507  024506  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
7508  024512  005300                64$:    DEC     R0              ;DECREMENT COUNT
7509  024514  001406                        BEQ     65$             ;IF ZERO - EXIT
7510  024516  004437  041740                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
7511  024522  040000                        40000
7512  024524  000402                        BR      65$             ;NO MORE ERRORS - EXIT
7513  024526  104016                        ERROR   16              ;REPORT NEXT ERROR
7514  024530  000770                        BR      64$             ;LOOP
7515  024532                        65$:
7516
7517  024532  104415                13$:    SCOP1                   ;LOCAL LOOP TO 8$
7518
7519  024534  005303                        DEC     R3              ;DEC READ LOOP COUNT
7520  024536  001423                        BEQ     14$             ;IF ZERO-EXIT
7521
7522  024540  105237  024452                INCB    10$             ;BUMP SECTOR
7523  024544  123727  024452  000026        CMPB    10$,#26         ;FINISHED WITH TRACK?
7524  024552  001332                        BNE     9$              ;NO-LOOP
7525  024554  105037  024452                CLRB    10$             ;CLEAR SECTOR
7526  024560  105237  024453                INCB    11$             ;BUMP TRACK
7527  024564  123727  024453  000003        CMPB    11$,#3          ;FINISHED WITH CYLINDER?
7528  024572  001322                        BNE     9$              ;NO-LOOP
7529  024574  105037  024453                CLRB    11$             ;CLEAR TRACK
7530  024600  005237  024454                INC     12$             ;BUMP CYL.
7531  024604  000715                        BR      9$              ;LOOP
7532
7533  024606                        14$:
7534                                ;;*****************************************************************
7535                                ;*TEST 74       MAXIMUM DATA TRANSFER (PART 2)
7536                                ;*
7537                                ;*      ZERO OUT FIRST 256 SECTORS OF THE DISK WITH
7538                                ;*      200000 WORD WRITE.  SEEK TO CYLINDER 632.
7539                                ;*      ISSUE A WRITE OF MAXIMUM DATA TRANSFER
7540                                ;*      200000 WORD WRITE.  MAKE SURE CONTROLLER TIME
7541                                ,*      OUT IS NOT SET.  CHECK CYLINDER ADDRESS
```

```
 7542                                   ;*      DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
 7543                                   ;*      SEEK TO CYLINDER 632.  ISSUE A WRITE CHECK
 7544                                   ;*      OF 200000 WORDS.  MAKE SURE NO ERROR SETS.
 7545                                   ;*
 7546                                   ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
 7547                                   ;*              IN THE FIRST 256 SECTORS ON THE PACK.
 7548                                   ;*
 7549                                   ;;*********************************************************************
 7550  024606  000004                   TST74:  SCOPE
 7551  024610  012737  000002  001262           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
 7552  024616  104416                           TSSINIT                 ;CLEAR SUBSYSTEM
 7553  024620  104003                           ERROR   3               ;BAD INIT ERROR
 7554  024622  012700  000620                   MOV     #400.,R0        ;SET COUNT FOR INTERRUPT WAIT
 7555  024626  005037  063414                   CLR     OBUFF
 7556
 7557  024632  004437  035530                   JSR     R4,LRLOAD       ;LOAD "L" REGS
 7558  024636  000123                           WRDATA                  ;WRDATA
 7559  024640  000000                           0                       ;0 WORDS
 7560  024642  063414                           OBUFF                   ;OBUFF IS BUFF ADDRESS
 7561  024644     000                           .BYTE   0               ;SECTOR 0
 7562  024645     000                           .BYTE   0               ;TRACK 0
 7563  024646  000000                           0                       ;CYLINDER 0
 7564  024650  052737  000020  001610           BIS     #BAI,L.CS2      ;SET BAI
 7565  024656  104417                           TLOADRK                 ;LOAD RK REGS
 7566  024660  104434                   1$:     TWAT159                 ;WAIT FOR INTERRUPT
 7567  024662  000401                           BR      2$              ;NO INTERRUPT-SKIP
 7568  024664  000403                           BR      3$              ;INTERRUPT-SKIP
 7569
 7570  024666  005300                   2$:     DEC     R0              ;DEC WAIT COUNTER
 7571  024670  001373                           BNE     1$              ;NO ZERO-LOOP
 7572  024672  104002                           ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7573
 7574  024674  032762  000200  000014   3$:     BIT     #BSE,RKER(R2)   ;DID BSE SET
 7575  024702  001415                           BEQ     4$              ;NO-SKIP
 7576
 7577  024704  032737  000200  001664           BIT     #BSERPT,OPTFLG  ;TEST IF TO MANY BAD SECTORS REPORTED
 7578  024712  001007                           BNE     12$             ;YES - SKIP
 7579  024714  052737  000200  001664           BIS     #BSERPT,OPTFLG  ;SET FLAG
 7580  024722  012737  052445  001360           MOV     #OPR017,EMIN    ;SET MESSAGE
 7581  024730  104001                           ERROR   1               ;"FIRST 256 SECTORS NOT BSE FREE"
 7582  024732  000137  025264           12$:    JMP     11$             ;EXIT
 7583
 7584  024736                           4$:
 7585  024736  104421                           TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7586  024740  104004                           ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 7587
 7588  024742  004437  035530                   JSR     R4,LRLOAD       ;LOAD "L" REGS
 7589  024746  000117                           SEEK                    ;SEEK
 7590  024750  000000                           0                       ;0 WORDS
 7591  024752  000000                           0                       ;0 IS BUFF ADDRESS
 7592  024754     000                           .BYTE   0               ;SECTOR 0
 7593  024755     000                           .BYTE   0               ;TRACK 0
 7594  024756  000632                           632                     ;CYLINDER 632
 7595
 7596  024760  104417                           TLOADRK                 ;LOAD RK REGS
 7597  024762  104423                           TWAT16                  ;WAIT FOR INTERRUPT
```

```
7598  024764  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7599  024766  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG
7600
7601  024772  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7602  024774  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7603
7604  024776  104427                        TWAT80                  ;WAIT FOR 2ND INTERRUPT
7605  025000  104002                        ERROR   2
7606
7607  025002  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7608  025004  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7609
7610  025006  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
7611  025012  000105                        CLEAR                   ;CLEAR
7612  025014  000000                        0                       ;0 WORDS
7613  025016  000000                        0                       ;0 IS BUFF ADDRESS
7614  025020     000                        .BYTE   0               ;SECTOR 0
7615  025021     000                        .BYTE   0               ;TRACK 0
7616  025022  000000                        0                       ;CYLINDER 0
7617
7618  025024  104417                        TLOADRK                 ;LOAD RK REGS
7619  025026  104423                        TWAT16                  ;WAIT FOR INTERRUPT
7620  025030  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7621
7622  025032  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7623  025034  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7624
7625  025036  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
7626  025042  000123                        WRDATA                  ;WRDATA
7627  025044  000000                        0                       ;0 WORDS
7628  025046  063414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
7629  025050     000                        .BYTE   0               ;SECTOR 0
7630  025051     000                        .BYTE   0               ;TRACK 0
7631  025052  000000                        0                       ;CYLINDER 0
7632
7633  025054  012737  135143  063414        MOV     #135143,OBUFF   ;SET WORD FOR OUTPUT
7634  025062  012700  000621                MOV     #401.,R0        ;SET COUNT FOR INTERRUPT WAIT
7635  025066  052737  000020  001610        BIS     #BAI,L.CS2      ;SET BUS ADDRESS INC INHIBIT
7636
7637  025074  104417                        TLOADRK                 ;LOAD RK REGS
7638  025076  104434                5$:     TWAT159                 ;WAIT FOR INTERRUPT
7639  025100  000401                        BR      6$              ;NO INTERRUPT-BRANCH
7640  025102  000403                        BR      7$              ;INTERRUPT-BRANCH
7641
7642  025104  005300                6$:     DEC     R0              ;DEC WAIT COUNT
7643  025106  001373                        BNE     5$              ;LOOP IF NOT ZERO
7644  025110  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7645
7646  025112                        7$:
7647  025112  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7648  025114  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7649  025116  004437  035530                JSR     R4,LRLOAD       ;LOAD "L" REGS
7650  025122  000117                        SEEK                    ;SEEK
7651  025124  000000                        0                       ;0 WORDS
7652  025126  000000                        0                       ;0 IS BUFF ADDRESS
7653  025130     000                        .BYTE   0               ;SECTOR 0
```

```
7654  025131  000                       .BYTE   0               ;TRACK 0
7655  025132  000632                    632                     ;CYLINDER 632
7656  025134  104417                    TLOADRK                 ;LOAD RK REGS
7657  025136  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7658  025140  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7659  025142  005037  001662            CLR     INTSET          ;CLEAR INTERRUPT FLAG
7660
7661  025146  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7662  025150  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7663
7664  025152  104427                    TWATBO                  ;WAIT FOR SECOND INIT
7665  025154  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7666  025156  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7667  025160  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7668
7669  025162  004437  035530            JSR     R4,LRLOAD       ;LOAD "L" REGS
7670  025166  000105                    CLEAR                   ;CLEAR
7671  025170  000000                    0                       ;0 WORDS
7672  025172  000000                    0                       ;0 IS BUFF ADDRESS
7673  025174  000                       .BYTE   0               ;SECTOR 0
7674  025175  000                       .BYTE   0               ;TRACK 0
7675  025176  000000                    0                       ;CYLINDER 0
7676
7677  025200  104417                    TLOADRK                 ;LOAD RK REGS
7678  025202  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7679  025204  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7680
7681  025206  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7682  025210  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7683
7684  025212  004437  035530            JSR     R4,LRLOAD       ;LOAD "L" REGS
7685  025216  000131                    WRTCHK                  ;WRTCHK
7686  025220  000000                    0                       ;0 WORDS
7687  025222  063414                    OBUFF                   ;OBUFF IS BUFF ADDRESS
7688  025224  000                       .BYTE   0               ;SECTOR 0
7689  025225  000                       .BYTE   0               ;TRACK 0
7690  025226  000000                    0                       ;CYLINDER 0
7691  025230  052737  000020  001610    BIS     #BAI,L.CS2      ;SET BAI FLAG
7692  025236  012700  000621            MOV     #401.,R0        ;SET WAIT COUNT
7693
7694  025242  104417                    TLOADRK                 ;LOAD RK REGS
7695  025244  104434            8$:     TWAT159                 ;WAIT FOR INTERRUPT
7696  025246  000401                    BR      9$              ;NO INTERRUPT-SKIP
7697  025250  000403                    BR      10$             ;INTERRUPT-SKIP
7698
7699  025252  005300            9$:     DEC     R0              ;DEC WAIT COUNT
7700  025254  001373                    BNE     8$              ;NOT ZERO-LOOP
7701  025256  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7702
7703  025260                    10$:
7704  025260  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7705  025262  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7706
7707  025264                    11$:
7708                            ;;*****************************************************************
7709                            ;*TEST 75        CONTROLLER TIME OUT
```

```
                                        ;*
                                        ;*      SEEK TO CYLINDER 632.  ISSUE A RECALIBRATE AND DO NOT
                                        ;*      WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER
                                        ;*      OF CYLINDER 0, TRACK 0.  MAKE SURE CONTROLLER TIME
                                        ;*      OUT SETS.
                                        ;*
                                        ;**************************************************************
025264  000004                  TST75:  SCOPE
025266  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
025274  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
025276  104003                          ERROR   3               ;BAD INIT ERROR

025300  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
025304  000117                          SEEK                    ;SEEK
025306  000000                          0                       ;0 WORDS
025310  000000                          0                       ;0 IS BUFF ADDRESS
025312     000                          .BYTE   0               ;SECTOR 0
025313     000                          .BYTE   0               ;TRACK 0
025314  000632                          632                     ;CYLINDER 632

025316  104417                          TLOADRK                 ;LOAD RK REGS
025320  104423                          TWP^16                  ;WAIT FOR INTERRUPT
025322  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

025324  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
025326  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

025330  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
025334  104427                          TWAT80                  ;WAIT FOR SECOND INTERRUPT
025336  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
025340  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
025342  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

025344  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
025350  000105                          CLEAR                   ;CLEAR
025352  000000                          0                       ;0 WORDS
025354  000000                          0                       ;0 IS BUFF ADDRESS
025356     000                          .BYTE   0               ;SECTOR 0
025357     000                          .BYTE   0               ;TRACK 0
025360  000000                          0                       ;CYLINDER 0

025362  104417                          TLOADRK                 ;LOAD RK REGS
025364  104423                          TWAT16                  ;WAIT FOR INTERRUPT
025366  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

025370  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
025372  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

025374  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
025400  000113                          RECAL                   ;RECAL
025402  000000                          0                       ;0 WORDS
025404  000000                          0                       ;0 IS BUFF ADDRESS
025406     000                          .BYTE   0               ;SECTOR 0
025407     000                          .BYTE   0               ;TRACK 0
025410  000000                          0                       ;CYLINDER 0
```

```
7766  025412  104417                          TLOADRK                 ;LOAD RK REGS
7767  025414  104423                          TWAT16                  ;WAIT FOR INTERRUPT
7768  025416  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7769
7770  025420  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7771  025424  000125                          RDHEAD                  ;RDHEAD
7772  025426  000000                          0                       ;0 WORDS
7773  025430  000000                          0                       ;0 IS BUFF ADDRESS
7774  025432     000                          .BYTE   0               ;SECTOR 0
7775  025433     000                          .BYTE   0               ;TRACK 0
7776  025434  000000                          0                       ;CYLINDER 0
7777
7778  025436  104417                          TLOADRK                 ;LOAD RK REGS
7779  025440  104436                          TWAT25                  ;WAIT FOR INTERRUPT
7780  025442  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7781
7782  025444  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7783  025446  000000                          0
7784  025450  000000                          0
7785  025452  000002                          CTOERR                  ;CONTROLLER TIME OUT
7786  025454  104004                          ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
7787  025456  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7788  025460  104003                          ERROR   3               ;BAD INIT ERROR
7789  025462  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
7790  025466  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INT ENABLE
7791  025474  104437                          TWAT85                  ;WAIT FOR SECOND INT
7792  025476  104002                          ERROR   2
7793
7794
7795
7796                                  .SBTTL  **ERRORS DURING DATA TRANSFER
7797
7798
7799                                  ;;*****************************************************************
7800                                  ;*TEST 76       LIMIT DETECT ON DATA TRANSFER
7801                                  ;*
7802                                  ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.  ISSUE
7803                                  ;*      A SEEK TO CYLINDER 2 WITH BAD PARITY.  ISSUE A DRIVE
7804                                  ;*      CLEAR.  ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
7805                                  ;*      TRACK 0, HEAD 0.  SEEK INCOMPLETE BECAUSE OF OUTER
7806                                  ;;*     LIMIT SHOULD BE THE ONLY ERROR SET.
7807                                  ;;*****************************************************************
7808  025500  000004                  TST76:  SCOPE
7809  025502  012737  000003  001262          MOV     #3.,$TIMES      ;;DO 3. ITERATIONS
7810
7811  025510  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7812  025512  104003                          ERROR   3               ;BAD INIT ERROR
7813
7814  025514  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7815  025520  000113                          RECAL                   ;RECAL
7816  025522  000000                          0                       ;0 WORDS
7817  025524  000000                          0                       ;0 IS BUFF ADDRESS
7818  025526     000                          .BYTE   0               ;SECTOR 0
7819  025527     000                          .BYTE   0               ;TRACK 0
7820  025530  000000                          0                       ;CYLINDER 0
7821
```

```
 7822  025532  104417                          TLOADRK                 ;LOAD RK REGS
 7823  025534  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7824  025536  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7825
 7826  025540  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
 7827  025544  104437                          TWAT8S                  ;WAIT FOR SECOND INTERRUPT
 7828  025546  104002                          ERROR   2
 7829
 7830  025550  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7831  025552  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7832
 7833  025554  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7834  025560  000117                          SEEK                    ;SEEK
 7835  025562  000000                          0                       ;0 WORDS
 7836  025564  000000                          0                       ;0 IS BUFF ADDRESS
 7837  025566     000                          .BYTE   0               ;SECTOR 0
 7838  025567     000                          .BYTE   0               ;TRACK 0
 7839  025570  000002                          2                       ;CYLINDER 2
 7840  025572  012737  000020  001615          MOV     #PAT,L.MR1      ;SET EVEN PARITY BIT
 7841  025600  104417                          TLOADRK                 ;LOAD RK REGS
 7842  025602  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7843  025604  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7844
 7845  025606  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 7846  025610  104003                          ERROR   3               ;BAD INIT ERROR
 7847
 7848  025612  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7849  025616  000123                          WRDATA                  ;WRDATA
 7850  025620  177400                          -400                    ;-400 WORDS
 7851  025622  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 7852  025624     000                          .BYTE   0               ;SECTOR 0
 7853  025625     000                          .BYTE   0               ;TRACK 0
 7854  025626  000001                          1                       ;CYLINDER 1
 7855
 7856  025630  104417                          TLOADRK                 ;LOAD RK REGS
 7857  025632  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7858  025634  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7859
 7860  025636  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
 7861  025640  000002                          SKIERR                  ;SEEK INCOMPLETE
 7862  025642  000000                          0
 7863  025644  000000                          0
 7864  025646  104004                          ERROR   4  ;OR 5,6,OR7  ;REPORT ALL DISCREPANCIES
 7865
 7866  025650  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 7867  025652  104003                          ERROR   3               ;BAD INIT ERROR
 7868
 7869
 7870  025654                          3$:
 7871  025654  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7872  025660  000101                          SELDRV                  ;SELDRV
 7873  025662  000000                          0                       ;0 WORDS
 7874  025664  000000                          0                       ;0 IS BUFF ADDRESS
 7875  025666     000                          .BYTE   0               ;SECTOR 0
 7876  025667     000                          .BYTE   0               ;TRACK 0
 7877  025670  000000                          0                       ;CYLINDER 0
```

```

025672  012737  000001  001616          MOV     #1,L.MR1          ;SET TO GET STATUS PAIR 1
025700  104417                          TLOADRK                   ;LOAD RK REGS
025702  104423                          TWAT16                    ;WAIT FOR INTERRUPT
025704  104002                          ERROR   2                 ;TO SLOW/NOT COMPLETE ERROR

025706  104421                          TCHKOP                    ;CHECK OPERATION FOR ANY ERRORS
025710  104004                          ERROR   4  ;OR 5, 6, 7    ;REPORT ALL ERRORS

025712  032737  000040  001574          BIT     #S.HOHM,T.MR2     ;TEST IF HEADS HOME
025720  001755                          BEQ     3$                ;NO - GET STATUS AGAIN

025722  104416                          TSSINIT                   ;CLEAR SUBSYSTEM
025724  104003                          ERROR   3                 ;BAD INIT ERROR

025726  012762  000100  000000          MOV     #IE,RKCS1(R2)     ;SET IE

025734  005037  001662                  CLR     INTSET            ;CLEAR INT FLAG
025740  104437                          TWAT8S                    ;WAIT FOR SECOND INTERRUPT
025742  000401                          BR      1$
025744  000404                          BR      2$

025746  012737  060302  001372  1$:     MOV     #DH016,DH2N       ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
                                                                  ;ALLOWING HEADS TO RELOAD"
025754  104002                          ERROR   2

025756                          2$:

                                ;;*************************************************
                                ;*TEST 77          PROGRAMMING ERROR
                                ;*        ISSUE A SUBSYSTEM CLEAR. ISSUE
                                ;*        A READ DATA OF 400 WORDS ON CYLINDER 0,
                                ;*        TRACK 0, SECTOR 0. DURING READ ISSUE A
                                ;*        WRITE TO THE SPARE REGISTER. MAKE SURE
                                ;*        PROGRAMMING ERROR SETS.
                                ;*************************************************
025756  000004                  TST77:  SCOPE
025760  012737  000012  001262          MOV     #10.,$TIMES       ;;DO 10. ITERATIONS
025766  104416                          TSSINIT                   ;CLEAR SUBSYSTEM
025770  104003                          ERROR   3                 ;BAD INIT ERROR

025772  004437  035530                  JSR     R4,LRLOAD         ;LOAD "L" REGS
025776  000121                          RDDATA                    ;RDDATA
026000  177400                          -400                      ;-400 WORDS
026002  061414                          IBUFF                     ;IBUFF IS BUFF ADDRESS
026004     000                          .BYTE   0                 ;SECTOR 0
026005     000                          .BYTE   0                 ;TRACK 0
026006  000000                          0                         ;CYLINDER 0

026010  104417                          TLOADRK          ;LOAD RK REGS

026012  012762  000001  000022          MOV     #1,RKSPAR(R2)     ;WRITE SPARE REGISTER

026020  104423                          TWAT16           ;WAIT FOR INTERRUPT
026022  104002                          ERROR   2                 ;TO SLOW/NOT COMPLETE ERROR

```

```
7934  026024  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7935  026026  000000                          0
7936  026030  000000                          0
7937  026032  000020                          PGERR                   ;PROG ERROR
7938  026034  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL DISCREPANCIES
7939
7940
7941                                  ;;*****************************************************************
7942                                  ;*TEST 100       ECC HARD
7943                                  ;*       ISSUE A SUBSYSTEM CLEAR. ISSUE
7944                                  ;*       A WRITE DATA WORDS CONSISTING OF 177777 TO
7945                                  ;*       CYLINDER 0, TRACK 0, SECTOR 0. NOW WRITE
7946                                  ;*       ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR 0.
7947                                  ;*       DURING WRITE ISSUE CONTROLLER CLEAR. MAKE
7948                                  ;*       SURE PROGRAMMING ERROR IS RESET. NOW
7949                                  ;*       ISSUE A READ DATA TO CYLINDER 0, TRACK 0,
7950                                  ;*       HEAD 0 AND AN ECC HARD ERROR SHOULD SET.
7951                                  ;;*****************************************************************
7952  026036  000004                  TST100: SCOPE
7953  026040  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
7954  026046  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7955  026050  104003                          ERROR   3               ;BAD INIT ERROR
7956
7957  026052  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA OF ALL ONES
7958  026056  000001                          1
7959  026060  000400                          400
7960
7961  026062  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7962  026066  000123                          WRDATA                  ;WRDATA
7963  026070  177400                          -400                    ;-400 WORDS
7964  026072  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7965  026074     000                          .BYTE   0               ;SECTOR 0
7966  026075     000                          .BYTE   0               ;TRACK 0
7967  026076  000000                          0                       ;CYLINDER 0
7968
7969  026100  104417                          TLOADRK                 ;LOAD RK REGS
7970  026102  104430                          TWAT96                  ;WAIT FOR INTERRUPT
7971  026104  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7972
7973  026106  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7974  026110  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7975
7976  026112  004437  041740                  JSR     R4,GENCOM       ;GENERATE DATA OF ZEROS
7977  026116  000002                          2
7978  026120  000400                          400
7979
7980  026122  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7981  026126  000123                          WRDATA                  ;WRDATA
7982  026130  177630                          -150                    ;-150 WORDS
7983  026132  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7984  026134     000                          .BYTE   0               ;SECTOR 0
7985  026135     000                          .BYTE   0               ;TRACK 0
7986  026136  000000                          0                       ;CYLINDER 0
7987
7988  026140  104417                          TLOADRK                 ;START OPERATION
7989
```

```
7990  026142  005737  001662          1$:     TST     INTSET          ;CHECK IF INTERRUPT HAS OCCURRED
7991  026146  001026                          BNE     2$              ;YES - MUCH TO SOON. REPORT ERROR
7992  026150  005762  000002                  TST     RKWC(R2)        ;TEST IF NPR'S DONE
7993  026154  001372                          BNE     1$              ;NO - LOOP
7994
7995  026156  052762  100000  000000          BIS     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER (CROWBAR WRITE)
7996
7997  026164  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7998  026170  000121                          RDDATA                  ;RDDATA
7999  026172  177400                          -400                    ;-400 WORDS
8000  026174  061414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
8001  026176     000                          .BYTE   0               ;SECTOR 0
8002  026177     000                          .BYTE   0               ;TRACK 0
8003  026200  000000                          0                       ;CYLINDER 0
8004
8005  026202  104417                          TLOADRK                 ;LOAD RK REGS
8006  026204  104425                          TWAT48                  ;WAIT FOR INTERRUPT
8007  026206  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8008
8009  026210  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
8010  026212  000000                          0
8011  026214  000003                          DCKERR!ECHERR           ;DATA CHECK AND ECC HARD
8012  026216  000000                          0
8013  026220  104004                          ERROR   4 ;OR 5,6,7     ;REPORT ALL DISCREPANCIES
8014
8015  026222  000402                          BR      3$              ;SKIP TO EXIT
8016  026224                          2$:
8017  026224  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8018  026226  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
8019  026230                          3$:
8020
8021
8022                                    ;;*****************************************************************
8023                                    ;*TEST 101      DRIVE TIMING ERROR
8024                                    ;*      ISSUE A SUBSYSTEM CLEAR. SEEK TO CYLINDER 632.
8025                                    ;*      ISSUE A RECALIBRATE BUT DO NOT WAIT FOR SECOND INTERRUPT.
8026                                    ;*      PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ HEADER
8027                                    ;*      OF CYLINDER 0, TRACK 0. CLOCK THROUGH SEEK
8028                                    ;*      AND DRIVE CLEAR MESSAGES. TURN OFF DIAGNOSTIC MODE.
8029                                    ;*      DRIVE TIMING ERROR SHOULD SET BECAUSE OF NO DATA
8030                                    ;*      TRANSISTIONS ON DATA LINE.
8031                                    ;;*****************************************************************
8032  026230  000004                  TST101: SCOPE
8033  026232  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
8034
8035  026240  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8036  026242  104003                          ERROR   3               ;BAD INIT ERROR
8037
8038  026244  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8039  026250  000117                          SEEK                    ;SEEK
8040  026252  000000                          0                       ;0 WORDS
8041  026254  000000                          0                       ;0 IS BUFF ADDRESS
8042  026256     000                          .BYTE   0               ;SECTOR 0
8043  026257     000                          .BYTE   0               ;TRACK 0
8044  026260  000632                          632                     ;CYLINDER 632
8045
```

```
8046  026262  104417                          TLOADRK                 ;LOAD RK REGS
8047  026264  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8048  026266  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8049  026270  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8050  026274  104430                          TWAT96                  ;WAIT FOR SECOND INTERRUPT
8051  026276  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8052
8053  026300  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8054  026304  000113                          RECAL                   ;RECAL
8055  026306  000000                          0                       ;0 WORDS
8056  026310  000000                          0                       ;0 IS BUFF ADDRESS
8057  026312     000                          .BYTE   0               ;SECTOR 0
8058  026313     000                          .BYTE   0               ;TRACK 0
8059  026314  000000                          0                       ;CYLINDER 0
8060
8061  026316  104417                          TLOADRK                 ;LOAD RK REGS
8062  026320  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8063  026322  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8064
8065  026324  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8066  026330  000125                          RDHEAD                  ;RDHEAD
8067  026332  000000                          0                       ;0 WORDS
8068  026334  000000                          0                       ;0 IS BUFF ADDRESS
8069  026336     000                          .BYTE   0               ;SECTOR 0
8070  026337     000                          .BYTE   0               ;TRACK 0
8071  026340  000000                          0                       ;CYLINDER 0
8072  026342  012737  000040  001616          MOV     #DMD,L.MR1      ;SET DIAG MODE
8073  026350  104417                          TLOADRK                 ;LOAD RK REGS
8074
8075  026352  004437  036004                  JSR     R4,MCLOCK       ;CLOCK CONTROLLER THROUGH SEEK
8076  026356  001062                          1062                    ;AND CLEAR TO READ
8077
8078  026360  005062  000026                  CLR     RKMR1(R2)       ;RESET DIAG MODE, LET RD HDRS COMPLETE
8079
8080  026364  104424                          TWAT32                  ;WAIT FOR INTERRUPT
8081  026366  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8082  026370  104422                          TCHKWE                  ;CHECK OPERATION WITH EXP ERROR
8083  026372  010000                          DTERR                   ;DRIVE TIMING ERROR
8084  026374  000000                          0
8085  026376  000000                          0
8086  026400  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL DISCREPANCIES
8087
8088  026402                          1$:
8089  026402  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8090  026404  104003                          ERROR   3               ;BAD INIT ERROR
8091  026406  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
8092  026414  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8093
8094  026420  104437                          TWAT8S                  ;WAIT FOR INTERRUPT FOR END OF RECAL
8095  026422  104002                          ERROR   2
8096
8097  026424  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8098  026430  000105                          CLEAR                   ;CLEAR
8099  026432  000000                          0                       ;0 WORDS
8100  026434  000000                          0                       ;0 IS BUFF ADDRESS
8101  026436     000                          .BYTE   0               ;SECTOR 0
```

```
026437  000                             .BYTE   0               ;TRACK 0
026440  000000                          0                       ;CYLINDER 0

026442  104417                          TLOADRK                 ;LOAD RK REGS
026444  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
026446  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
026450  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
026452  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

                                .SBTTL  **ERROR FORCING IN DRIVE

                                ;;*****************************************************************
                                ;*TEST 102       INITIALIZE CLEARING SACK
                                ;*
                                ;*      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
                                ;*      DRIVE.  ISSUE A SUBSYSTEM CLEAR.  PUT CONTROLLER IN
                                ;*      DIAGNOSTIC MODE.  ISSUE A SELECT COMMAND WITH
                                ;*      MESSAGE ID = 3 AND DRIVE SELECTED = 0.  CLOCK THROUGH
                                ;*      PHASE ADDRESS 6.  TURN OFF DIAGNOSTIC MODE.  MAKE
                                ;*      SURE UNIT FIELD ERROR DOES NOT SET.
                                ;*
                                ;;*****************************************************************
026454  000004                  TST102: SCOPE
026456  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
026464  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
026466  104003                          ERROR   3               ;BAD INIT ERROR

026470  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
026474  000101                          SELDRV                  ;SELDRV
026476  000000                          0                       ;0 WORDS
026500  000000                          0                       ;0 IS BUFF ADDRESS
026502  000                             .BYTE   0               ;SECTOR 0
026503  000                             .BYTE   0               ;TRACK 0
026504  000000                          0                       ;CYLINDER 0

026506  104417                          TLOADRK                 ;LOAD RK REGS
026510  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
026512  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

026514  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
026516  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

026520  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
026522  104003                          ERROR   3               ;BAD INIT ERROR

026524  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
026530  000101                          SELDRV                  ;SELDRV
026532  000000                          0                       ;0 WORDS
026534  000000                          0                       ;0 IS BUFF ADDRESS
026536  000                             .BYTE   0               ;SECTOR 0
026537  000                             .BYTE   0               ;TRACK 0
026540  000000                          0                       ;CYLINDER 0
026542  012737  000043  001616          MOV     #3!DMD,L.MR1    ;SET DIAG MODE AND MESSAGE PAIR 3
026550  005037  001610                  CLR     L.CS2           ;SELECT DRIVE 0

026554  104417                          TLOADRK                 ;LOAD RK REGS
```

```
8158
8159  026556  004437  036004          JSR     R4,MCLOCK       ;CLOCK THROUGH PHASE ADDRESS 6
8160  026562  001027                  1027
8161
8162  026564  042762  000040  000026  BIC     #DMD,RKMR1(R2)  ;CLEAR MAINTENANCE MODE
8163
8164  026572  104424                  TWAT32                  ;WAIT FOR INTERRUPT
8165  026574  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE
8166
8167  026576  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8168  026600  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8169                          ;;*****************************************************************
8170                          ;*TEST 103      DRIVE OFF TRACK
8171                          ;*
8172                          ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.  ISSUE
8173                          ;*      OFFSET OF +1200 MICRO-INCHES.  PUT CONTROLLER IN DIAGNOSTIC
8174                          ;*      MODE.  ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
8175                          ;*      TRACK 0, SECTOR 0.  CLOCK THROUGH SEEK AND DRIVE CLEAR
8176                          ;*      MESSAGES.  TURN OFF DIAGNOSTIC MODE.  DRIVE OFF TRACK
8177                          ;*      SHOULD SET IN DRIVE.  REPEAT FOR ALL AVAILIABLE DRIVES.
8178                          ;*
8179                          ;;*****************************************************************
8180  026602  000004          TST103: SCOPE
8181  026604  012737  000005  001262  MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
8182  026612  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
8183  026614  104003                  ERROR   3               ;BAD INIT ERROR
8184
8185  026616  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
8186  026622  000113                  RECAL                   ;RECAL
8187  026624  000000                  0                       ;0 WORDS
8188  026626  000000                  0                       ;0 IS BUFF ADDRESS
8189  026630     000                  .BYTE   0               ;SECTOR 0
8190  026631     000                  .BYTE   0               ;TRACK 0
8191  026632  000000                  0                       ;CYLINDER 0
8192
8193  026634  104417                  TLOADRK                 ;LOAD RK REGS
8194  026636  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8195  026640  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8196  026642  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
8197
8198  026646  104437                  TWATBS                  ;WAIT FOR INTERRUPT #2
8199  026650  104002                  ERROR   2
8200
8201  026652  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
8202  026656  000105                  CLEAR                   ;CLEAR
8203  026660  000000                  0                       ;0 WORDS
8204  026662  000000                  0                       ;0 IS BUFF ADDRESS
8205  026664     000                  .BYTE   0               ;SECTOR 0
8206  026665     000                  .BYTE   0               ;TRACK 0
8207  026666  000000                  0                       ;CYLINDER 0
8208
8209  026670  104417                  TLOADRK                 ;LOAD RK REGS
8210  026672  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8211  026674  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8212
8213  026676  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
```

```
8214  026700  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8215
8216  026702  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8217  026706  000115                          OFFSET                  ;OFFSET
8218  026710  000000                          0                       ;0 WORDS
8219  026712  000000                          0                       ;0 IS BUFF ADDRESS
8220  026714     000                          .BYTE   0               ;SECTOR 0
8221  026715     000                          .BYTE   0               ;TRACK 0
8222  026716  000000                          0                       ;CYLINDER 0
8223  026720  112737  000060  001612          MOVB    #60,L.ASOF      ;SET OFFSET AT +1200
8224
8225  026726  104417                          TLOADRK                 ;LOAD RK REGS
8226  026730  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8227  026732  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8228
8229  026734  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8230  026736  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8231
8232  026740  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8233
8234  026744  104424                          TWAT32                  ;WAIT FOR INT #2
8235  026746  104002                          ERROR   2
8236
8237  026750  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8238  026752  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8239
8240  026754  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8241  026760  000105                          CLEAR                   ;CLEAR
8242  026762  000000                          0                       ;0 WORDS
8243  026764  000000                          0                       ;0 IS BUFF ADDRESS
8244  026766     000                          .BYTE   0               ;SECTOR 0
8245  026767     000                          .BYTE   0               ;TRACK 0
8246  026770  000000                          0                       ;CYLINDER 0
8247
8248  026772  104417                          TLOADRK                 ;LOAD RK REGS
8249  026774  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8250  026776  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8251
8252  027000  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8253  027002  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8254
8255  027004  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8256  027010  000023                          23                      ;WRITE DATA W/O INTERRUPT ENABLE
8257  027012  177777                          -1                      ;-1 WORDS
8258  027014  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
8259  027016     000                          .BYTE   0               ;SECTOR 0
8260  027017     000                          .BYTE   0               ;TRACK 0
8261  027020  000000                          0                       ;CYLINDER 0
8262  027022  012737  000040  001616          MOV     #DMD,L.MR1      ;SET DIAGNOSTIC MODE
8263
8264  027030  104417                          TLOADRK
8265
8266  027032  004437  036004                  JSR     R4,MCLOCK       ;CLOCK THROUGH SEEK & DRIVE CLEAR
8267  027036  001064                          1064
8268
8269  027040  005062  000026                  CLR     RKMR1(R2)       ;CLEAR DIAGNOSTIC MODE
```

```
8270  027044  104426                          TWAT64                          ;WAIT FOR INTERRUPT
8271  027046  012762  100000  000000          MOV     #CCLR,RKCS1(R2)         ;CLEAR CONTROLLER
8272  027054  104423                          TWAT16                          ;STALL FOR 16 MS
8273  027056  000240                          NOP
8274  027060  000240                          NOP
8275
8276  027062  004437  035530                  JSR     R4,LRLOAD               ;LOAD "L" REGS
8277  027066  000101                          SELDRV                          ;SELDRV
8278  027070  000000                          0                               ;0 WORDS
8279  027072  000000                          0                               ;0 IS BUFF ADDRESS
8280  027074     000                          .BYTE   0                       ;SECTOR 0
8281  027075     000                          .BYTE   0                       ;TRACK 0
8282  027076  000000                          0                               ;CYLINDER 0
8283  027100  005037  001616                  CLR     L.MR1                   ;RESET DIAG MODE
8284
8285  027104  104417                          TLOADRK                         ;LOAD RK REGS
8286  027106  104423                          TWAT16                          ;WAIT FOR INTERRUPT
8287  027110  104002                          ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR
8288
8289  027112  104422                          TCHKWE                          ;CHECK OPERATION WITH ERROR EXPECTED
8290  027114  000400                          DROTERR                         ;DRIVE OFF TRACK
8291  027116  000000                          0
8292  027120  000000                          0
8293  027122  104004                          ERROR   4; OR 5,6,7             ;REPORT ANY DISCREPANCIES
8294                                ;;*****************************************************************
8295                                ;*TEST 104      FILE UNSAFE
8296                                ;*
8297                                ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECLAIBRATE.  ISSUE
8298                                ;*      A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR
8299                                ;*      FORMAT.  DO A SELECT COMMAND IN 26 SECTOR FORMAT.
8300                                ;*      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
8301                                ;*      HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR
8302                                ;*      FORMAT.  CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.
8303                                ;*      SIMULATE INDEX PULSE.  TURN OFF DIAGNOSTIC MODE.  FILE
8304                                ;*      UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE
8305                                ;*      THROUGH SECTOR PULSE.  REPEAT FOR ALL AVAILIABLE DRIVES.
8306                                ;*
8307                                ;*****************************************************************
8308  027124  000004                TST104: SCOPE
8309  027126  012737  000005  001262          MOV     #5.,$TIMES              ;;DO 5. ITERATIONS
8310  027134  012737  177777  001676          MOV     #-1,REFMT               ;SET REFORMAT SWITCH
8311  027142  104416                          TSSINIT                         ;CLEAR SUBSYSTEM
8312  027144  104003                          ERROR   3                       ;BAD INIT ERROR
8313
8314  027146  004437  035530                  JSR     R4,LRLOAD               ;LOAD "L" REGS
8315  027152  000113                          RECAL                           ;RECAL
8316  027154  000000                          0                               ;0 WORDS
8317  027156  000000                          0                               ;0 IS BUFF ADDRESS
8318  027160     000                          .BYTE   0                       ;SECTOR 0
8319  027161     000                          .BYTE   0                       ;TRACK 0
8320  027162  000000                          0                               ;CYLINDER 0
8321
8322  027164  104417                          TLOADRK                         ;LOAD RK REGS
8323  027166  104423                          TWAT16                          ;WAIT FOR INTERRUPT
8324  027170  104002                          ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR
8325
```

```
8326  027172  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8327  027174  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8328
8329  027176  005037  001662          CLR     INTSET          ;CLEAR INT FLAG
8330  027202  104437                  TWATBS                  ;WAIT FOR SECOND INT
8331  027204  104002                  ERROR   2
8332
8333  027206  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8334  027210  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8335
8336  027212  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
8337  027216  000105                  CLEAR                   ;CLEAR
8338  027220  000000                  0                       ;0 WORDS
8339  027222  000000                  0                       ;0 IS BUFF ADDRESS
8340  027224     000                  .BYTE   0               ;SECTOR 0
8341  027225     000                  .BYTE   0               ;TRACK 0
8342  027226  000000                  0                       ;CYLINDER 0
8343
8344  027230  104417                  TLOADRK                 ;LOAD RK REGS
8345  027232  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8346  027234  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8347
8348  027236  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8349  027240  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8350
8351  027242  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
8352  027246  010125                  RDHEAD!CFM†                     ;RDHEAD!CFMT
8353  027250  000000                  0                       ;0 WORDS
8354  027252  000000                  0                       ;0 IS BUFF ADDRESS
8355  027254     000                  .BYTE   0               ;SECTOR 0
8356  027255     000                  .BYTE   0               ;TRACK 0
8357  027256  000000                  0                       ;CYLINDER 0
8358
8359  027260  104417                  TLOADRK                 ;LOAD RK REGS
8360  027262  104424                  TWAT32                  ;WAIT FOR INTERRUPT
8361  027264  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8362
8363  027266  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8364  027270  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8365
8366  027272  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
8367  027276  000101                  SELDRV                  ;SELDRV
8368  027300  000000                  0                       ;0 WORDS
8369  027302  000000                  0                       ;0 IS BUFF ADDRESS
8370  027304     000                  .BYTE   0               ;SECTOR 0
8371  027305     000                  .BYTE   0               ;TRACK 0
8372  027306  000000                  0                       ;CYLINDER 0
8373
8374  027310  104417                  TLOADRK                 ;LOAD RK REGS
8375  027312  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8376  027314  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8377
8378  027316  004437  035530          JSR     R4,LRLOAD       ;LOAD "L" REGS
8379  027322  000127                  WRHEAD                  ;WRHEAD
8380  027324  177777                  -1                      ;-1 WORDS
8381  027326  063414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
```

```
8382  027330  000                       .BYTE   0               ;SECTOR 0
8383  027331  000                       .BYTE   0               ;TRACK 0
8384  027332  000000                    0                       ;CYLINDER 0
8385  027334  012737  000040  001616    MOV     #DMD,L.MR1      ;SET DIAGNOSTIC-MODE
8386
8387  027342  104417                    TLOADRK                 ;LOAD RK REGS
8388  027344  004437  036004            JSR     R4,MCLOCK       ;CLOCK THROUGH SEEK AND DRIVE CLEAR
8389  027350  001064                    1064
8390
8391  027352  052762  000200  000026    BIS     #MIND,RKMR1(R2) ;SET INDEX
8392
8393  027360  004437  036004            JSR     R4,MCLOCK       ;CLOCK INDEX
8394  027364  001001                    1001
8395
8396  027366  042762  000200  000026    BIC     #MIND,RKMR1(R2) ;CLEAR INDEX
8397
8398  027374  004437  036004            JSR     R4,MCLOCK       ;CLOCK CLEAR
8399  027400  001001                    1001
8400
8401  027402  005062  000026            CLR     RKMR1(R2)       ;CLEAR DIAGNOSTIC MODE
8402
8403  027406  104426                    TWAT64                  ;WAIT FOR INTERRUPT
8404  027410  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8405
8406  027412  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8407  027414  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
8408
8409  027416  004437  035530            JSR     R4,LRLOAD       ;LOAD "L" REGS
8410  027422  000101                    SELDRV                  ;SELDRV
8411  027424  000000                    0                       ;0 WORDS
8412  027426  000000                    0                       ;0 IS BUFF ADDRESS
8413  027430  000                       .BYTE   0               ;SECTOR 0
8414  027431  000                       .BYTE   0               ;TRACK 0
8415  027432  000000                    0                       ;CYLINDER 0
8416
8417  027434  005037  001616            CLR     L.MR1           ;CLEAR DIAG MODE
8418
8419  027440  104417                    TLOADRK                 ;LOAD RK REGS
8420  027442  104423                    TWAT16                  ;WAIT FOR INTERRUPT
8421  027444  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8422
8423  027446  104422                    TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
8424  027450  040400                    UNSERR!DROTERR          ;UNSAFE AND DRIVE OFF TRACK
8425  027452  000000                    0
8426  027454  000000                    0
8427  027456  104004                    ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
8428
8429  027460  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
8430  027462  104003                    ERROR   3               ;BAD INIT ERROR
8431
8432  027464  004437  035530            JSR     R4,LRLOAD       ;LOAD "L" REGS
8433  027470  000101                    SELDRV                  ;SELDRV
8434  027472  000000                    0                       ;0 WORDS
8435  027474  000000                    0                       ;0 IS BUFF ADDRESS
8436  027476  000                       .BYTE   0               ;SECTOR 0
8437  027477  000                       .BYTE   0               ;TRACK 0
```

```
8438  027500  000000                          0                       ;CYLINDER 0
8439
8440  027502  012737  000001  001616          MOV     #1,L.MR1        ;SET MESSAGE SELECT ONE
8441
8442  027510                                  1$:
8443  027510  104417                          TLOADRK                 ;LOAD RK REGS
8444  027512  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
8445  027514  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8446
8447  027516  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8448  027520  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8449
8450  027522  032737  000040  001574          BIT     #S.HDHM,T.MR2   ;TEST IF HEADS HOME
8451  027530  001767                          BEQ     1$
8452
8453  027532  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8454  027534  104003                          ERROR   3               ;BAD INIT ERROR
8455
8456  027536  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8457  027542  104434                          TWAIT159                ;WAIT FOR APPROX 160 MS
8458  027544  000240                          NOP                     ;DON'T CARE ERROR RETURN
8459
8460  027546  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8461  027550  104003                          ERROR   3               ;BAD INIT ERROR
8462
8463  027552  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
8464
8465  027560  104437                          TWATBS                  ;WAIT FOR SECOND INTERRUPT
8466  027562  104002                          ERROR   2
8467
8468  027564  005037  001616                  CLR     L.MR1           ;CLEAR MR1
8469
8470  027570  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8471  027574  000105                          CLEAR                   ;CLEAR
8472  027576  000000                          0                       ;0 WORDS
8473  027600  000000                          0                       ;0 IS BUFF ADDRESS
8474  027602     000                          .BYTE   0               ;SECTOR 0
8475  027603     000                          .BYTE   0               ;TRACK 0
8476  027604  000000                          0                       ;CYLINDER 0
8477
8478  027606  104417                          TLOADRK                 ;LOAD RK REGS
8479  027610  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
8480  027612  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8481
8482  027614  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8483  027616  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8484
8485  027620  004437  041740                  JSR     R4,GENCOM       ;BUILD HEADERS
8486  027624  001200                          1200
8487
8488  027626  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8489  027632  000127                          WRHEAD                  ;WRHEAD
8490  027634  177676                          -102                    ;-102 WORDS
8491  027636  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
8492  027640     000                          .BYTE   0               ;SECTOR 0
8493  027641     000                          .BYTE   0               ;TRACK 0
```

```
 8494  027642  000000                          0                       ;CYLINDER 0
 8495
 8496  027644  104417                          TLOADRK                 ;LOAD RK REGS
 8497  027646  104426                          TWAT64                  ;WAIT FOR INTERRUPT
 8498  027650  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8499
 8500  027652  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8501  027654  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8502
 8503                                  ;;*****************************************************************
 8504                                  ;*TEST 105      DUMMY TEST FOR PREVIOUS TEST EXIT
 8505                                  ;*      THIS TEST IS PRESENT TO MAKE $SWDBTB TABLE HAVE AN ENTRY
 8506                                  ;*      WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS
 8507                                  ;*      IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST.
 8508                                  ;*      IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE
 8509                                  ;*      "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".
 8510                                  ;*
 8511                                  ;*      IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED
 8512                                  ;*      TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS
 8513                                  ;*      FROM THE DRIVE COMING READY AT A LATER TIME.
 8514                                  ;*****************************************************************
 8515  027656  000004                  TST105: SCOPE
 8516  027660  012737  000001  001262          MOV     #1,$TIMES       ;;DO 1 ITERATION
 8517
 8518  027666  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 8519  027670  104003                          ERROR   3               ;BAD INIT ERROR
 8520
 8521  027672  013762  001626  000010          MOV     DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
 8522  027700  012762  000001  000000          MOV     #1,RKCS1(R2)    ;SELECT THE DRIVE
 8523  027706  032762  000200  000012  1$:     BIT     #DRDY,RKDS(R2)  ;TEST IF DRIVE READY
 8524  027714  001774                          BEQ     1$              ;NO LOOP
 8525
 8526  027716  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 8527  027720  104003                          ERROR   3               ;BAD INIT ERROR
 8528
 8529  027722  004437  041740                  JSR     R4,GENCOM       ;GENERATE HEADERS FOR CYL 0
 8530  027726  001200                          1200
 8531
 8532  027730  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 8533  027734  000127                          WRHEAD                  ;WRHEAD
 8534  027736  177676                          -102                    ;-102 WORDS
 8535  027740  063414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 8536  027742     000                          .BYTE   0               ;SECTOR 0
 8537  027743     000                          .BYTE   0               ;TRACK 0
 8538  027744  000000                          0                       ;CYLINDER 0
 8539
 8540  027746  104417                          TLOADRK                 ;LOAD RK REGS
 8541  027750  104426                          TWAT64                  ;WAIT FOR INTERRUPT
 8542  027752  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8543
 8544  027754  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8545  027756  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8546  027760  005037  001676                  CLR     REFMT           ;CLEAR REFORMAT SWITCH
```

```
                                        .SBTTL  **MULTI-DRIVE OPERATIONS
                                NEWDRV: SCOPE
                                        MOV     #1,$TIMES       ;DO ONLY ONCE
                                        BIT     #BIT7,DRVBIT    ;WERE WE TESTING DRIVE 7?
                                        BNE     3$              ;YES-SKIP
                                
                                1$:     INC     DRVNUM          ;BUMP TO NEXT SEQUENTIAL ADDRESS
                                        ASL     DRVBIT          ;BUMP DRIVEBIT TO THAT POSITION
                                        BIT     DRVBIT,$DEVM    ;IS THIS DRIVE TO BE TESTED?
                                        BNE     2$              ;YES-EXIT
                                        BIT     #BIT8,DRVBIT    ;ALL DRIVES TESTED?
                                        BNE     3$              ;YES-EXIT
                                        BR      1$              ;ELSE CHECK NEXT DRIVE AVAILABLE
                                
                                2$:     MOVB    #4,$TSTNM       ;SET TEST NUMBER FOR REPORTS
                                        JMP     TSTLUP          ;GO TO TEST LOOP TO CHECK THIS DRIVE
                                3$:     CLR     DRVBIT          ;CLEAR DRIVE BIT
                                        CLR     DRVNUM          ;CLEAR DRIVE NUMBER
                                
                                ;;*****************************************************************
                                ;*TEST 106      RESET ATTENTIONS WITH UNIBUS INIT
                                ;*
                                ;*      DO A RECALIBRATE ON ALL AVAILIABLE DRIVES.
                                ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
                                ;*
                                ;;*****************************************************************
                                TST106: SCOPE
                                        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
                                        CLR     R0              ;CLEAR DRIVE POSITION COUNTER
                                        MOV     #1,R1           ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
                                        MOV     $DEVM,R3        ;GET DEVICE MAP
                                        TSSINIT                 ;CLEAR SUBSYSTEM
                                        ERROR   3               ;BAD INIT ERROR
                                1$:     BIT     R1,R3           ;TEST IF THIS DRIVE AVAILABLE
                                        BNE     2$              ;YES-SKIP TO SEEK
                                3$:     ASL     R1              ;SHIFT DRIVE SELECT BIT
                                        INC     R0              ;BUMP DRIVE POSITION COUNTER
                                        BIT     #BIT8,R1        ;ALL DRIVE POSITIONS CHECKED
                                        BEQ     1$              ;NO-LOOP
                                        BR      4$              ;SKIP TO RESET
                                
                                2$:     MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
                                        MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
                                
                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT16                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                
                                        CLR     INTSET          ;CLEAR INTERRUPT FLAG
                                        MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
                                        MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
                                12$:    MOV     RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
                                        MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
                                        BIC     #177400,R4      ;CLEAR UNUSED BITS
                                        BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
                                        BNE     10$             ;YES - SKIP
```

```
8603
8604  030206  104423                        TWAT16                  ;WAIT FOR 16 MS
8605  030210  000240                        NOP                     ;DON'T CARE RETURNS
8606  030212  000240                        NOP
8607  030214  005305                        DEC     R5              ;TATOL WAIT TIME IS 8 SECONDS
8608  030216  001362                        BNE     12$             ;CHECK ATTENTION EACH 16 MS
8609  030220  104002                        ERROR   2               ;REPORT IF NO ATTENTION IN 8 SEC
8610
8611  030222                          10$:
8612  030222  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8613  030224  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8614
8615  030226  000731                        BR      3$              ;LOOP FOR NEXT DRIVE
8616  030230  000005                4$:     RESET                   ;UNIBUS RESET
8617  030232  004737  044626                JSR     PC,$TKINT       ;RESET KEYBOARD INTERRUPT ENABLE
8618
8619  030236  012701  000031                MOV     #25.,R1         ;DO A SHORT DELAY
8620  030242  005301                5$:     DEC     R1
8621  030244  001376                        BNE     5$
8622  030246  004737  034434                JSR     PC,OPTTST       ;SET UP OPTIONS
8623
8624  030252  104420                        TGETRK                  ;GET RK611 REGS
8625
8626  030254  105737  001557                TSTB    T.ASOF+1        ;ALL ATTENTION RESET?
8627  030260  001407                        BEQ     6$              ;YES-SKIP
8628
8629  030262  012737  056526  001470        MOV     #EMDA,EM12N     ;"DRIVE ATT NOT RESET RESULT OF
8630  030270  012737  056652  061010        MOV     #EMUR,DF011A    ;UNIBUS RESET"
8631  030276  104012                        ERROR   12
8632
8633  030300                          6$:
8634                                  ;;*****************************************************************
8635                                  ;*TEST 107      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
8636                                  ;*
8637                                  ;*      DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
8638                                  ;*      ISSUE A SUBSYSTEM CLEAR.  MAKE SURE ALL ATTENTIONS
8639                                  ;*      RESET.
8640                                  ;*
8641                                  ;;*****************************************************************
8642  030300  000004                TST107: SCOPE
8643  030302  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
8644  030310  005000                        CLR     R0              ;CLEAR DRIVE POSITION COUNTER
8645  030312  012701  000001                MOV     #1,R1           ;PRESET TO TEST POSITION 0
8646  030316  013703  001354                MOV     $DEVM,R3        ;GET DEVICE MAP
8647  030322  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
8648  030324  104003                        ERROR   3               ;BAD INIT ERROR
8649  030326  030103                1$:     BIT     R1,R3           ;THIS DRIVE AVAILABLE?
8650  030330  001006                        BNE     2$              ;YES-SKIP TO SEEK
8651  030332  006301                3$:     ASL     R1              ;SHIFT TO NEXT DRIVE POSITION
8652  030334  005200                        INC     R0              ;DUMP POSITION COUNTER
8653  030336  032701  000400                BIT     #BIT8,R1        ;ALL POSITIONS CHECKED
8654  030342  001771                        BEQ     1$              ;NO-LOOP
8655  030344  000441                        BR      4$              ;YES-SKIP TO CLEAR
8656
8657  030346  010037  001610        2$:     MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
8658  030352  012737  000113  001600        MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
```

```
 8659  030360  104417                          TLOADRK                 ;LOAD RK REGS
 8660  030362  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 8661  030364  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8662
 8663  030366  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
 8664  030372  012705  000764                  MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
 8665  030376  012762  000012  000000          MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
 8666  030404  016237  000016  001556  12$:    MOV     RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
 8667  030412  113704  001557                  MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
 8668  030416  042704  177400                  BIC     #177400,R4      ;CLEAR UNUSED BITS
 8669  030422  030104                          BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
 8670  030424  001006                          BNE     10$             ;YES - SKIP
 8671
 8672  030426  104423                          TWAT16                  ;WAIT FOR 16 MS
 8673  030430  000240                          NOP                     ;DON'T CARE RETURNS
 8674  030432  000240                          NOP
 8675  030434  005305                          DEC     R5              ;TATOL WAIT TIME IS 8 SECONDS
 8676  030436  001362                          BNE     12$             ;CHECK ATTENTION EACH 16 MS
 8677  030440  104002                          ERROR   2               ;REPORT IF NO ATTENTION IN 8 SEC
 8678
 8679  030442                          10$:
 8680  030442  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8681  030444  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8682
 8683  030446  000731                          BR      3$              ;LOOP FOR NEXT DRIVE
 8684
 8685  030450  052762  000040  000010  4$:     BIS     #SCLR,RKCS2(R2) ;DO SUBSYSTEM CLEAR
 8686  030456  012701  000031                  MOV     #25.,R1         ;DO A SHORT DELAY
 8687  030462  005301                  5$:     DEC     R1
 8688  030464  001376                          BNE     5$
 8689
 8690  030466  104420                          TGETRK                  ;GET RK611 REGS
 8691
 8692  030470  105737  001557                  TSTB    T.ASOF+1        ;TEST ALL ATTENTION RESET
 8693  030474  001407                          BEQ     6$              ;YES-SKIP
 8694
 8695  030476  012737  056526  001470          MOV     #EMDA,EM12N     ;"DRIVE ATT NOT RESET AS RESULT OF
 8696  030504  012737  056742  061010          MOV     #EMSCLR,DF011A  ;SUBSYSTEM CLEAR"
 8697  030512  104012                          ERROR   12
 8698
 8699  030514                          6$:
 8700                                  ;;*****************************************************************
 8701                                  ;*TEST 110      SVAL AND ATTENTION FROM OTHER DRIVE
 8702                                  ;*
 8703                                  ;*      DO A RECALIBRATE ON ONE AVAILABLE DRIVE.  DO A SELECT
 8704                                  ;*      ON ANOTHER AVAILABLE DRIVE.  MAKE SURE STATUS VALID
 8705                                  ;*      IS SET.  WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
 8706                                  ;*      MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
 8707                                  ;*      CHANGE REMAINS RESET.
 8708                                  ;*
 8709                                  ;*      REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.
 8710                                  ;*
 8711                                  ;*      NOTE:   THIS TEST WILL ONLY BE DONE IF AT LEAST
 8712                                  ;*              TWO DRIVES ARE AVAILABLE.
 8713                                  ;*
 8714                                  ;;*****************************************************************
```

```
 8715  030514  000004                        TST11O: SCOPE
 8716  030516  012737  000012  001262                MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 8717  030524  013746  001354                        MOV     $DEVM,-(SP)     ;PUT DEVICE MAP ON STACK
 8718  030530  004437  035730                        JSR     R4,BITCNT       ;COUNT NUMBER OF BITS(# OF DRIVES)
 8719  030534  022627  000001                        CMP     (SP)+,#1        ;COMPARE TO 1
 8720  030540  101007                                BHI     2$              ;SKIP IF MORE THAN 1
 8721  030542  005737  001304                        TST     $PASS           ;CHECK IF PASS 0
 8722  030546  001002                                BNE     1$              ;NO-SKIP
 8723
 8724  030550  104401  052562                        TYPE    ,OPRO18         ;"OVERLAPPED OPERATION BYPASSED"
 8725  030554  000137  031770              1$:       JMP     $EOP            ;GET OUT.
 8726
 8727  030560  012737  177777  001224    2$:       MOV     #-1,$TMP1       ;SET LOOP CONTROL FLAG
 8728  030566  013705  001354                        MOV     $DEVM,R5        ;GET DEVICE MAP
 8729  030572  005000                                CLR     R0              ;CLEAR FOR DRIVE #A
 8730  030574  005001                                CLR     R1              ;CLEAR FOR DRIVE #B
 8731  030576  012703  000001                        MOV     #1,R3           ;SET DRIVE POSITION A
 8732  030602  012704  000001                        MOV     #1,R4           ;SET DRIVE POSITION B
 8733  030606  012737  030616  001110                MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
 8734  030614  000477                                BR      11$             ;GO SET UP POINTERS
 8735
 8736  030616                              3$:
 8737  030616  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
 8738  030620  104003                                ERROR   3               ;BAD INIT ERROR
 8739
 8740  030622  010037  001610                        MOV     R0,L.CS2        ;LOAD DRIVE A
 8741  030626  010037  001626                        MOV     R0,DRVNUM       ;LOAD FOR REPORT
 8742  030632  012737  000113  001600                MOV     #RECAL,L.CS1    ;LOAD RECAL
 8743
 8744  030640  104417                                TLOADRK                 ;LOAD RK REGS
 8745  030642  104423                                TWAT16                  ;WAIT FOR INTERRUPT
 8746  030644  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8747  030646  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8748  030650  104004                                ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8749
 8750  030652  005037  001662                        CLR     INTSET          ;CLEAR INT FLAG
 8751
 8752  030656  010137  001610                        MOV     R1,L.CS2        ;LOAD DRIVE B
 8753  030662  010137  001626                        MOV     R1,DRVNUM       ;LOAD FOR REPORT
 8754  030666  012737  000101  001600                MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
 8755
 8756  030674  104417                                TLOADRK                 ;LOAD RK REGS
 8757  030676  104423                                TWAT16                  ;WAIT FOR INTERRUPT
 8758  030700  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8759
 8760  030702  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8761  030704  104004                                ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8762
 8763  030706  032737  100000  001552                BIT     #SVAL,T.DS      ;CHECK IF STATUS VALID SET
 8764  030714  001007                                BNE     4$              ;YES - SKIP
 8765  030716  012737  056342  001460                MOV     #EMSVAL,EM11N   ;"STATUS VALID NOT SET RESULT OF
 8766  030724  012737  047640  061010                MOV     #OPER00,DF011A  ;DRIVE SELECT"
 8767  030732  104011                                ERROR   11
 8768
 8769  030734  005037  001662              4$:       CLR     INTSET          ;CLEAR INT FLAG
 8770  030740  104436                                TWAT2S                  ;WAIT FOR SEEK COMPLETE INTERRUPT
```

```
8771  030742  000401                        BR      44$             ;NONE RECEIVED - SKIP
8772  030744  000406                        BR      55$             ; RECEIVED - SKIP
8773
8774  030746  010037  001626        44$:    MOV     R0,DRVNUM       ;SET DRIVE FOR REPORT
8775  030752  012737  060377  001372        MOV     #DH017,DH2N     ;"COMMAND - SELECT AFTER RECAL"
8776  030760  104002                        ERROR   2
8777
8778  030762  104420                55$:    TGETRK                  ;GET RK REGS
8779  030764  032737  100000  001552        BIT     #SVAL,T.DS      ;TEST IF SVAL STILL SET
8780  030772  001007                        BNE     5$              ;YES - SKIP
8781
8782  030774  012737  056342  001510        MOV     #EMSVAL,EM14N   ;"STATUS VALID RESET RESULT OF
8783  031002  012737  060422  061010        MOV     #DH018,DF011A   ;RECAL COMPLETE ATTENTION AFTER SEL"
8784  031010  104014                        ERROR   14
8785
8786  031012  104415                5$:     SCOP1                   ;LOCAL LOOP TO 3$
8787
8788                                        ;       THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
8789                                        ;       OF DRIVES AVAILABLE.  THE FIRST PASS OF THE PROGRAM WILL USE THE
8790                                        ;       LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
8791                                        ;       B.  THE SECOND PASS SWAPS DRIVE A & B.  THE THIRD PASS USES
8792                                        ;       THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
8793                                        ;       AS A.  THE FORTH PASS SWAPS A & B AGAIN.  THIS CONTINUES
8794                                        ;       UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
8795                                        ;       DRIVE.
8796                                        ;
8797                                        ;       THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
8798                                        ;       THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
8799                                        ;
8800                                        ;       THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
8801                                        ;       CHECKED.
8802
8803  031014  005237  001224        11$:    INC     $TMP1           ;INCREMENT PASS CONTROL
8804  031020  001024                        BNE     16$             ;SKIP IF NOT ZERO
8805                                                                ;(IT WILL BE ZERO ON THE 1ST PASS)
8806
8807  031022  030305                12$:    BIT     R3,R5           ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8808  031024  001006                        BNE     13$             ;YES-SKIP
8809
8810  031026  005200                22$:    INC     R0              ;BUMP R0 (DRIVE A)
8811  031030  006303                        ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
8812  031032  032703  000400                BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
8813  031036  001771                        BEQ     12$             ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8814  031040  000464                        BR      50$             ;DONE-EXIT
8815
8816  031042  010001                13$:    MOV     R0,R1           ;SET DRIVE B TO THE SAME AS A
8817  031044  010304                        MOV     R3,R4
8818  031046  005201                14$:    INC     R1              ;BUMP R1 (DRIVE B)
8819  031050  006304                        ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
8820  031052  030405                        BIT     R4,R5           ;IS THIS DRIVE AVAIL?
8821  031054  001004                        BNE     15$             ;YES-SKIP
8822  031056  032704  000400                BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
8823  031062  001771                        BEQ     14$             ;NO-LOOP
8824  031064  000452                        BR      50$             ;DONE-EXIT
8825
8826  031066  000137  030616        15$:    JMP     3$              ;GO DO THE TEST ON THE DRIVE A & B
```

```
 8827                                                           ;CONTAINED IN R0 & R1
 8828  031072  032737  000001  001224  16$:    BIT     #BIT0,$TMP1     ;IS PASS FLAGS ODD?
 8829  031100  001410                          BEQ     17$             ;NO-SKIP
 8830
 8831  031102  010046                          MOV     R0,-(SP)        ;
 8832  031104  010346                          MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
 8833  031106  010403                          MOV     R4,R3           ;TO EXCHANGE DRIVE A & B
 8834  031110  010100                          MOV     R1,R0
 8835  031112  012604                          MOV     (SP)+,R4
 8836  031114  012601                          MOV     (SP)+,R1
 8837  031116  000137  030616                  JMP     3$              ;REPEAT TEST ON THIS COMBO.
 8838
 8839  031122  032737  000002  001224  17$:    BIT     #BIT1,$TMP1     ;TEST IF PASS FLAGS AT HALF MODULE 4?
 8840  031130  001410                          BEQ     19$             ;NO-SKIP TO BUMP DRIVE B
 8841  031132  005200                  18$:    INC     R0              ;BUMP DRIVE A
 8842  031134  006303                          ASL     R3              ;SHIFT DRIVE SELECT BIT
 8843  031136  030305                          BIT     R3,R5           ;AVAILABLE?
 8844  031140  001014                          BNE     20$             ;YES-SKIP
 8845  031142  032703  000400                  BIT     #BIT8,R3        ;ALL CHECKED?
 8846  031146  001771                          BEQ     18$             ;NO-SKIP
 8847  031150  000412                          BR      21$             ;GO TO NEXT PASS
 8848
 8849  031152  005201                  19$:    INC     R1              ;BUMP DRIVE B
 8850  031154  006304                          ASL     R4              ;SHIFT DRIVE SELECT BIT
 8851  031156  030405                          BIT     R4,R5           ;AVAILABLE?
 8852  031160  001004                          BNE     20$             ;YES-SKIP
 8853  031162  032704  000400                  BIT     #BIT8,R4        ;ALL CHECKED?
 8854  031166  001771                          BEQ     19$             ;NO-LOOP
 8855  031170  000404                          BR      23$             ;YES-SKIP TO NEXT PASS
 8856
 8857  031172  000137  030616          20$:    JMP     3$              ;GO TEST THIS COMBO
 8858
 8859  031176  010100                  21$:    MOV     R1,R0           ;SET DRIVE 0 TO LOW POSITION THIS PASS
 8860  031200  010403                          MOV     R4,R3           ;SET SELECT BITS TO AGREE
 8861  031202  005037  001224          23$:    CLR     $TMP1           ;CLEAR PASS FLAGS
 8862  031206  000137  031026                  JMP     22$             ;GO SET UP A & B
 8863  031212                          50$:
 8864                                  ;;*****************************************************************
 8865                                  ;*TEST 111      OVERLAPPED OPERATIONS
 8866                                  ;*
 8867                                  ;*      DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B.  ISSUE  A
 8868                                  ;*      SEEK ON DRIVE A TO CYLINDER 1.  IMMEDIATELY ISSUE A WRITE
 8869                                  ;*      DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B.
 8870                                  ;*      AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD
 8871                                  ;*      BE SET AND DRIVE A HAS ATTENTION SET.
 8872                                  ;*
 8873                                  ;*      REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.
 8874                                  ;*
 8875                                  ;*      NOTE:   IF ONLY ONE DRIVE IS AVAILABLE THE
 8876                                  ;*              TEST WILL NOT BE DONE.
 8877                                  ;*
 8878                                  ;;*****************************************************************
 8879  031212  000004                  TST111: SCOPE
 8880  031214  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
 8881
 8882  031222  012737  177777  001224  2$:     MOV     #-1,$TMP1       ;SET LOOP CONTROL FLAG
```

```
 8883  031230  013705  001354          MOV     $DEVM,R5        ;GET DEVICE MAP
 8884  031234  005000                  CLR     R0              ;CLEAR FOR DRIVE #A
 8885  031236  005001                  CLR     R1              ;CLEAR FOR DRIVE #B
 8886  031240  012703  000001          MOV     #1,R3           ;SET DRIVE POSITION A
 8887  031244  012704  000001          MOV     #1,R4           ;SET DRIVE POSITION B
 8888  031250  012737  031260  001110  MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
 8889  031256  000545                  BR      11$             ;GO SET UP POINTERS
 8890  031260                  3$:
 8891  031260  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
 8892  031262  104003                  ERROR   3               ;BAD INIT ERROR
 8893
 8894  031264  010037  001626          MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
 8895  031270  010037  001610          MOV     R0,L.CS2        ;SETUP DRIVE A TO RECAL
 8896  031274  012737  000113  001600  MOV     #RECAL,L.CS1
 8897
 8898  031302  104417                  TLOADRK                 ;LOAD RK REGS
 8899  031304  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8900  031306  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8901  031310  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
 8902
 8903  031314  104437                  TWAT8S                  ;WAIT FOR SECOND INTERRUPT
 8904  031316  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8905
 8906  031320  012737  000105  001600  MOV     #CLEAR,L.CS1    ;SET UP TO CLEAR DRIVE
 8907  031326  104417                  TLOADRK                 ;LOAD RK REGS
 8908  031330  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8909  031332  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8910  031334                  4$:
 8911  031334  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8912  031336  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8913  031340  032737  040000  001540  BIT     #DI,T.CS1       ;TEST IF DI STILL SET
 8914  031346  001372                  BNE     4$              ;YES - LOOP
 8915  031350  010137  001626          MOV     R1,DRVNUM       ;STORE DRIVE FOR REPORT
 8916  031354  010137  001610          MOV     R1,L.CS2        ;SETUP DRIVE B TO RECAL
 8917  031360  012737  000113  001600  MOV     #RECAL,L.CS1
 8918
 8919  031366  104417                  TLOADRK                 ;LOAD RK REGS
 8920  031370  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8921  031372  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8922  031374  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
 8923  031400  104437                  TWAT8S                  ;WAIT FOR SECOND INTERRUPT
 8924  031402  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8925  031404  012737  000105  001600  MOV     #CLEAR,L.CS1    ;SET UP DRIVE CLEAR
 8926  031412  104417                  TLOADRK                 ;LOAD RK REGS
 8927  031414  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8928  031416  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8929  031420                  5$:
 8930  031420  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8931  031422  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8932  031424  032737  040000  001540  BIT     #DI,T.CS1       ;TEST IF DI STILL SET
 8933  031432  001372                  BNE     5$              ;YES - LOOP
 8934
 8935  031434  010037  001626          MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
 8936  031440  010037  001610          MOV     R0,L.CS2        ;SETUP DRIVE A TO SEEK
 8937  031444  012737  000001  001614  MOV     #1,L.DCYL       ;TO CYL 1
 8938  031452  012737  000117  001600  MOV     #SEEK,L.CS1
```

```
8939
8940  031460  104417                          TLOADRK                 ;LOAD RK REGS
8941  031462  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8942  031464  104002                          ERROR   2               ;TC SLOW/NOT COMPLETE ERROR
8943
8944  031466  010137  001626                  MOV     R1,DRVNUM       ;STORE DRIVE FOR REPORT
8945  031472  010137  001610                  MOV     R1,L.CS2        ;SETUP DRIVE B TO WRITE DATA
8946  031476  012737  000100  001614          MOV     #100,L.DCYL     ;AT CYL 100
8947  031504  012737  177400  001602          MOV     #-400,L.WC      ;400 WORDS
8948  031512  012737  063414  001604          MOV     #OBUFF,L.BA
8949  031520  012737  000123  001600          MOV     #WRDATA,L.CS1
8950
8951  031526  104417                          TLOADRK                 ;LOAD RK REGS-DO WRITE
8952
8953  031530  104427                          TWAT80                  ;WAIT FOR INTERRUPT
8954  031532  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8955
8956  031534  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8957  031536  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8958
8959  031540  010037  001626                  MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
8960  031544  130337  001557                  BITB    R3,T.ASOF+1     ;CHECK IF DRIVE ATTENTION IS DRIVE A
8961  031550  001007                          BNE     10$             ;YES-SKIP
8962  031552  012737  056526  001460          MOV     #EMDA,EM11N     ;"DRIVE ATTENTION NOT SET RESULT OF
8963  031560  012737  047752  061010          MOV     #EMSK,DF011A    ;SEEK"
8964  031566  104011                          ERROR   11
8965
8966  031570  104415                  10$:    SCOP1                   ;LOCAL LOOP TO 3$
8967
8968                                          ;       THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
8969                                          ;       OF DRIVES AVAILABLE.  THE FIRST PASS OF THE PROGRAM WILL USE THE
8970                                          ;       LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
8971                                          ;       B.  THE SECOND PASS SWAPS DRIVE A & B.  THE THIRD PASS USES
8972                                          ;       THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
8973                                          ;       AS A.  THE FORTH PASS SWAPS A & B AGAIN.  THIS CONTINUES
8974                                          ;       UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
8975                                          ;       DRIVE.
8976                                          ;
8977                                          ;       THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
8978                                          ;       THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
8979                                          ;
8980                                          ;       THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
8981                                          ;       CHECKED.
8982
8983  031572  005237  001224          11$:    INC     $TMP1           ;INCREMENT PASS CONTROL
8984  031576  001024                          BNE     16$             ;SKIP IF NOT ZERO
8985                                                                  ;(IT WILL BE ZERO ON THE 1ST PASS)
8986
8987  031600  030305                  12$:    BIT     R3,R5           ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8988  031602  001006                          BNE     13$             ;YES-SKIP
8989
8990  031604  005200                  22$:    INC     R0              ;BUMP R0 (DRIVE A)
8991  031606  006303                          ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
8992  031610  032703  000400                  BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
8993  031614  001771                          BEQ     12$             ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8994  031616  000464                          BR      50$             ;DONE-EXIT
```

```
 8995
 8996  031620  010001                 13$:    MOV     R0,R1           ;SET DRIVE B TO THE SAME AS A
 8997  031622  010304                         MOV     R3,R4
 8998  031624  005201                 14$:    INC     R1              ;BUMP R1 (DRIVE B)
 8999  031626  006304                         ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
 9000  031630  030405                         BIT     R4,R5           ;IS THIS DRIVE AVAIL?
 9001  031632  001004                         BNE     15$             ;YES-SKIP
 9002  031634  032704  000400               BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
 9003  031640  001771                         BEQ     14$             ;NO-LOOP
 9004  031642  000452                         BR      50$             ;DONE-EXIT
 9005
 9006  031644  000137  031260         15$:    JMP     3$              ;GO DO THE TEST ON THE DRIVE A & B
 9007                                                                 ;CONTAINED IN R0 & R1
 9008  031650  032737  000001  001224 16$:    BIT     #BIT0,$TMP1     ;IS PASS FLAGS ODD?
 9009  031656  001410                         BEQ     17$             ;NO-SKIP
 9010
 9011  031660  010046                         MOV     R0,-(SP)        ;
 9012  031662  010346                         MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
 9013  031664  010403                         MOV     R4,R3           ;TO EXCHANGE DRIVE A & B
 9014  031666  010100                         MOV     R1,R0
 9015  031670  012604                         MOV     (SP)+,R4
 9016  031672  012601                         MOV     (SP)+,R1
 9017  031674  000137  031260               JMP     3$              ;REPEAT TEST ON THIS COMBO.
 9018
 9019  031700  032737  000002  001224 17$:    BIT     #BIT1,$TMP1     ;TEST IF PASS FLAGS AT HALF MODULE 4?
 9020  031706  001410                         BEQ     19$             ;NO-SKIP TO BUMP DRIVE B
 9021  031710  005200                 18$:    INC     R0              ;BUMP DRIVE A
 9022  031712  006303                         ASL     R3              ;SHIFT DRIVE SELECT BIT
 9023  031714  030305                         BIT     R3,R5           ;AVAILABLE?
 9024  031716  001014                         BNE     20$             ;YES-SKIP
 9025  031720  032703  000400               BIT     #BIT8,R3        ;ALL CHECKED?
 9026  031724  001771                         BEQ     18$             ;NO-SKIP
 9027  031726  000412                         BR      21$             ;GO TO NEXT PASS
 9028
 9029  031730  005201                 19$:    INC     R1              ;BUMP DRIVE B
 9030  031732  006304                         ASL     R4              ;SHIFT DRIVE SELECT BIT
 9031  031734  030405                         BIT     R4,R5           ;AVAILABLE?
 9032  031736  001004                         BNE     20$             ;YES-SKIP
 9033  031740  032704  000400               BIT     #BIT8,R4        ;ALL CHECKED?
 9034  031744  001771                         BEQ     19$             ;NO-LOOP
 9035  031746  000404                         BR      23$             ;YES-SKIP TO NEXT PASS
 9036
 9037  031750  000137  031260         20$:    JMP     3$              ;GO TEST THIS COMBO
 9038
 9039  031754  010100                 21$:    MOV     R1,R0           ;SET DRIVE 0 TO LOW POSITION THIS PASS
 9040  031756  010403                         MOV     R4,R3           ;SET SELECT BITS TO AGREE
 9041  031760  005037  001224         23$:    CLR     $TMP1           ;CLEAR PASS FLAGS
 9042  031764  000137  031604               JMP     22$             ;GO SET UP A & B
 9043  031770                         50$:
```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;**************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
                                ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO TST1

031770                          $EOP:
031770  000004                          SCOPE
031772  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
031776  005037  001262                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
032002  005237  001304                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
032006  042737  100000  001304          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
032014  005327                          DEC     (PC)+           ;;LOOP?
032016  000001                  $EOPCT: .WORD   1
032020  003063                          BGT     $DOAGN          ;;YES
032022  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
032024  000001                  $ENDCT: .WORD   1
032026  032016                          $EOPCT
032030  104401  032036                  TYPE    ,65$            ;;TYPE ASCIZ STRING
032034  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
032054                          64$:
032054  013746  001304                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
032060  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
032062  104401  032070                  TYPE    ,67$            ;;TYPE ASCIZ STRING
032066  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
032132                          66$:
032132  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
032136  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
032140  104401  001273                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
032144  005037  001112                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
032150  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
032154  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
032156  000005                          RESET                   ;;CLEAR THE WORLD
032160  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
032162  000240                          NOP                     ;;SAVE ROOM
032164  000240                          NOP                     ;;FOR
032166  000240                          NOP                     ;;ACT11
032170                          $DOAGN:
032170  000137                          JMP     @(PC)+          ;;RETURN
032172  003144                  $RTNAD: .WORD   TST1
032174    377     377     000   $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
          032200                        .EVEN
                                .SBTTL  ROUTINE TO SIZE MEMORY

                                ;;**************************************************************
                                ;*CALL:
                                ;*      JSR     PC,$SIZE
                                ;*      RETURN
                                ;*$LSTAD WILL CONTAIN:
```

```
 9100                                   ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
 9101                                   ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
 9102                                   ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
 9103                                   ;*$KT11 IS THE MEMORY MANAGEMENT KEY
 9104                                   ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
 9105                                   ;*      MUST BE SETUP BEFORE THE CALL
 9106                                   ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
 9107                                   ;*      DETERMINED BY ROUTINE
 9108
 9109  032200  010046                  $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
 9110  032202  010146                          MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 9111  032204  010246                          MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
 9112  032206  010346                          MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
 9113  032210  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
 9114  032214  013746  000006                  MOV     @#ERRVEC+2,-(SP)
 9115  032220  010600                          MOV     SP,R0           ;;SAVE THE STACK POINTER
 9116                                  ;;SET THE ERRVEC PS TO THE PRESENT PS
 9117  032222  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
 9118  032224  012637  000006                  MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
 9119  032230  012701  003776                  MOV     #3776,R1        ;;SETUP ADDRESS
 9120  032234  105727                          TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
 9121  032236  000200                  $KT11:  .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
 9122  032240  100062                          BPL     $CORE           ;;BR IF NO
 9123  032242  012737  032400  000004          MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
 9124  032250  005737  177572                  TST     @#SR0           ;;KT11 ARE YOU THERE?
 9125  032254  052737  100000  032236          BIS     #100000,$KT11   ;;YES--SET KT11 KEY
 9126  032262  005046                          CLR     -(SP)           ;;INITIALIZE FOR "PAR" LOADING
 9127  032264  012702  172340                  MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST "PAR"
 9128  032270  012703  000010                  MOV     #^D8,R3         ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
 9129  032274  012762  077406  177740  1$:     MOV     #77406,-40(R2)  ;;PDR = 4K, UP, READ/WRITE
 9130  032302  011622                          MOV     (SP),(R2)+      ;;LOAD "PAR"
 9131  032304  062716  000200                  ADD     #200,(SP)       ;;UPDATE FOR NEXT "PAR"
 9132  032310  077307                          SOB     R3,1$           ;;LOOP UNTIL ALL EIGHT ARE LOADED
 9133  032312  012742  177600                  MOV     #177600,-(R2)   ;;SETUP KIPAR7 FOR I/O
 9134  032316  005042                          CLR     -(R2)           ;;SETUP KIPAR6 FOR TESTING
 9135  032320  012737  032336  000004          MOV     #2$,@#ERRVEC    ;;CATCH TIMEOUT IF NO SR3
 9136  032326  012737  000020  172516          MOV     #20,@#SR3       ;;ENABLE 22 BIT MODE
 9137  032334  000401                          BR      3$              ;;THIS PDP-11 HAS A SR3 REGISTER
 9138  032336  022626                  2$:     CMP     (SP)+,(SP)+     ;;CLEAN OFF THE STACK--NO SR3
 9139  032340  005237  177572          3$:     INC     @#SR0           ;;TURN ON MEMORY MANAGEMENT
 9140  032344  012737  032370  000004          MOV     #$KTOUT,@#ERRVEC ;;SET FOR TIME OUT
 9141  032352  005737  143776          4$:     TST     @#143776        ;;TRAP ON NON-EX-MEM
 9142  032356  062712  000040                  ADD     #40,(R2)        ;;MAKE A 1K STEP
 9143  032362  023712  172356                  CMP     @#KIPAR7,(R2)   ;;LAST ONE?
 9144  032366  101371                          BHI     4$              ;;NO--TRY IT
 9145  032370  011202                  $KTOUT: MOV     (R2),R2         ;;GET LAST BANK+1
 9146  032372  005037  177572                  CLR     @#SR0           ;;TURN OFF MEMORY MANAGEMENT
 9147  032376  000421                          BR      $SIZEX
 9148  032400  042737  100000  032236  $KTNEX: BIC     #100000,$KT11   ;;KT11 NON-EXISTENT
 9149  032406  012737  032436  000004  $CORE:  MOV     #$CROUT,@#ERRVEC ;;SET FOR TIMEOUT
 9150  032414  005002                          CLR     R2              ;;SET UP BANK
 9151  032416  062701  004000          1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
 9152  032422  062702  000040                  ADD     #40,R2          ;;1K STEP
 9153  032426  005711                          TST     (R1)            ;;TRAP ON TIME OUT
 9154  032430  022701  177776                  CMP     #177776,R1      ;;LAST ONE
 9155  032434  001370                          BNE     1$              ;;NO--TRY AGAIN
```

```
 9156  032436  162701  004000          $CROUT: SUB     #4000,R1
 9157  032442  162702  000040          $SIZEX: SUB     #40,R2          ;;DROP BACK
 9158  032446  010006                          MOV     R0,SP           ;;RESTORE THE STACK
 9159  032450  012637  000006                  MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
 9160  032454  012637  000004                  MOV     (SP)+,@#ERRVEC
 9161  032460  010137  032502                  MOV     R1,$LSTAD       ;;LAST ADDRESS
 9162  032464  010237  032504                  MOV     R2,$LSTBK       ;;LAST BANK
 9163  032470  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 9164  032472  012602                          MOV     (SP)+,R2        ;;RESTORE R2
 9165  032474  012601                          MOV     (SP)+,R1        ;;RESTORE R1
 9166  032476  012600                          MOV     (SP)+,R0        ;;RESTORE R0
 9167  032500  000207                          RTS     PC
 9168  032502  000000          $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
 9169  032504  000000          $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
 9170                          .SBTTL  SCOPE HANDLER ROUTINE
 9171
 9172                          ;;**************************************************************
 9173                          ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 9174                          ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 9175                          ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 9176                          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 9177                          ;*SW14=1        LOOP ON TEST
 9178                          ;*SW11=1        INHIBIT ITERATIONS
 9179                          ;*SW09=1        LOOP ON ERROR
 9180                          ;*SW08=1        LOOP ON TEST IN SWR<7:0>
 9181                          ;*CALL
 9182                          ;*      SCOPE           ;;SCOPE=IOT
 9183
 9184  032506                  $SCOPE:
 9185  032506  104407                  CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
 9186                          ;;GO TO ERROR ROUTINE IF RETURN PC LESS THAN 1002
 9187                          ;;OTHERWISE CONTINUE
 9188  032510  021627  001002          CMP     (SP),#1002      ;;UNEXPECTED TRAP OR INTERRRUPT
 9189  032514  101002                  BHI     1$              ;;ARE TRAPPED HERE VIA IOT
 9190  032516  000137  033526          JMP     $ERROR          ;;GO PROCESS UNEXPECTED TRAP
 9191  032522  032777  040000  146410 1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 9192  032530  001131                  BNE     $OVER           ;;YES IF SW14=1
 9193                          ;#####START OF CODE FOR THE XOR TESTER#####
 9194  032532  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 9195                                                          ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 9196  032534  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 9197  032540  012737  032560  000004  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 9198  032546  005737  177060          TST     @#177060        ;;TIME OUT ON XOR?
 9199  032552  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9200  032556  000500                  BR      $SVLAD          ;;GO TO THE NEXT TEST
 9201  032560  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 9202  032562  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9203  032566  000440                  BR      7$              ;;LOOP ON THE PRESENT TEST
 9204  032570                  6$:;#####END OF CODE FOR THE XOR TESTER#####
 9205  032570  032777  000400  146342          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
 9206  032576  001421                  BEQ     2$              ;;BR IF NO
 9207  032600  005046                  CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
 9208  032602  117716  146332          MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
 9209  032606  001414                  BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
 9210  032610  022716  000111          CMP     #111,(SP)       ;;CHECK THE NUMBER IN THE SWR
 9211  032614  002411                  BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
```

```
9212  032616  011637  001102            MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
9213  032622  005316                    DEC     (SP)            ;;BACKUP BY ONE
9214  032624  006316                    ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
9215  032626  062716  033032            ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
9216  032632  013637  001106            MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
9217  032636  000466                    BR      $OVER           ;;GO LOOP ON THE TEST
9218  032640  005726            8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
9219  032642  105737  001103    2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
9220  032646  001421                    BEQ     3$              ;;BR IF NO
9221  032650  123737  001115  001103    CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
9222  032656  101015                    BHI     3$              ;;BR IF NO
9223  032660  032777  001000  146252    BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
9224  032666  001404                    BEQ     4$              ;;BR IF NO
9225  032670  013737  001110  001106  7$:  MOV   $LPERR,$LPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
9226  032676  000446                    BR      $OVER
9227  032700  105037  001103    4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
9228  032704  005037  001262            CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
9229  032710  000415                    BR      1$              ;;ESCAPE TO THE NEXT TEST
9230  032712  032777  004000  146220  3$:  BIT   #BIT11,@SWR     ;;INHIBIT ITERATIONS?
9231  032720  001011                    BNE     1$              ;;BR IF YES
9232  032722  005737  001304            TST     $PASS           ;;IF FIRST PASS OF PROGRAM
9233  032726  001406                    BEQ     1$              ;;      INHIBIT ITERATIONS
9234  032730  005237  001104            INC     $ICNT           ;;INCREMENT ITERATION COUNT
9235  032734  023737  001262  001104    CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
9236  032742  002024                    BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
9237  032744  012737  000001  001104  1$:  MOV   #1,$ICNT       ;;REINITIALIZE THE ITERATION COUNTER
9238  032752  013737  033030  001262    MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
9239  032760  105237  001102    $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
9240  032764  113737  001102  001302    MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
9241  032772  011637  001106            MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
9242  032776  011637  001110            MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
9243  033002  005037  001264            CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
9244  033006  112737  000001  001115    MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9245  033014  013777  001102  146120  $OVER:  MOV   $TSTNM,@DISPLAY  ;;DISPLAY TEST NUMBER
9246  033022  013716  001106            MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
9247  033026  000002                    RTI                     ;;FIXES PS
9248  033030  003720            $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
9249  033032                    $SW08TBL:
9250  033032  003146                    .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
9251  033034  003264                    .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
9252  033036  003352                    .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
9253  033040  004470                    .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
9254  033042  004572                    .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
9255  033044  004750                    .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
9256  033046  005076                    .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
9257  033050  005202                    .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
9258  033052  006126                    .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
9259  033054  006244                    .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
9260  033056  006354                    .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
9261  033060  006616                    .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
9262  033062  006750                    .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
9263  033064  007160                    .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
9264  033066  007350                    .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
9265  033070  007472                    .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
9266  033072  010262                    .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
9267  033074  010660                    .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
```

```
9268  033076  011072                .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
9269  033100  011310                .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
9270  033102  011434                .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
9271  033104  011566                .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
9272  033106  011744                .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
9273  033110  012076                .WORD   TST30+2         ;;STARTING ADDRESS OF TEST 30
9274  033112  012372                .WORD   TST31+2         ;;STARTING ADDRESS OF TEST 31
9275  033114  012566                .WORD   TST32+2         ;;STARTING ADDRESS OF TEST 32
9276  033116  013022                .WORD   TST33+2         ;;STARTING ADDRESS OF TEST 33
9277  033120  013210                .WORD   TST34+2         ;;STARTING ADDRESS OF TEST 34
9278  033122  013442                .WORD   TST35+2         ;;STARTING ADDRESS OF TEST 35
9279  033124  013630                .WORD   TST36+2         ;;STARTING ADDRESS OF TEST 36
9280  033126  014052                .WORD   TST37+2         ;;STARTING ADDRESS OF TEST 37
9281  033130  014240                .WORD   TST40+2         ;;STARTING ADDRESS OF TEST 40
9282  033132  014432                .WORD   TST41+2         ;;STARTING ADDRESS OF TEST 41
9283  033134  014626                .WORD   TST42+2         ;;STARTING ADDRESS OF TEST 42
9284  033136  015032                .WORD   TST43+2         ;;STARTING ADDRESS OF TEST 43
9285  033140  015236                .WORD   TST44+2         ;;STARTING ADDRESS OF TEST 44
9286  033142  016102                .WORD   TST45+2         ;;STARTING ADDRESS OF TEST 45
9287  033144  016306                .WORD   TST46+2         ;;STARTING ADDRESS OF TEST 46
9288  033146  016512                .WORD   TST47+2         ;;STARTING ADDRESS OF TEST 47
9289  033150  016730                .WORD   TST50+2         ;;STARTING ADDRESS OF TEST 50
9290  033152  017134                .WORD   TST51+2         ;;STARTING ADDRESS OF TEST 51
9291  033154  017340                .WORD   TST52+2         ;;STARTING ADDRESS OF TEST 52
9292  033156  017464                .WORD   TST53+2         ;;STARTING ADDRESS OF TEST 53
9293  033160  017610                .WORD   TST54+2         ;;STARTING ADDRESS OF TEST 54
9294  033162  017734                .WORD   TST55+2         ;;STARTING ADDRESS OF TEST 55
9295  033164  020060                .WORD   TST56+2         ;;STARTING ADDRESS OF TEST 56
9296  033166  020204                .WORD   TST57+2         ;;STARTING ADDRESS OF TEST 57
9297  033170  020250                .WORD   TST60+2         ;;STARTING ADDRESS OF TEST 60
9298  033172  020322                .WORD   TST61+2         ;;STARTING ADDRESS OF TEST 61
9299  033174  020600                .WORD   TST62+2         ;;STARTING ADDRESS OF TEST 62
9300  033176  020762                .WORD   TST63+2         ;;STARTING ADDRESS OF TEST 63
9301  033200  021374                .WORD   TST64+2         ;;STARTING ADDRESS OF TEST 64
9302  033202  021562                .WORD   TST65+2         ;;STARTING ADDRESS OF TEST 65
9303  033204  022266                .WORD   TST66+2         ;;STARTING ADDRESS OF TEST 66
9304  033206  022476                .WORD   TST67+2         ;;STARTING ADDRESS OF TEST 67
9305  033210  022602                .WORD   TST70+2         ;;STARTING ADDRESS OF TEST 70
9306  033212  023224                .WORD   TST71+2         ;;STARTING ADDRESS OF TEST 71
9307  033214  023646                .WORD   TST72+2         ;;STARTING ADDRESS OF TEST 72
9308  033216  023756                .WORD   TST73+2         ;;STARTING ADDRESS OF TEST 73
9309  033220  024610                .WORD   TST74+2         ;;STARTING ADDRESS OF TEST 74
9310  033222  025266                .WORD   TST75+2         ;;STARTING ADDRESS OF TEST 75
9311  033224  025502                .WORD   TST76+2         ;;STARTING ADDRESS OF TEST 76
9312  033226  025760                .WORD   TST77+2         ;;STARTING ADDRESS OF TEST 77
9313  033230  026040                .WORD   TST100+2                ;;STARTING ADDRESS OF TEST 100
9314  033232  026232                .WORD   TST101+2                ;;STARTING ADDRESS OF TEST 101
9315  033234  026456                .WORD   TST102+2                ;;STARTING ADDRESS OF TEST 102
9316  033236  026604                .WORD   TST103+2                ;;STARTING ADDRESS OF TEST 103
9317  033240  027126                .WORD   TST104+2                ;;STARTING ADDRESS OF TEST 104
9318  033242  027660                .WORD   TST105+2                ;;STARTING ADDRESS OF TEST 105
9319  033244  030062                .WORD   TST106+2                ;;STARTING ADDRESS OF TEST 106
9320  033246  030302                .WORD   TST107+2                ;;STARTING ADDRESS OF TEST 107
9321  033250  030516                .WORD   TST110+2                ;;STARTING ADDRESS OF TEST 110
9322  033252  031214                .WORD   TST111+2                ;;STARTING ADDRESS OF TEST 111
9323  033254  031772                .WORD   $EOP+2          ;ADDRESS OF END OF PASS
```

```
 9324  033256  043652                          .WORD   ABTFAIL+2       ;ADDRESS OF ABORT FAILURE HANDLER
 9325
 9326                                          .SBTTL  APT COMMUNICATIONS ROUTINE
 9327
 9328                                          ;;**********************************************************************
 9329  033260  112737  000001  033524  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 9330  033266  112737  000001  033522  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 9331  033274  000403                          BR      $ATYC
 9332  033276  112737  000001  033524  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 9333  033304                          $ATYC:
 9334  033304  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9335  033306  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9336  033310  105737  033522                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 9337  033314  001450                          BEQ     5$              ;;IF NOT:  BR
 9338  033316  122737  000001  001316          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 9339  033324  001031                          BNE     3$              ;;IF NOT:  BR
 9340  033326  132737  000100  001317          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 9341  033334  001425                          BEQ     3$              ;;IF NOT:  BR
 9342  033336  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 9343  033342  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9344  033350  005737  001276          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 9345  033354  001375                          BNE     1$              ;;IF NOT:  WAIT
 9346  033356  010037  001312                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 9347  033362  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 9348  033364  001376                          BNE     2$
 9349  033366  163700  001312                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 9350  033372  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 9351  033374  010037  001314                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 9352  033400  012737  000004  001276          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 9353  033406  000413                          BR      5$
 9354  033410  017637  000004  033434  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 9355  033416  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 9356  033424  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 9357  033430  004737  043662                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
 9358  033434  000000                  4$:     .WORD   0
 9359  033436                          5$:
 9360  033436  105737  033524          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9361  033442  001416                          BEQ     12$             ;;IF NOT:  BR
 9362  033444  005737  001316                  TST     $ENV            ;;RUNNING UNDER APT?
 9363  033450  001413                          BEQ     12$             ;;IF NOT:  BR
 9364  033452  005737  001276          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9365  033456  001375                          BNE     11$             ;;IF NOT:  WAIT
 9366  033460  017637  000004  001300          MOV     @4(SP),$FATAL   ;;GET ERROR #
 9367  033466  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9368  033474  005237  001276                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9369  033500  105037  033524          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 9370  033504  105037  033523                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 9371  033510  105037  033522                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9372  033514  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9373  033516  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9374  033520  000207                          RTS     PC              ;;RETURN
 9375  033522     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
 9376  033523     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
 9377  033524     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
 9378          033526                          .EVEN
 9379          000200                  APTSIZE=200
```

```
  9380            000001                  APTENV=001
  9381            000100                  APTSPOOL=100
  9382            000040                  APTCSUP=040
  9383                                    .SBTTL  ERROR HANDLER ROUTINE
  9384
  9385                                    ;;*****************************************************************
  9386                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
  9387                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
  9388                                    ;*AND GO TO TYPERR ON ERROR
  9389                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  9390                                    ;*SW15=1        HALT ON ERROR
  9391                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
  9392                                    ;*SW10=1        BELL ON ERROR
  9393                                    ;*SW09=1        LOOP ON ERROR
  9394                                    ;*CALL
  9395                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
  9396
  9397  033526                            $ERROR:
  9398  033526  104407                           CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
  9399  033530  105237  001103           7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
  9400  033534  001775                           BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
  9401  033536  013777  001102  145376         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
  9402  033544  032777  002000  145366         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
  9403  033552  001402                         BEQ     1$              ;;NO - SKIP
  9404  033554  104401  001266                 TYPE    ,$BELL          ;;RING BELL
  9405  033560  005237  001112           1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
  9406  033564  011637  001116                 MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
  9407  033570  162737  000002  001116         SUB     #2,$ERRPC
  9408  033576  117737  145314  001114         MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
  9409  033604  032777  020000  145326         BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
  9410  033612  001055                         BNE     20$             ;;SKIP TYPEOUTS
  9411  033614  021627  001002                 CMP     (SP),#1002      ;;IF RETURN PC LESS THAN 1002
  9412  033620  101046                         BHI     12$             ;;ERROR IS ILLEGAL TRAP
  9413                                    ;;PROCESS UNEXPECTED TRAP OR INTERRUPT
  9414  033622  016637  000004  001116         MOV     4(SP),$ERRPC    ;;GET PC AT TIME OF FALSE TRAP
  9415  033630  162737  000002  001116         SUB     #2,$ERRPC       ;;ADJUST PC
  9416  033636  104401  033702                 TYPE    ,10$            ;;TYPE HEADER
  9417  033642  013746  001116                 MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
  9418  033646  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  9419  033650  104401  033720                 TYPE    ,11$
  9420  033654  162716  000004                 SUB     #4,(SP)         ;;GET FALSE TRAP VECTOR ADDR
  9421  033660  011637  001116                 MOV     (SP),$ERRPC
  9422  033664  013746  001116                 MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
  9423  033670  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  9424  033672  104401  001273                 TYPE    ,$CRLF
  9425  033676  022626                         CMP     (SP)+,(SP)+     ;;POP FALSE TRAP VECTOR PC&ADDR
  9426  033700  000422                         BR      20$
  9427  033702  050200  036503  000040  10$:    .ASCIZ  <200>'PC= '
  9428  033710  020040  047125  054105  11$:    .ASCIZ  '  UNEXPECTED TRAP TO '
  9429  033716  042520  052103  042105
  9430  033724  052040  040522  020120
  9431  033732  047524  000040
  9432                                          .EVEN
  9433  033736                            12$:
  9434  033736  004737  034050                 JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
  9435  033742  104401  001273                 TYPE    ,$CRLF
```

```
9436  033746                                  20$:
9437  033746  122737  000001  001316                  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
9438  033754  001007                                  BNE     2$              ;;NO,SKIP APT ERROR REPORT
9439  033756  113737  001114  033770                  MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
9440  033764  004737  033276                          JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
9441  033770     000                          21$:    .BYTE   0
9442  033771     000                                  .BYTE   0
9443  033772  000777                          22$:    BR      22$             ;;APT ERROR LOOP
9444  033774  005777  145140                  2$:     TST     @SWR            ;;HALT ON ERROR
9445  034000  100002                                  BPL     3$              ;;SKIP IF CONTINUE
9446  034002  000000                                  HALT                    ;;HALT ON ERROR!
9447  034004  104407                                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
9448  034006  032777  001000  145124          3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
9449  034014  001402                                  BEQ     4$              ;;BR IF NO
9450  034016  013716  001110                          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
9451  034022  005737  001264                  4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
9452  034026  001402                                  BEQ     5$              ;;BR IF NONE
9453  034030  013716  001264                          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
9454  034034                                  5$:
9455  034034  022737  032160  000042                  CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
9456  034042  001001                                  BNE     6$              ;;BRANCH IF NO
9457  034044  000000                                  HALT                    ;;YES
9458  034046                                  6$:
9459  034046  000002                                  RTI                     ;;RETURN
9460                                          ;;**************************************************************
9461                                          .SBTTL  TYPE ERROR ROUTINE
9462                                          ;*ENTRY JSR PC,TYPERR
9463                                          ;*RETURN RTS PC
9464                                          ;*
9465                                          ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9466                                          ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
9467                                          ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
9468                                          ;*THE ERROR.
9469                                          ;;**************************************************************
9470  034050  104413                          TYPERR: SAVREG
9471  034052  113737  001102  001302                  MOVB    $TSTNM,$TESTN   ;GET TEST NUMBER FOR REPORT
9472  034060  042737  177400  001302                  BIC     #177400,$TESTN  ;CLEAR UNUSED BITS
9473  034066  113700  001114                          MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
9474  034072  042700  177400                          BIC     #177400,R0      ;CLEAR UNUSED BITS
9475  034076  005300                                  DEC     R0              ;FORM INDEX FOR ERROR TABLE
9476  034100  006300                                  ASL     R0
9477  034102  006300                                  ASL     R0
9478  034104  006300                                  ASL     R0
9479  034106  062700  001360                  1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
9480  034112  012037  034126                          MOV     (R0)+,2$        ;GET EM POINTER
9481  034116  001404                                  BEQ     3$              ;BRANCH IF THERE ISN'T ONE
9482  034120  104401  001273                          TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
9483  034124  104401                                  TYPE                    ;TYPE ERROR MESSAGE (EM)
9484  034126  000000                          2$:     .WORD   0               ;EM POINTER GOES HERE
9485  034130  012037  034144                  3$:     MOV     (R0)+,4$        ;GET DH POINTER
9486  034134  001404                                  BEQ     5$              ;BRANCH IF THERE ISN'T ONE
9487  034136  104401  001273                          TYPE    ,$CRLF          ;TYPE CR-LF
9488  034142  104401                                  TYPE                    ;TYPE DATA HEADER
9489  034144  000000                          4$:     .WORD   0               ;DH POINTER GOES HERE
9490  034146  012001                          5$:     MOV     (R0)+,R1        ;GET DT POINTER
9491  034150  001445                                  BEQ     20$             ;BRANCH IF THERE ARE NONE
```

```
 9492  034152  005004                        CLR     R4              ;SET INDENT SWITCH
 9493  034154  012000                        MOV     (R0)+,R0        ;GET DF POINTER
 9494  034156  012002                        MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
 9495  034160  104401  001273                TYPE    ,$CRLF
 9496  034164  112003                10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
 9497  034166  105720                        TSTB    (R0)+           ;BUMP PAST FORMAT WORD
 9498  034170  005703                        TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
 9499  034172  001416                        BEQ     14$             ;NO - SKIP DATA PRINT
 9500  034174  005704                        TST     R4              ;CHECK FOR INDENT
 9501  034176  001004                        BNE     12$             ;YES, GO INDENT
 9502  034200  013146                11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
 9503  034202  104402                        TYPOC                   ;TYPE IT
 9504  034204  005303                        DEC     R3              ;MORE DATA WORDS
 9505  034206  001403                        BEQ     13$             ;NO-BRANCH
 9506  034210  104401  050443          12$:  TYPE    ,SPACE2         ;TYPE SEPARATORS
 9507  034214  000771                        BR      11$             ;LOOP
 9508  034216  104401  001273          13$:  TYPE    ,$CRLF          ;TYPE <CR><LF>
 9509  034222  005710                        TST     (R0)            ;CHECK IF NEXT HEADER AVAILABLE
 9510  034224  001401                        BEQ     14$             ;NO, DO NOT CHANGE INDENT
 9511  034226  005104                        COM     R4              ;CHANGE INDENT
 9512  034230  005302                14$:    DEC     R2              ;MORE DH'S?
 9513  034232  003414                        BLE     20$             ;NO-BRANCH
 9514  034234  012037  034254        15$:    MOV     (R0)+,18$       ;GET NEXT DH POINTER
 9515  034240  001751                        BEQ     10$             ;IF 0 GET DATA
 9516  034242  005704                        TST     R4              ;INDENT?
 9517  034244  001402                        BEQ     17$             ;NO, BRANCH
 9518  034246  104401  050443                TYPE    ,SPACE2         ;TYPE INDENT
 9519  034252  104401                17$:    TYPE                    ;TYPE DH
 9520  034254  000000                18$:    .WORD   0               ;DH POINTER GOES HERE
 9521  034256  104401  001273                TYPE    ,$CRLF
 9522  034262  000740                        BR      10$             ;GET DATA
 9523  034264  104414                20$:    RESREG
 9524  034266  005237  001632                INC     ERRCNT          ;INCREMENT THE ERROR COUNT
 9525  034272  032777  010000  144640        BIT     #SW12,@SWR      ;CHECK IF SWITCH 12 SET
 9526  034300  001421                        BEQ     25$             ;NO, RETURN
 9527  034302  022737  000024  001632        CMP     #20.,ERRCNT     ;CHECK IF ERROR THRESHOLD EXCEEDED
 9528  034310  103015                        BHIS    25$             ;NO, RETURN
 9529  034312  104401  053215                TYPE    ,ABORT          ;TYPE "PROGRAM ABORTED BECAUSE
 9530                                                                ;ERROR THRESHOLD EXCEEDED"
 9531  034316  005737  000042                TST     42              ;CHECK IF CHAIN MODE
 9532  034322  001407                        BEQ     22$             ;NO, HALT PROCESSOR
 9533  034324  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK
 9534  034330  012737  000001  032016        MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
 9535  034336  000137  031770                JMP     $EOP
 9536  034342  000000                22$:    HALT
 9537  034344  000207                25$:    RTS     PC
 9538                                .SBTTL  NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER
 9539                                ;*      THIS ROUTINE SETS THE INTERRUPT FLAG AND DOES AN RTI.
 9540                                ;*      THIS IS THE INDICATION TO THE ROUTINE CHECKING
 9541                                ;*      NON-EXIXTANT MEMORY THAT AN INTERRUPT DID OCCUR.
 9542
 9543  034346  005237  001662        NEXINT: INC     INTSET          ;BUMP THE INTERRUPT COUNTER
 9544  034352  000002                        RTI
 9545
 9546                                .SBTTL  RK611 INTERRUPT HANDLER
 9547                                ;*      MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE.  ACTUAL
```

```
 9548                                   ;*      PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
 9549                                   ;*      PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
 9550                                   ;*      INTERRUPT OCCURRED.
 9551
 9552  034354  000240                   INTHLR: NOP
 9553  034356  005237  001662                   INC     INTSET          ;BUMP THE INTERRUPT FLAG
 9554  034362  000002                           RTI                     ;RETURN.
 9555
 9556                                   .SBTTL  MEMORY PARITY ERROR TRAP HANDLER
 9557                                   ;*      MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE. THE REPORT
 9558                                   ;*      WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.
 9559
 9560  034364  032777  020000  144546 PERHLR: BIT     #BIT13,@SWR     ;TEST IF INHIBIT REPORT
 9561  034372  001003                         BNE     1$              ;YES - SKIP
 9562  034374  104401  053444                 TYPE    ,EM3            ;TYPE PARITY ERROR MESSAGE
 9563  034400  104402                         TYPOC                   ;AND PC VALUE
 9564  034402  012737  000001  032016 1$:     MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
 9565  034410  012706  001100                 MOV     #STACK,SP       ;CLEAN OFF STACK
 9566  034414  000137  031770                 JMP     $EOP
 9567
 9568                                   .SBTTL  LINE CLOCK INTERRUPT HANDLER
 9569                                   ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
 9570                                   ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.
 9571
 9572  034420  005237  001674 LCKHLR: INC     LCLKTK          ;INCREMENT CLOCK TICK COUNTER
 9573  034424  042777  000200  145256         BIC     #BIT7,@KWLADD   ;CLEAR MONITOR BIT
 9574  034432  000002                         RTI
 9575
 9576                                   ;;*****************************************************************
 9577                                   .SBTTL  OPTION PRESENT TEST AND SETUP
 9578                                   ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
 9579                                   ;*      LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF PRESENT; CLEARED
 9580                                   ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
 9581  034434  104413                 OPTTST: SAVREG
 9582  034436  013746  000004                 MOV     ERRVEC,-(SP)    ;STORE OLD NEM CONTENTS
 9583  034442  013746  000006                 MOV     ERRVEC+2,-(SP)
 9584  034446  012737  034634  000004         MOV     #20$,ERRVEC     ;DET INTERRUPT FOR NEM
 9585  034454  012737  000340  000006         MOV     #PR7,ERRVEC+2   ;SET PRIORITY
 9586  034462  005037  001670                 CLR     MEMPAR          ;CLEAR PARITY WORD FLAGS
 9587  034466  042737  000100  001664         BIC     #PARPRE,OPTFLG  ;CLEAR PARITY PRESENT FLAG
 9588  034474  005737  170200                 TST     170200          ;TEST IF 11/70 REGISTER PRESENT
 9589  034500  000240                         NOP
 9590  034502  000240                         NOP
 9591  034504  012737  177750  001672         MOV     #177750,CSRPTR  ;SET POINTER FOR 11/70
 9592  034512  052737  002000  001664         BIS     #CP1170,OPTFLG  ;SET 11/70 FLAG
 9593  034520  052737  000100  001664         BIS     #PARPRE,OPTFLG  ;SET PARITY PRESENT FLAG
 9594  034526  000464                         BR      35$             ;GO TO VECTOR SETUP
 9595
 9596  034530  013703  001714         3$:     MOV     MEMBAS,R3       ;SET UP POINTER TO FIRST PARITY CSR
 9597  034534  012704  000001                 MOV     #1,R4           ;INIT MASK
 9598  034540  012713  000001         6$:     MOV     #1,(R3)         ;SET PARITY DETECT IN THAT CSR
 9599  034544  005713                         TST     (R3)
 9600  034546  050437  001670                 BIS     R4,MEMPAR       ;SET PARITY PRESENT BIT
 9601  034552  032737  000100  001664         BIT     #PARPRE,OPTFLG  ;WAS PARITY PRESENT SET BEFORE
 9602  034560  001017                         BNE     10$             ;YES - SKIP
 9603  034562  013700  001716                 MOV     MMVECA,R0       ;SET UP FOR PARITY TRAPS
```

```
 9604  034566  012720  034646          MOV     #40$,(R0)+      ;  TO 40$
 9605  034572  012710  000340          MOV     #PR7,(R0)
 9606  034576  012700  063640          MOV     #OBUFF+224,R0   ;SET POINTER TO BUFFER WHERE BAD PARITY
 9607                                                          ;  IS USED IN THE TESTS
 9608  034602  012713  000004          MOV     #BIT2,(R3)      ;SET TO WRITE WRONG PARITY
 9609  034606  012710  177777          MOV     #-1,(R0)        ;WRITE WORD BAD
 9610  034612  012713  000001          MOV     #1,(R3)         ;SET TO DETECT PARITY ERROR
 9611  034616  005710                  TST     (R0)            ;READ BAD WORD
 9612  034620  062703  000002  10$:    ADD     #2,R3           ;BUMP TO NEXT CSR
 9613  034624  000241                  CLC
 9614  034626  006104                  ROL     R4              ;SHIFT MASK
 9615  034630  001343                  BNE     6$              ;TEST IF ALL DONE
 9616  034632  000422                  BR      35$             ;YES - SKIP
 9617
 9618  034634  022626          20$:    CMP     (SP)+,(SP)+     ;CLEAR STACK
 9619  034636  012737  034674  000004  MOV     #30$,ERRVEC     ;SET NEW NEM TRAP ADDRESS
 9620  034644  000731                  BR      3$              ;GO TO CSR CHECKS
 9621
 9622  034646  022626          40$:    CMP     (SP)+,(SP)+     ;CLEAR STACK
 9623  034650  010337  001672          MOV     R3,CSRPTR       ;SET CSR POINTER FOR CSR TO BE USED
 9624  034654  052737  000100  001664  BIS     #PARPRE,OPTFLG  ;SET PARITY PRESENT FLAG
 9625  034662  005013                  CLR     (R3)            ;CLEAR CSR
 9626  034664  005010                  CLR     (R0)            ;CLEAR BAD WORD
 9627  034666  012713  000001          MOV     #1,(R3)         ;SET TO DETECT PARITY ERRORS
 9628  034672  000752                  BR      10$             ;CHECK NEXT CSR
 9629
 9630  034674  022626          30$:    CMP     (SP)+,(SP)+     ;CLEAR STACK
 9631  034676  000750                  BR      10$             ;GO CHECK NEXT CSR
 9632  034700  013700  001716  35$:    MOV     MMVECA,R0       ;SET UP POINTER TO PARITY VECTOR
 9633  034704  005737  001670          TST     MEMPAR          ;TEST IF ANY PARITY PRESENT
 9634  034710  001004                  BNE     37$             ;YES - SKIP
 9635  034712  032737  002000  001664  BIT     #CP1170,OPTFLG  ;TEST IF 11/70
 9636  034720  001405                  BEQ     39$             ;NO - SKIP
 9637  034722  012720  034364  37$:    MOV     #PERHLR,(R0)+   ;SET UP PARITY VECTOR
 9638  034726  012710  000340          MOV     #PR7,(R0)       ;SET PRIORITY
 9639  034732  000403                  BR      38$
 9640  034734  012720  000116  39$:    MOV     #116,(R0)+      ;ELSE RESTORE TRAP CATCHER
 9641  034740  005010                  CLR     (R0)
 9642  034742  012737  034346  000004  38$:    MOV     #NEXINT,ERRVEC  ;SET UP NEM VECTOR FOR LINE CLOCK TEST
 9643  034750  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT COUNTER
 9644  034754  013700  001712          MOV     KWLVEC,R0       ;SET POINTER TO VECTOR
 9645  034760  012720  034420          MOV     #LCKHLR,(R0)+   ;INSERT ADDRESS OF INTERRUPT HDLR
 9646  034764  012710  000340          MOV     #PR7,(R0)       ;INSERT PRIORITY
 9647  034770  012777  000100  144712  MOV     #BIT6,@KWLADD   ;LOAD KW11-L FOR INTERRUPT ENABLE
 9648  034776  000240                  NOP
 9649  035000  000240                  NOP
 9650  035002  000240                  NOP
 9651  035004  005737  001662          TST     INTSET          ;TEST IN NEM ON KW11-P REFERENCE
 9652  035010  001003                  BNE     4$              ;THIS BRANCH WILL BYPASS SET UP OF
 9653                                                          ;CLOCK OPTION
 9654  035012  052737  100000  001664  BIS     #LCLKPR,OPTFLG  ;SET CLOCK PRESENT FLAG
 9655  035020  012701  000006  4$:     MOV     #6,R1           ;RESTORE OLD VECTOR
 9656  035024  005037  001662          CLR     INTSET          ;CLEAR INT FLAG
 9657  035030  012637  000006          MOV     (SP)+,ERRVEC+2
 9658  035034  012637  000004          MOV     (SP)+,ERRVEC
 9659  035040  012746  000000          MOV     #PR0,-(SP)      ;RESTORE PRIORITY TO 0
```

```
 9660  035044  012746  035052            MOV     #12$,-(SP)
 9661  035050  000002                    RTI
 9662  035052                    12$:
 9663  035052  104414                    RESREG
 9664  035054  000207                    RTS     PC
 9665
 9666                            ;;*****************************************************************
 9667                            .SBTTL  LOOP ON INTERNAL ERROR
 9668                            ;*      THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER
 9669                            ;*      IS EXPECTED TO SET $LPERR TO THE START OF THE SCOPE LOOP
 9670                            ;*      TO BE EXECUTED ON ERROR.
 9671
 9672  035056  032777  001000  144054  SCOP1$: BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
 9673  035064  001405                    BEQ     5$              ;NO, CONTINUE
 9674  035066  105737  001103                    TSTB    $ERFLG          ;CHECK IF ERROR OCCURRED
 9675  035072  001402                    BEQ     5$
 9676  035074  013716  001110                    MOV     $LPERR,(SP)     ;LOAD ERROR RETURN
 9677  035100  000002                    5$:     RTI                     ;RETURN
 9678                            .SBTTL  LINE CLOCK CALIBRATE
 9679                            ;*      WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS
 9680                            ;*      TO A  MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS
 9681                            ;*      THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE
 9682                            ;*      LINE CLOCK SIMULATOR.
 9683
 9684  035102  005037  001674    CLKCAL: CLR     LCLKTK          ;CLEAR TICK COUNTER
 9685  035106  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
 9686  035114  001004                    BNE     1$              ;YES - SKIP
 9687  035116  012737  000020  001654          MOV     #16.,TIMCNT     ;ELSE PRESET TIMCNT
 9688  035124  000410                    BR      2$              ;AND EXIT
 9689  035126  005737  001662    1$:     TST     INTSET          ;TEST IF INTERRUPT HAS OCCURRED
 9690  035132  001005                    BNE     2$              ;YES- ABORT CALIBRATION
 9691  035134  005737  001674          TST     LCLKTK          ;WAIT FOR CLOCK TICK
 9692  035140  001772                    BEQ     1$              ;NOT YET - LOOP
 9693  035142  005037  001674          CLR     LCLKTK          ;CLEAR TICK COUNT
 9694  035146  000207            2$:     RTS     PC      ;RETURN
 9695
 9696                            .SBTTL  LINE CLOCK SIMULATION ROUTINE
 9697                            ;*      THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO
 9698                            ;*      DO THIS THE VALUE STORED IN MILCNT IS USED AS THE
 9699                            ;*      BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT
 9700                            ;*      AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE
 9701                            ;*      TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE
 9702                            ;*      LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT
 9703                            ;*      IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
 9704                            ;*      TICK).
 9705                            ;*
 9706                            ;*      THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND
 9707                            ;*      AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.
 9708                            ;*
 9709
 9710  035150  010046            MYTIME: MOV     R0,-(SP)        ;SAVE R0
 9711  035152  013700  001652          MOV     MILCNT,R0       ;SET COUNT
 9712  035156  005737  001662    1$:     TST     INTSET          ;TEST IF INTERRUPT SET
 9713  035162  001012                    BNE     2$              ;YES - SKIP
 9714  035164  005300                    DEC     R0              ;DECREMENT COUNT TO ZERO
 9715  035166  001373                    BNE     1$
```

```
 9716  035170  005337  001654                 DEC     TIMCNT          ;DECREMENT TIMCNT
 9717  035174  001005                         BNE     2$              ;IF NOT ZERO - EXIT
 9718  035176  005237  001674                 INC     LCLKTK          ;ELSE BUMP TICK COUNTER
 9719  035202  012737  000020  001654         MOV     #16.,TIMCNT     ;RESET TIME COUNT
 9720  035210  012600                 2$:     MOV     (SP)+,R0        ;RESTORE R0
 9721  035212  000207                         RTS     PC              ;RETURN
 9722
 9723                                 .SBTTL  WAIT FOR INTERRUPT ROUTINE
 9724                                 ;*      THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL
 9725                                 ;*      SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE
 9726                                 ;*      WHILE WAITING FOR INTERRUPT. IF INTERRUPT DOES NOT OCCUR
 9727                                 ;*      IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED
 9728                                 ;*      (BUT NOT PRINTED IN THE ROUTINE)AND THEN RETURNS TO THE
 9729                                 ;*      LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE
 9730                                 ;*      RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL
 9731                                 ;*      BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.
 9732
 9733  035214  104413                 IWAT8S: SAVREG          ;ENTRY FOR 8 SECOND WAIT
 9734  035216  012700  000764                 MOV     #500.,R0
 9735  035222  000463                         BR      WATSRT
 9736  035224  104413                 IWAT1M: SAVREG          ;ENTRY FOR 1 MIN WAIT
 9737  035226  012700  007246                 MOV     #3750.,R0
 9738  035232  000457                         BR      WATSRT
 9739  035234  104413                 IWAT2S: SAVREG  ;ENTRY FOR 2 SECOND WAIT
 9740  035236  012700  000200                 MOV     #128.,R0
 9741  035242  000453                         BR      WATSRT
 9742  035244  104413                 IWAT1S: SAVREG                  ;ENTRY FOR 1 SECOND WAIT
 9743  035246  012700  000077                 MOV     #63.,R0
 9744  035252  000447                         BR      WATSRT
 9745  035254  104413                 IWAT159:        SAVREG          ;ENTRY FOR 160 MS WAIT
 9746  035256  012700  000012                 MOV     #10.,R0
 9747  035262  000443                         BR      WATSRT
 9748  035264  104413                 IWAT144:        SAVREG          ;ENTRY FOR 144 MS WAIT
 9749  035266  012700  000011                 MOV     #9.,R0
 9750  035272  000437                         BR      WATSRT
 9751  035274  104413                 IWAT128:        SAVREG          ;ENTRY FOR 128 MS WAIT
 9752  035276  012700  000010                 MOV     #8.,R0
 9753  035302  000433                         BR      WATSRT
 9754  035304  104413                 IWAT112:        SAVREG          ;ENTRY FOR 112 MS WAIT
 9755  035306  012700  000007                 MOV     #7,R0
 9756  035312  000427                         BR      WATSRT
 9757  035314  104413                 IWAT96: SAVREG                  ;ENTRY FOR 96 MS WAIT
 9758  035316  012700  000006                 MOV     #6,R0
 9759  035322  000423                         BR      WATSRT
 9760  035324  104413                 IWAT80: SAVREG                  ;ENTRY FOR 80 MS WAIT
 9761  035326  012700  000005                 MOV     #5,R0
 9762  035332  000417                         BR      WATSRT
 9763  035334  104413                 IWAT64: SAVREG                  ;ENTRY FOR 64 MS WAIT
 9764  035336  012700  000004                 MOV     #4,R0
 9765  035342  000413                         BR      WATSRT
 9766  035344  104413                 IWAT48: SAVREG                  ;ENTRY FOR 48 MS WAIT
 9767  035346  012700  000003                 MOV     #3,R0
 9768  035352  000407                         BR      WATSRT
 9769  035354  104413                 IWAT32: SAVREG                  ;ENTRY FOR 32 MS WAIT
 9770  035356  012700  000002                 MOV     #2,R0
 9771  035362  000403                         BR      WATSRT
```

```
 9772  035364  104413                  IWAT16: SAVREG                  ;ENTRY FOR 16 MS WAIT
 9773  035366  012700  000001                  MOV     #1,R0
 9774  035372  012746  000000          WATSRT: MOV     #PR0,-(SP)      ;LOAD PRIORITY 0 ON STACK
 9775  035376  012746  035404                  MOV     #5$,-(SP)       ;LOAD ADDRESS
 9776  035402  000002                         RTI
 9777
 9778  035404  012737  000020  001654  5$:     MOV     #16.,TIMCNT     ;PRESET TIME COUNTER
 9779  035412  004737  035102                  JSR     PC,CLKCAL       ;GO CALIBRATE THE CLOCK
 9780  035416  005737  001662          1$:     TST     INTSET          ;TEST IF INTERRUPT OCCURRED
 9781  035422  001036                         BNE     3$              ;YES - EXIT
 9782  035424  032737  100000  001664         BIT     #LCLKPR,OPTFLG  ;TEST IF KW11-L AVAILABLE
 9783  035432  001002                         BNE     2$              ;YES - SKIP
 9784  035434  004737  035150                  JSR     PC,MYTIME       ;ELSE CALL SIMULATOR
 9785  035440  023700  001674          2$:     CMP     LCLKTK,R0       ;TEST IF ENOUGH TICKS COUNTED
 9786  035444  103764                         BLO     1$              ;NO - LOOP
 9787  035446  104420                         TGETRK                  ;ELSE GET '611 REGS
 9788  035450  013701  001540                  MOV     T.CS1,R1        ;PUT CS1 IN R1- STRIP ALL BUT
 9789  035454  042701  177741                  BIC     #177741,R1      ;COMMAND CODE; INDEX INTO TABLE
 9790  035460  016137  047600  001372         MOV     CMNDLB(R1),DH2N ;AND SELECT HEADER TO IDENTIFY OPERATION
 9791  035466  012737  053514  001370         MOV     #EM4,EM2N       ;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
 9792  035474  013700  001302                  MOV     $TESTN,R0       ;GET NUMBER OF PRESENT TEST
 9793  035500  006300                         ASL     R0              ;SHIFT FOR INDEX
 9794  035502  016037  033032  001264         MOV     $SW08TB(R0),$ESCAPE  ;LOAD ESCAPE TO ABORT TESTS
 9795  035510  162737  000002  001264         SUB     #2,$ESCAPE      ;BUT GO TO NEXT SCOPE CALL
 9796  035516  000402                         BR      4$
 9797  035520  062716  000002          3$:     ADD     #2,(SP)         ;BUMP RETURN AROUND ERROR
 9798  035524  104414                  4$:     RESREG                  ;RESTORE REGS
 9799  035526  000002                         RTI                     ;RETURN
 9800
 9801                                  .SBTTL  "L" REGISTER LOADING ROUTINE
 9802                                  ;*      THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
 9803                                  ;*      THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT
 9804                                  ;*      LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED
 9805                                  ;*      FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.
 9806                                  ;*
 9807                                  ;*      CALL:   JSR     R4,LRLOAD
 9808                                  ;*              COMMAND
 9809                                  ;*              WORD COUNT
 9810                                  ;*              BUS ADDRESS
 9811                                  ;*              .BYTE           SECTOR ADDRESS
 9812                                  ;*              .BYTE           TRACK ADDRESS
 9813                                  ;*              CYLINDER ADDRESS
 9814
 9815  035530                          LRLOAD:
 9816  035530  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9817  035532  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9818  035534  012701  001600                  MOV     #L.CS1,R1       ;GET ADDRESS OF L REGS
 9819  035540  012700  000004                  MOV     #4,R0           ;PRESET COUNT
 9820  035544  012421          1$:     MOV     (R4)+,(R1)+     ;MOVE FIRST FOUR WORDS INTO "L" REGS
 9821  035546  005300                         DEC     R0              ;CS1, WC, BA, DA
 9822  035550  001375                         BNE     1$
 9823  035552  013721  001626                  MOV     DRVNUM,(R1)+    ;LOAD DRIVE NUMBER
 9824  035556  005721                         TST     (R1)+           ;BUMP PAST ASOF
 9825  035560  012411                         MOV     (R4)+,(R1)      ;LOAD DCYL
 9826  035562  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 9827  035564  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
```

```
 9828  035566  000204                          RTS     R4
 9829
 9830                                  .SBTTL  LOAD RK611 FOR OPERATION
 9831                                  ;*      THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
 9832                                  ;*      THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
 9833                                  ;*      OF THE REGISTER CONTENTS.  L.CS1 IS TRANSFERRED LAST AS IT
 9834                                  ;*      SHOULD BE IF THE GO BIT IS SET.
 9835  035570  005037  001662          LOADRK: CLR     INTSET          ;CLEAR INTERRUPT FLAG.
 9836  035574  010046                          MOV     R0,-(SP)        ;STORE REGISTER
 9837  035576  010146                          MOV     R1,-(SP)
 9838  035600  012700  001602                  MOV     #L.WC,R0        ;GET ADDRESS OF SETUP STORAGE WC
 9839  035604  010201                          MOV     R2,R1           ;GET BASE ADDRESS
 9840  035606  062701  000002                  ADD     #2,R1           ;PUT R1 PAST RKCS1
 9841  035612  012021                          MOV     (R0)+,(R1)+     ;LOAD WORD COUNT
 9842  035614  012021                          MOV     (R0)+,(R1)+     ;LOAD BUS ADDRESS
 9843  035616  012021                          MOV     (R0)+,(R1)+     ;LOAD DISK ADDRESS
 9844  035620  012011                          MOV     (R0)+,(R1)      ;LOAD CS2
 9845  035622  062701  000006                  ADD     #6,R1           ;BUMP R1 TO ASOF
 9846  035626  012021                          MOV     (R0)+,(R1)+     ;LOAD OFFSET
 9847  035630  012021                          MOV     (R0)+,(R1)+     ;LOAD CYLINDER
 9848  035632  062701  000004                  ADD     #4,R1           ;BUMP R1 TO MR1
 9849  035636  011011                          MOV     (R0),(R1)       ;LOAD MR1
 9850  035640  013712  001600                  MOV     L.CS1,(R2)      ;LOAD CS1
 9851  035644  012601                          MOV     (SP)+,R1        ;RESTORE REGISTER
 9852  035646  012600                          MOV     (SP)+,R0
 9853  035650  000002                          RTI                     ;RETURN
 9854
 9855                                  .SBTTL  STORE RK611 REGISTERS
 9856                                  ;*      ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
 9857                                  ;*      WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT STORED IN
 9858                                  ;*      THIS ROUTINE.
 9859
 9860  035652  010046                  GETRK:  MOV     R0,-(SP)        ;STORE REGISTERS TO BE USED
 9861  035654  010146                          MOV     R1,-(SP)
 9862  035656  010346                          MOV     R3,-(SP)
 9863  035660  012700  001540                  MOV     #T.CS1,R0       ;SET POINTER TO TEST TABLE
 9864  035664  010201                          MOV     R2,R1           ;SET POINTER TO RK611 BASE
 9865  035666  012703  000012                  MOV     #10.,R3         ;SET COUNT FOR 1ST 10 REGS
 9866  035672  012120                  1$:     MOV     (R1)+,(R0)+     ;STORE RKCS1 THROUGH RKSPAR
 9867  035674  005303                          DEC     R3
 9868  035676  001375                          BNE     1$
 9869  035700  062701  000002                  ADD     #2,R1           ;BUMP POINTER PAST RKDB
 9870  035704  005720                          TST     (R0)+           ;BUMP POINTER PAST T.RKDB
 9871  035706  012703  000004                  MOV     #4,R3           ;SET COUNT FOR LAST 5 REGS
 9872  035712  012120                  2$:     MOV     (R1)+,(R0)+     ;STORE RKMR1 THROUTH RKMR3
 9873  035714  005303                          DEC     R3
 9874  035716  001375                          BNE     2$
 9875  035720  012603                          MOV     (SP)+,R3        ;RESTORE REGISTERS
 9876  035722  012601                          MOV     (SP)+,R1
 9877  035724  012600                          MOV     (SP)+,R0
 9878  035726  000002                          RTI                     ;RETURN
 9879                                  ;SBTTL  BIT COUNTER IN A WORD
 9880                                  ;*      THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE STACK
 9881                                  ;*      BY THE CALLING ROUTINE.  THE NUMBER OF BITS FOUND IN THE WORD
 9882                                  ;*      ARE PASSED BACK ON THE STACK.
 9883
```

```
9884  035730  016637  000002  001256  BITCNT: MOV     2(SP),$TMP16    ;GET WORD WHOSE BITS ARE TO BE COUNTED
9885  035736  010346                          MOV     R3,-(SP)        ;STORE R3
9886  035740  005037  001260                  CLR     $TMP17          ;CLEAR $TMP16 FOR COUNTING
9887  035744  012703  000021                  MOV     #17.,R3         ;SET A SHIFT COUNTER
9888  035750  000241                          CLC                     ;CLEAR CARRY
9889  035752  006037  001256          1$:     ROR     $TMP16          ;ROTATE WORD.
9890  035756  103407                          BCS     3$              ;WAS BIT SHIFTED OUT A 1?
9891  035760  005303                  2$:     DEC     R3              ;NO - DEC COUNT
9892  035762  001373                          BNE     1$              ;LOOP IF NOT ZERO
9893  035764  012603                          MOV     (SP)+,R3        ;RESTORE R3
9894  035766  013766  001260  000002          MOV     $TMP17,2(SP)    ;PUT COUNT OF BITS ON STACK
9895  035774  000204                          RTS     R4              ;RETURN
9896  035776  005237  001260          3$:     INC     $TMP17          ;BUMP COUNT
9897  036002  000766                          BR      2$              ;LOOP
9898
9899                                  .SBTTL  MAINTENANCE CLOCK ROUTINE
9900                                  ;*      THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
9901                                  ;*      ADDRESS OFTER THE CALL. THE FIRST BYTE CANTAINS THE NUMBER
9902                                  ;*      OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
9903                                  ;*      CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
9904                                  ;*      CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
9905
9906  036004                          MCLOCK:
9907  036004  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9908  036006  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9909  036010  112400                          MOVB    (R4)+,R0        ;GET NUMBER OF CONTROLLER PHASE ADDRESSES
9910  036012  112401                          MOVB    (R4)+,R1        ;GET PARTIAL PHASE ADDRESS COUNT
9911
9912  036014  006300                          ASL     R0              ;MULTIPLY PHASE ADDRESS  COUNT BY 4
9913  036016  006300                          ASL     R0
9914  036020  060100                          ADD     R1,R0           ;ADD IN PARTIALS
9915  036022  052762  000400  000026  1$:     BIS     #MCLK,RKMR1(R2) ;SET CLOCK
9916  036030  042762  000400  000026          BIC     #MCLK,RKMR1(R2) ;CLEAR MCLK
9917  036036  005300                          DEC     R0              ;DECREMENT COUNT
9918  036040  001370                          BNE     1$              ;LOOP IF NOT ZERO
9919  036042  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9920  036044  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
9921  036046  000204                          RTS     R4
9922                                  .SBTTL  READ AND SORT HEADERS
9923                                  ;*      THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
9924                                  ;*      THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
9925                                  ;*      ASSCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
9926                                  ;*      READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
9927                                  ;*      EACH READ OF THE DATA BUFFER.
9928                                  ;*
9929                                  ;*      CALL:   JSR     R4,RDSTHD
9930                                  ;*              TCHKOP                  ;RETURN POINT IF CERR IN READ HDR
9931                                  ;*              ERROR   4  ;OR 5, 6, 7
9932                                  ;*              ERROR   13              ;RETURN IF DATA LATE IN DB UNLOAD
9933                                  ;*              ERROR   2               ;RETURN IF TO SLOW OR
9934                                  ;*                                      ;IF HDR 0 NOT FOUND
9935
9936  036050  104413                  RDSTHD: SAVREG
9937  036052  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
9938  036060  001402                          BEQ     20$             ;NO - SKIP
9939  036062  005077  143622                  CLR     @KWLADD ;RESET INTERRUPT
```

```
 9940  036066  012700  000026          20$:    MOV     #26,R0          ;PRESET FOR 26 SECTOR FORMAT
 9941  036072  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS 24 SECTOR MODE SET?
 9942  036100  001402                          BEQ     1$              ;NO - SKIP
 9943  036102  012700  000024                  MOV     #24,R0          ;ELSE CHANGE TO 24 SECTOR MODE
 9944  036106  012701  061414          1$:     MOV     #IBUFF,R1       ;SET POINTER TO INPUT BUFFER
 9945  036112  010005                          MOV     R0,R5           ;SAVE NUMBER OF SECTORS
 9946  036114  010104                          MOV     R1,R4           ;SAVE IBUFF ADDRESS
 9947  036116  010203                          MOV     R2,R3           ;SET UP POINTER TO RKDB
 9948  036120  062703  000024                  ADD     #RKDB,R3
 9949  036124  013762  001626  000010          MOV     DRVNUM,RKCS2(R2) ;LOAD DRIVE NUM
 9950  036132  013762  001614  000020          MOV     L.DCYL,RKDCYL(R2) ;LOAD CYLINDER NUM
 9951  036140  013762  001606  000006          MOV     L.DA,RKDA(R2)   ;LOAD TRACK AND SECTOR
 9952
 9953  036146  012737  000020  001654  2$:     MOV     #16.,TIMCNT     ;SET TIME COUNTER
 9954  036154  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
 9955  036160  005037  001674                  CLR     LCLKTK          ;CLEAR TICK COUNTER
 9956  036164  013762  001600  000000          MOV     L.CS1,RKCS1(R2) ;LOAD COMMAND
 9957
 9958  036172  005737  001662          3$:     TST     INTSET          ;TEST IF INT OCCURRED
 9959  036176  001020                          BNE     4$              ;YES - SKIP
 9960  036200  004737  035150                  JSR     PC,MYTIME       ;WAIT 1 MS
 9961  036204  005737  001674                  TST     LCLKTK          ;HAVE WE WAITED 16 MS?
 9962  036210  001770                          BEQ     3$              ;NO - LOOP ON WAIT
 9963
 9964  036212  062766  000006  000006          ADD     #6,6(SP)        ;SET RETURN FOR TO SLOW
 9965  036220  104420                          TGETRK                  ;GET RK REGS
 9966  036222  012737  053514  001370          MOV     #EM4,EM2N       ;LOAD MESSAGE "TO SLOW/NOT COMPLETE"
 9967  036230  012737  050004  001372          MOV     #OPER24,DH2N    ;LOAD COMMAND "READ HEADER" FOR REPORT
 9968  036236  000466                          BR      10$             ;SKIP
 9969
 9970  036240  005762  000000          4$:     TST     RKCS1(R2)       ;TEST FOR CONTROLLER ERROR
 9971  036244  100474                          BMI     11$             ;YES - SKIP
 9972
 9973  036246  011324                          MOV     (R3),(R4)+      ;STORE HEADERS
 9974  036250  011324                          MOV     (R3),(R4)+
 9975  036252  011324                          MOV     (R3),(R4)+
 9976
 9977  036254  005762  000010                  TST     RKCS2(R2)       ;TEST IF DATA LATE
 9978  036260  100443                          BMI     8$              ;YES - SKIP
 9979
 9980  036262  005300                          DEC     R0              ;DEC SECTOR COUNT
 9981  036264  001330                          BNE     2$              ;IF NOT ZERO - LOOP
 9982
 9983  036266  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
 9984  036274  001403                          BEQ     5$              ;NO - SKIP
 9985  036276  012777  000100  143404          MOV     #BIT6,@KWLADD   ;SET INTERRUPT ENABLE
 9986  036304  032761  000037  000002  5$:     BIT     #37,2(R1)       ;HEADER AT TOP OF BUFF=HEAD 0?
 9987  036312  001413                          BEQ     6$              ;YES - SKIP
 9988  036314  012124                          MOV     (R1)+,(R4)+     ;ELSE MOV THIS HEADER TO BOTTOM
 9989  036316  012124                          MOV     (R1)+,(R4)+
 9990  036320  012124                          MOV     (R1)+,(R4)+
 9991
 9992  036322  005305                          DEC     R5              ;TEST FO INSURE HEAD 0 IS FOUND
 9993  036324  001367                          BNE     5$              ;IF ALL HEADERS NOT CHECKED - LOOP
 9994  036326  012737  056264  001370          MOV     #EM56,EM2N      ;ELSE "HEADER 0 NOT FOUND" MESSAGE
 9995  036334  005037  001372                  CLR     DH2N
```

```
 9996  036340  000421                          BR      9$              ;SKIP
 9997
 9998  036342  012700  061414          6$:     MOV     #IBUFF,R0       ;GET TOP OF IBUFF
 9999  036346  012120                  7$:     MOV     (R1)+,(R0)+     ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
10000  036350  012120                          MOV     (R1)+,(R0)+
10001  036352  012120                          MOV     (R1)+,(R0)+
10002  036354  020004                          CMP     R0,R4           ;ALL HEADERS MOVED?
10003  036356  001373                          BNE     7$              ;NO - LOOP
10004
10005  036360  062766  000010  000006          ADD     #10,6(SP)       ;SET UP FOR GOOD RETURN
10006  036366  000423                          BR      11$
10007
10008  036370  012737  054527  001500  8$:     MOV     #EMDLT,EM13N    ;"DATA LATE SET RESULT OF READ DB"
10009  036376  012737  056675  061010          MOV     #EMRDB,DF011A
10010  036404  062766  000004  000006  9$:     ADD     #4,6(SP)        ;SET ERROR RETURN
10011  036412  104420                          TGETRK                  ;GET RK REGS
10012  036414  013700  001302          10$:    MOV     $TESTN,R0       ;GET TEST NUMBER
10013  036420  006300                          ASL     R0              ;SHIFT FOR INDEX
10014  036422  016037  033032  001264          MOV     $SWO8TB(R0),$ESCAPE  ;SET ESCAPE
10015  036430  162737  000002  001264          SUB     #2,$ESCAPE      ;TO NEXT SCOPE CALL
10016
10017  036436  104414                  11$:    RESREG
10018  036440  000204                          RTS     R4
10019
10020                                  .SBTTL  GET DRIVE STATUS
10021                                  ;*      THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
10022                                  ;*      THROUGH $REG17.  THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
10023                                  ;*      THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
10024                                  ;*      IN THE REGISTERS.
10025                                  ;*
10026                                  ;*CALL: JSR     R4,GETDRS
10027                                  ;*      BR      ERROR PROCESSING        ERROR RETURN
10028                                  ;*      BR      NO ERROR PROCESSING     GOOD RETURN
10029
10030  036442  104413                  GETDRS: SAVREG
10031  036444  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
10032  036452  012700  001202                  MOV     #$REG10,R0      ;PRESET ALL STATUS STORAGE TO
10033  036456  012701  000010                  MOV     #8.,R1          ;ALL ONES
10034  036462  012720  177777          1$:     MOV     #177777,(R0)+
10035  036466  005301                          DEC     R1
10036  036470  001374                          BNE     1$
10037  036472  012700  001206                  MOV     #$REG12,R0      ;SET POINTER TOREG12 FOR A01 & B01
10038  036476  012701  000001                  MOV     #1,R1           ;PRESET FOR PAIR ONE.
10039  036502  005037  001230                  CLR     $TMP3           ;CLEAR ERROR SWITCH
10040  036506  013762  001610  000010  2$:     MOV     L.CS2,RKCS2(R2) ;LOAD DRIVE #
10041  036514  010162  000026                  MOV     R1,RKMR1(R2)    ;LOAD MR1
10042  036520  012762  000001  000000          MOV     #BIT0,RKCS1(R2) ;DO SELECT
10043  036526  012703  000050                  MOV     #40.,R3         ;WAIT FOR A FEW MICRO RECORDS TO
10044  036532  005303                  3$:     DEC     R3              ;BIT SELECT FINISH
10045  036534  001376                          BNE     3$
10046  036536  032762  100000  000000          BIT     #CERR,RKCS1(R2) ;ANY ERROR SET AS A RESULT OF SELECT?
10047  036544  001415                          BEQ     4$              ;NO - SKIP
10048  036546  032762  024000  000000          BIT     #CTO!SPAR,RKCS1(R2) ;CHECK IF TIMEOUT OR PARITY ERROR
10049  036554  001004                          BNE     8$              ;YES - SKIP
10050  036556  032762  037400  000010          BIT     #37400,RKCS2(R2) ;TEST FOR ERRORS:
10051                                                                  ;   NED!UPE!MDS!UFE!NEM!PGE
```

```
10052  036564  001405                          BEQ     4$              ;NO - SKIP
10053  036566  012737  000001  001230  8$:     MOV     #1,$TMP3        ;SET ERROR FLAG
10054  036574  022020                          CMP     (R0)+,(R0)+     ;BUMP TO LET THAT PAIR STAY ALL 1'S.
10055  036576  000404                          BR      5$              ;SKIP
10056  036600  016220  000034          4$:     MOV     RKMR2(R2),(R0)+ ;STORE A WORD
10057  036604  016220  000036                  MOV     RKMR3(R2),(R0)+ ;STORE B WORD
10058  036610  012762  100000  000000  5$:     MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERROR IN CONTROLLER
10059  036616  005701                          TST     R1              ;IS R1 A 0 (LAST TRANSFER, PAIR 0)
10060  036620  001410                          BEQ     6$              ;YES - SKIP
10061  036622  005201                          INC     R1              ;ELSE BUMP TO NEXT PAIR
10062  036624  022701  000004                  CMP     #4,R1           ;PAIR 3  JUST STORED?
10063  036630  001326                          BNE     2$              ;NO - SKIP
10064  036632  005001                          CLR     R1              ;ELSE SET TO PAIR 0
10065  036634  012700  001202                  MOV     #$REG10,R0      ;PRESET POINTER FOR PAIR 0
10066  036640  000722                          BR      2$              ;GO GET THEM
10067  036642  104414                  6$:     RESREG                  ;EXIT HERE
10068  036644  005737  001230                  TST     $TMP3           ;ANY ERROR IN STATUS GETTING
10069  036650  001001                          BNE     7$              ;YES - SKIP
10070  036652  005724                          TST     (R4)+           ;ELSE BUMP PART ERROR
10071  036654  000204                  7$:     RTS     R4              ;RETURN
10072
10073                                  .SBTTL  SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
10074                                  ;*      THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
10075                                  ;*      COMMAND.  CERR AND DI ARE MONITORED FOR A SHORT
10076                                  ;*      PEIROD OF TIME DURING WHICH THEY SHOULD BOTH RESET.
10077                                  ;*
10078                                  ;*      IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
10079                                  ;*
10080                                  ;*      IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE
10081                                  ;*      ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE
10082                                  ;*      RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN
10083                                  ;*      SKIPS OVER THE ERROR CALL AND TEST ABORT.
10084                                  ;*
10085                                  ;*      THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY
10086                                  ;*      AN ERROR MESSAGE AND BRANCH TO END OF TEST.  THIS
10087                                  ;*      IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
10088                                  ;*      THE TEST.
10089
10090  036656                          SSINIT:
10091  036656  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10092  036660  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10093  036662  012701  000007                  MOV     #7,R1           ;SET CLEAR COUNT
10094  036666  012700  001600                  MOV     #L.CS1,R0       ;GET ADDRESS OF "L" REGS
10095  036672  012720  000100                  MOV     #100,(R0)+      ;PRESET CS1
10096  036676  005020                  7$:     CLR     (R0)+           ;CLEAR THE NEXT
10097  036700  005301                          DEC     R1              ;COUNT 0?
10098  036702  001375                          BNE     7$              ;NO - LOOP
10099  036704  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
10100  036712  012737  000012  001222          MOV     #10.,$TMP0      ;SET A COUNTER
10101  036720  016237  000000  001540  1$:     MOV     RKCS1(R2),T.CS1 ;GET CS1
10102  036726  032737  140000  001540          BIT     #CERR!DI,T.CS1  ;TEST IF ERROR OR DI SET
10103  036734  001433                          BEQ     2$              ;NO - SKIP TO READY TEST
10104  036736  005337  001222                  DEC     $TMP0           ;ELSE DECREMENT COUNTER
10105  036742  001366                          BNE     1$              ;AND LOOP
10106  036744  032737  100000  001540          BIT     #CERR,T.CS1     ;TEST - IS IT CERR STILL SET
10107  036752  001404                          BEQ     3$              ;NO  - SKIP TO DI MESSAGE
```

```
10108  036754  012737  053607  001400          MOV     #EM5,EM3N       ;MESSAGE (SUBSYS CLR NOT RESET ERROR)
10109  036762  000403                          BR      4$
10110  036764  012737  053653  001400  3$:     MOV     #EM6,EM3N       ;MESSAGE (SUBSYS CLEAR NOT RESET DI)
10111  036772  104420                  4$:     TGETRK
10112  036774  013700  001302                  MOV     $TESTN,R0       ;GET PRESENT TEST NUMBER
10113  037000  006300                          ASL     R0              ;SHIFT FOR INDEX
10114  037002  016037  033032  001264          MOV     $SWO8TBL(R0),$ESCAPE ;LOAD ESCAPE TO ABORT TEST
10115  037010  162737  000002  001264          SUB     #2.$ESCAPE      ;SET TO NEXT SCOPE CALL
10116  037016  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10117  037020  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10118  037022  000414                          BR      6$              ;OSKIP TO EXIT
10119  037024  032737  000200  001540  2$:     BIT     #RDY,T.CS1      ;TEST READY SET
10120  037032  001004                          BNE     5$              ;YES - GOOD EXIT
10121  037034  012737  053732  001400          MOV     #EM7,EM3N       ;MESSAGE (SUBSYS CLR NOT SET READY)
10122  037042  000753                          BR      4$
10123  037044  012601                  5$:     MOV     (SP)+,R1        ;RESTORE REGS
10124  037046  012600                          MOV     (SP)+,R0
10125  037050  062716  000002                  ADD     #2,(SP)         ;GOOD RETURN
10126  037054  000002                  6$:     RTI
10127
10128                                   .SBTTL  WORD COUNT AT END OF OPERATION CHECK
10129                                   ;*      THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR
10130                                   ;*      THE WORD COUNT AGAINST THE SUPPLIED VALUE.  IF UNEQUAL, THE
10131                                   ;*      ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)
10132                                   ;*
10133                                   ;*CALL: JSR     R4,CHKWC
10134                                   ;*      .WORD                   ;EXPECTED WC VALUE
10135
10136  037056  012437  047424          CHKWC:  MOV     (R4)+,EXPWC     ;STORE EXPECTED VALUE
10137                                                                  ;
10138  037062  023737  047424  001542          CMP     EXPWC,T.WC      ;COMPARE
10139  037070  001406                          BEQ     1$              ;EQUAL - SKIP
10140  037072  052737  000001  047462          BIS     #WCERR,GRP4ER   ;SET ERROR FLAG
10141  037100  013737  001542  047440          MOV     T.WC,REALWC     ;STORE REAL WORD COUNT
10142  037106  000204                  1$:     RTS     R4              ;RETURN.
10143
10144                                   .SBTTL  BUS  ADDRESS AT END OF OPERATION CHECK
10145                                   ;*      THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF
10146                                   ;*      A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE
10147                                   ;*      INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.
10148                                   ;*      IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA
10149                                   ;*      AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)
10150                                   ;*
10151                                   ;*      IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS
10152                                   ;*      ADDRESS IS THE STARTING BUS ADDRESS.
10153                                   ;*CALL: JSR     R4,CHKBA
10154
10155  037110  010046                  CHKBA:  MOV     R0,-(SP)
10156  037112  010146                          MOV     R1,-(SP)
10157  037114  010346                          MOV     R3,-(SP)
10158  037116  032737  000020  001610          BIT     #BAI,L.CS2      ;TEST IF BAI SET
10159  037124  001404                          BEQ     4$              ;NO - SKIP
10160  037126  013737  001604  047430          MOV     L.BA,EXPBA      ;STORE EXPECTED BA (SAME AS STARTING BA)
10161  037134  000441                          BR      3$
10162  037136  013700  001602          4$:     MOV     L.WC,R0         ;GET INITIAL WORD COUNT
10163  037142  005400                          NEG     R0
```

```
10164  037144  113703  001601              MOVB    L.CS1+1,R3      ;GET BA16 & BA17
10165  037150  042703  177774              BIC     #177774,R3      ;CLEAR UNWANTED BITS
10166
10167  037154  005700                      TST     R0              ;TEST IF INITIAL WORD COUNT 0
10168  037156  001003                      BNE     6$              ;NO - SKIP
10169  037160  062703  000002              ADD     #2,R3           ;ADD 2 TO BA16,17 (65K WORD XFER)
10170  037164  000407                      BR      9$              ;SKIP
10171  037166  005700          6$:         TST     R0              ;TEST IF INITIAL WC BIT 15 SET
10172  037170  100001                      BPL     5$              ;NO - SKIP
10173  037172  005203                      INC     R3              ;BUMP BA16,17 (32K WORD XFER)
10174  037174  006300          5$:         ASL     R0              ;SHIFT WORD COUNT TO MAKE MEM ADD CNT
10175  037176  063700  001604              ADD     L.BA,R0         ;ADD IN START BUFFER ADD
10176  037202  005503                      ADC     R3              ;IF CARRY - ADD INTO BA16,17
10177  037204  013701  001542      9$:     MOV     T.WC,R1         ;GET END OF OPERATION WORD COUNT
10178  037210  001411                      BEQ     1$              ;BRANCH IF ZERO
10179  037212  005401                      NEG     R1
10180  037214  005701                      TST     R1              ;TEST END OPERATION WC BIT 15 SET
10181  037216  100001                      BPL     7$              ;NO - SKIP
10182  037220  005303                      DEC     R3              ;DEC BA 16,17 (32K WC LEFT)
10183  037222  006301          7$:         ASL     R1              ;SHIFT WC TO MAKE MEM ADD CNT
10184  037224  160100                      SUB     R1,R0           ;SUB FROM COMPUTED BUS ADDRESS
10185  037226  005603                      SBC     R3              ;SUB CARRY FROM BA16,17
10186  037230  010337  047426              MOV     R3,EXPUBA       ;STORE EXPECTED UPPER BA BITS
10187  037234  010037  047430      1$:     MOV     R0,EXPBA
10188  037240  020037  001544      3$:     CMP     R0,T.BA         ;EQUAL TO COMPUTED?
10189  037244  001406                      BEQ     2$              ;YES - SKIP
10190  037246  052737  000004  047462      BIS     #BAERR,GRP4ER   ;ELSE SET BAERR FLAG
10191  037254  013737  001544  047444      MOV     T.BA,REALBA     ;STORE REAL BUS ADDRESS
10192  037262  113703  001541      2$:     MOVB    T.CS1+1,R3      ;GET REAL UPPER BA
10193  037266  042703  177774              BIC     #177774,R3      ;CLEAR UNWANTED BITS
10194  037272  020337  047426              CMP     R3,EXPUBA       ;CHECK IF EQUAL
10195  037276  001405                      BEQ     8$              ;YES - SKIP
10196  037300  052737  000002  047462      BIS     #UBAERR,GRP4ER  ;ELSE SET UBA ERROR
10197  037306  010337  047442              MOV     R3,REALUB       ;STORE REAL UPPER BA
10198  037312  012603          8$:         MOV     (SP)+,R3
10199  037314  012601                      MOV     (SP)+,R1
10200  037316  012600                      MOV     (SP)+,R0
10201  037320  000204                      RTS     R4
10202
10203                          .SBTTL  CYLINDER,TRACK,SECTOR TEST AT END OF OPERATION
10204                          ;*      THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA
10205                          ;*      ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE
10206                          ;*      OPERATION.  THE CONTENTS OF THE LOAD REGISTER STORAGE ARE
10207                          ;*      COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE
10208                          ;*      NECESSARY CALCULATION.
10209                          ;*
10210                          ;*      ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES
10211                          ;*      STORAGE EXPCYL, EXPTRK, EXPSEC.  ALL 3 ARE CHECKED AND
10212                          ;*      IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE
10213                          ;*      ERROR FLAGS FIELD (GRP4ER) IS SET.
10214                          ;*
10215                          ;*CALL: JSR     R4,CHKCTS
10216
10217  037322  104413          CHKCTS: SAVREG
10218  037324  013700  001602              MOV     L.WC,R0         ;GET SPECIFIED WORD COUNT
10219  037330  005400                      NEG     R0              ;NEGATE IT
```

```
10220  037332  013701  001542          MOV     T.WC,R1         ;GET END OF OPERATION WORD COUNT
10221  037336  001401                  BEQ     10$             ;IF ZERO - SKIP
10222  037340  005401                  NEG     R1              ;NEGATE IT
10223  037342  160100          10$:    SUB     R1,R0           ;COMPUTE ACTUAL WORDS TRANSFERRED
10224  037344  005001                  CLR     R1              ;CLEAR R1 FOR COUNTING
10225                          ;       THE FOLLOWING CODE DETERMINES HOW MAY SECTORS OF DATA HAS BEEN
10226                          ;       TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE
10227                          ;       END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR
10228                          ;       IS CALCULATED.
10229  037346  022700  000400    1$:     CMP     #400,R0
10230  037352  003004                  BGT     2$
10231  037354  005201                  INC     R1
10232  037356  162700  000400          SUB     #400,R0
10233  037362  000771                  BR      1$
10234  037364  005700          2$:     TST     R0
10235  037366  001401                  BEQ     3$
10236  037370  005201                  INC     R1
10237                          ;       AT THIS POINT R1 HAS A COUNT OF THE
10238                          ;       NUMBER OF FULL SECTOR TRANSFER + 1 IF A
10239                          ;       PARTIAL SECTOR WAS TRANSFERRED.
10240  037372  012703  000026    3$:     MOV     #26,R3
10241  037376  032737  010000  001600  BIT     #CFMT,L.CS1
10242  037404  001402                  BEQ     4$
10243  037406  012703  000024          MOV     #24,R3
10244                          ;       R3 HAS BEEN SET UP WTH THE NUMBER
10245                          ;       OF SECTORS IN A TRACK FOR THE FORMAT USED
10246  037412  013737  001614  047432  4$:     MOV     L.DCYL,EXPCYL   ;GET STARTING VALUES FOR CYLINDER
10247  037420  113704  001607          MOVB    L.DT,R4         ;TRACK
10248  037424  042704  177400          BIC     #177400,R4
10249  037430  113705  001606          MOVB    L.DS,R5         ;SECTOR
10250  037434  042705  177400          BIC     #177400,R5
10251  037440  005301                  DEC     R1              ;ADJUST COUNT FOR ZERO DETECT
10252  037442  005205          5$:     INC     R5              ;BUMP SECTOR COUNT
10253  037444  020503                  CMP     R5,R3           ;DID THIS MAKE SECTOR COUTN > 1 TRACK?
10254  037446  001010                  BNE     6$              ;NO - SKIP
10255  037450  005005                  CLR     R5              ;ELSE CLEAR SECTOR COUNT
10256  037452  005204                  INC     R4              ;BUMP TRACK COUNT
10257  037454  022704  000003          CMP     #3,R4           ;DID THIS MAKE TRK COUNT > 1 CYLINDER?
10258  037460  001003                  BNE     6$              ;NO - SKIP
10259  037462  005004                  CLR     R4              ;ELSE CLEAR TRACK COUNT
10260  037464  005237  047432          INC     EXPCYL          ;BUMP CYLINDER COUNT
10261  037470  005301          6$:     DEC     R1              ;DEC COUNT
10262  037472  001363                  BNE     5$              ;IF ZERO - EXIT
10263  037474  010437  047436          MOV     R4,EXPTRK       ;STORE EXPECTED TRACK
10264  037500  010537  047434          MOV     R5,EXPSEC       ;STORE EXPECTED SECTOR (CYL ALREADY SLOW)
10265  037504  023737  001560  047432  CMP     T.DCYL,EXPCYL   ;TEST IF CYLINDER OK
10266  037512  001403                  BEQ     7$              ;YES - SKIP
10267  037514  052737  000010  047462  BIS     #CYLERR,GRP4ER  ;NO - SET ERROR FLAG
10268  037522  120437  001547    7$:     CMPB    R4,T.DA+1       ;TEST TRACK OK
10269  037526  001403                  BEQ     8$              ;YES - SKIP
10270  037530  052737  000020  047462  BIS     #TRKERR,GRP4ER  ;NO - SET ERROR FLAG
10271  037536  120537  001546    8$:     CMPB    R5,T.DA         ;TEST SECTOR COUNT OK
10272  037542  001403                  BEQ     9$              ;YES - SKIP
10273  037544  052737  000040  047462  BIS     #SECERR,GRP4ER  ;USE SET ERROR FLAG
10274  037552  012700  047440    9$:     MOV     #REALWC,R0
10275  037556  013720  001542          MOV     T.WC,(R0)+      ;STORE REAL WORD COUNT
```

```
10276  037562  013720  001544                  MOV     T.BA,(R0)+      ;STORE REAL BUS ADDRESS
10277  037566  013720  001560                  MOV     T.DCYL,(R0)+    ;STORE REAL CYLINDER ADDRESS
10278  037572  113710  001547                  MOVB    T.DA+1,(R0)     ;STORE REAL TRACK ADDRESS
10279  037576  005720                          TST     (R0)+
10280  037600  113710  001546                  MOVB    T.DA,(R0)       ;STORE REAL SECTOR ADDRESS
10281  037604  104414                          RESREG
10282  037606  000204                          RTS     R4
10283
10284                                  .SBTTL  OPERATION CHECK ROUTINE
10285                                  ;*      THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME  SOFTWARE
10286                                  ;*      ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS:
10287                                  ;*      THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION
10288                                  ;*      FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH
10289                                  ;*      THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN
10290                                  ;*      ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED
10291                                  ;*      BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL
10292                                  ;*      BE THE PROPER ITEM FOR  THE FORMAT REQUIRED BY THIS
10293                                  ;*      ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4,5,6,
10294                                  ;*      7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION
10295                                  ;*      OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT
10296                                  ;*      FORMAT.
10297
10298  037610  104413                  CHKWE:  SAVREG
10299  037612  011600                          MOV     (SP),R0         ;GET POINTER TO ERROR WORDS
10300  037614  012037  001242                  MOV     (R0)+,$TMP10    ;STORE EXPECTED ERROR GROUP 1
10301  037620  012037  001244                  MOV     (R0)+,$TMP11    ;STORE EXPECTED ERROR GROUP 2
10302  037624  012037  001246                  MOV     (R0)+,$TMP12    ;STORE EXPECTED ERROR GROUP 3
10303  037630  010016                          MOV     R0,(SP)         ;STORE RETURN
10304  037632  012737  177777  001250          MOV     #-1,$TMP13      ;SET FLAG - EXPECTED ERROR
10305  037640  000403                          BR      CHKST
10306
10307  037642  104413                  CHKOP:  SAVREG
10308  037644  005037  001250                  CLR     $TMP13          ;RESET EXPECTED ERROR FLAG
10309
10310  037650  104420                  CHKST:  TGETRK                  ;GET 611 REGS IO TRAP
10311  037652  005037  047454                  CLR     GRP1ER          ;CLEAR ERROR FLAGS
10312  037656  005037  047456                  CLR     GRP2ER
10313  037662  005037  047460                  CLR     GRP3ER
10314  037666  005037  047462                  CLR     GRP4ER
10315  037672  005037  047576                  CLR     GPSUMF          ;CLEAR SUMMARY FLAGS
10316  037676  032737  024000  001540          BIT     #CS1ERBIT,T.CS1 ;TEST IF ERROR SET IN CS1
10317  037704  001111                          BNE     4$              ;YES - SKIP
10318  037706  032737  177400  001550          BIT     #CS2ERBIT,T.CS2 ;TEST IF ERROR SET IN CS2
10319  037714  001105                          BNE     4$              ;YES - SKIP
10320  037716  032737  000070  001552          BIT     #DSERBIT,T.DS   ;TEST IF ERROR SET IN DS
10321  037724  001101                          BNE     4$              ;YES - SKIP
10322  037726  005737  001554                  TST     T.ER            ;TEST IF ERROR SET IN ER
10323  037732  001076                          BNE     4$              ;YES - SKIP
10324  037734  032737  100000  001540          BIT     #CERR,T.CS1     ;COMBINED ERROR SET?
10325  037742  001405                          BEQ     9$              ;NO - SKIP
10326  037744  052737  100000  047454          BIS     #CERNER,GRP1ER  ;SET ERROR FLAG IN GROUP 1
10327  037752  000137  040502                  JMP     25$             ;SKIP
10328
10329                                  ;       CODE TO CHECK WORD COUNT, BUFFER ADDRESS, CYLINDER, TRACK,
10330                                  ;       AND SECTOR AT THE END OF THE OPERATION. THIS IS DONE ONLY
10331                                  ;       IF CERR NOT SET BY THE OPERATION.
```

```
10332                                   ;
10333                                   ;       ALL OF THE ABOVE CONDITIONS ARE CHECKED AND A BIT SET FOR
10334                                   ;       EACH CHECK THAT FAILS. HOWEVER, ONLY ONE ERROR IS REPORTED.
10335                                   ;       THE ORDER OF PRIORITY FOR REPORTING THE ERROR IS THE ORDER
10336                                   ;       LISTED ABOVE.
10337
10338  037756  005737  001250         9$:     TST     $TMP13          ;TEST IF ERROR EXPECTED
10339  037762  001402                         BEQ     62$             ;NO - SKIP
10340  037764  000137  040502                 JMP     25$             ;YES - JUMP
10341  037770  013700  001540         62$:    MOV     T.CS1,R0        ;GET CS1
10342  037774  042700  177741                 BIC     #177741,R0      ;CHECK IF OPERATION IS READ DATA,
10343  040000  022700  000020                 CMP     #20,R0          ;WRITE DATA, OR WRITE CHECK, IF
10344  040004  002445                         BLT     3$              ;NOT, SKIP ALL CHECKING IN GROUP
10345  040006  022700  000030                 CMP     #30,R0          ;FOUR
10346  040012  003042                         BGT     3$
10347  040014  004437  037056                 JSR     R4,CHKWC        ;CHECK WORD COUNT
10348  040020  000000                         .WORD   0               ;EXPECTED WORD COUNT
10349  040022  004437  037110                 JSR     R4,CHKBA        ;CHECK BUS ADDRESSS
10350  040026  004437  037322                 JSR     R4,CHKCTS       ;CHECK CYL, TRACK, & SECTOR
10351  040032  005737  047462                 TST     GRP4ER          ;ANY GROUP 4 ERRORS?
10352  040036  001430                         BEQ     3$              ;NO - SKIP
10353  040040  016037  047600  060770         MOV     CMNDLB(R0),DF010A ;LOAD ADDRESS OF COMMAND MESSAGE
10354  040046  013700  047462                 MOV     GRP4ER,R0       ;PUT GROUP 4 ERROR FLAG IN R0
10355  040052  005001                         CLR     R1              ;CLEAR R1 FOR INDEX COUNTER
10356  040054  006200                 1$:     ASR     R0              ;SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
10357  040056  103402                         BCS     2$              ;BE REPORTED, REST ARE IGNORED
10358  040060  005720                         TST     (R0)+           ;WHEN AN ERROR BIT IS FOUND,
10359  040062  000774                         BR      1$              ;GET THE ERROR MESSAGE ASSOCIATED
10360  040064  016037  047464  001450 2$:     MOV     GRP4MS(R0),EMION ;WITH IT AND SET ERROR TABLE ITEM TO
10361  040072  016037  047424  001202         MOV     EXPWC(R0),$REG10 ;POINT TO THE MESSAGE. LOAD REG10 & 11
10362  040100  016037  047440  001204         MOV     REALWC(R0),$REG11 ;WITH EXPECTED & IS VALUES
10363  040106  104414                         RESREG                  ;RESTORE REGISTER
10364  040110  012776  000010  000000         MOV     #10,2(SP)       ;MAKE THE ERROR CALL POINT TO THE
10365  040116  000002                         RTI                     ;RIGHT TABLE ENTRY & RETURN.
10366  040120                         3$:
10367  040120  104414                         RESREG
10368  040122  062716  000002                 ADD     #2,(SP)         ;BUMP RETURN PAST ERROR
10369  040126  000002                         RTI                     ;RETURN
10370
10371                                   ;       THE FOLLOWING CODE BUILDS THE GROUP 1,2, & 3 ERROR WORDS.
10372
10373  040130  012700  047454         4$:     MOV     #GRP1ER,R0      ;SET UP GENERAL REGISTER AS POINTER
10374  040134  012701  001540                 MOV     #T.CS1,R1       ;CS1
10375  040140  012703  001550                 MOV     #T.CS2,R3       ;CS2
10376  040144  012704  001552                 MOV     #T.DS,R4        ;DS
10377  040150  012705  001554                 MOV     #T.ER,R5        ;AND ER
10378
10379  040154  051510                         BIS     (R5),(R0)       ;SET ALL BITS IN GRP1ER THAT
10380                                                                  ;CORRESPOND TO ERROR BITS IN RKER
10381  040156  042710  120701                 BIC     #ILF!ECH!BSE!HVRC!OPI!DCK,(R0) ; CLEAR ALL THAT DON'T BELONG GRP1
10382
10383  040162  032711  020000                 BIT     #SPAR,(R1)      ;TEST IF SPAR SET
10384  040166  001402                         BEQ     5$              ;NO - SKIP
10385  040170  052710  000001                 BIS     #SPARERR,(R0)   ;SET SPAR ERROR FLAG
10386
10387  040174  032714  000010         5$:     BIT     #ACLO,(R4)      ;TEST ACLO SET
```

```
10388  040200  001402                      BEQ     6$              ;NO - SKIP
10389  040202  052710  000100              BIS     #ACLOERR,(R0)   ;SET ACLO ERROR FLAG
10390
10391  040206  032714  000020      6$:     BIT     #SPDLSS,(R4)    ;TEST SPEED LOSS SET
10392  040212  001402                      BEQ     7$              ;NO - SKIP
10393  040214  052710  000200              BIS     #SPDERR,(R0)    ;SET SPEED LOSS ERROR FLAG
10394
10395  040220  032714  000040      7$:     BIT     #DROT,(R4)      ;TEST IF DROT SET
10396  040224  001402                      BEQ     8$              ;NO - SKIP
10397  040226  052710  000400              BIS     #DROTERR,(R0)   ;SET DROT ERROR FLAG
10398
10399  040232  032711  100000      8$:     BIT     #CERR,(R1)      ;TEST CERR ITSELF SET
10400  040236  001002                      BNE     10$             ;YES - SKIP
10401  040240  032710  020000              BIT     #NCERWE,(R0)    ;SET NO CERR WITH ERROR FLAG
10402
10403  040244  012700  047456      10$:    MOV     #GRP2ER,R0      ;SET POINTER TO GROUP 2 ERROR FLAGS
10404
10405  040250  032715  000100              BIT     #ECH,(R5)       ;TEST IF ECH SET
10406  040254  001402                      BEQ     11$             ;NO - SKIP
10407  040256  052710  000001              BIS     #ECHERR,(R0)    ;SET ECH FLAG
10408
10409  040262  032715  100000      11$:    BIT     #DCK,(R5)       ;TEST DCK SET
10410  040266  001402                      BEQ     12$             ;NO - SKIP
10411  040270  052710  000002              BIS     #DCKERR,(R0)    ;SET DCK ERROR FLAG.
10412
10413  040274  032713  040000      12$:    BIT     #WCE,(R3)       ;TEST WRITE CHECK ERROR
10414  040300  001402                      BEQ     120$            ;NO - SKIP
10415  040302  052710  000004              BIS     #WCKERR,(R0)    ;SET WCE BIT
10416  040306  032713  100000      120$:   BIT     #DLT,(R3)       ;TEST DATA LATE
10417  040312  001402                      BEQ     13$             ;NO - SKIP
10418  040314  052710  000010              BIS     #DLTERR,(R0)    ;SET DLT ERROR FLAG
10419
10420  040320  032715  020000      13$:    BIT     #OPI,(R5)       ;TEST OPI SET
10421  040324  001402                      BEQ     14$             ;NO - SKIP
10422  040326  052710  000020              BIS     #OPIERR,(R0)    ;SET OPI ERROR FLAG
10423
10424  040332  032715  000400      14$:    BIT     #HVRC,(R5)      ;TEST HVRC SET
10425  040336  001402                      BEQ     16$             ;NO - SKIP
10426  040340  052710  000040              BIS     #HVRCERR,(R0)   ;SET HVRC FLAG
10427
10428  040344  032715  000200      16$:    BIT     #BSE,(R5)       ;TEST BSE ERROR FLAG
10429  040350  001402                      BEQ     17$             ;NO - SKIP
10430  040352  052710  000100              BIS     #BSERR,(R0)     ;SET BSE FLAG
10431
10432  040356  012700  047460      17$:    MOV     #GRP3ER,R0      ;SET POINTER TO GROUP 3 FLAGS
10433
10434  040362  032713  010000              BIT     #NED,(R3)       ;TEST NED SET
10435  040366  001402                      BEQ     18$             ;NO - SKIP
10436  040370  052710  000001              BIS     #NEDERR,(R0)    ;SET NED FLAG
10437
10438  040374  032711  004000      18$:    BIT     #CTO,(R1)       ;TEST CTO SET
10439  040400  001402                      BEQ     19$             ;NO - SKIP
10440  040402  052710  000002              BIS     #CTOERR,(R0)    ;SET CTO FLAG
10441
10442  040406  032713  000400      19$:    BIT     #UFE,(R3)       ;TEST UFE SET
10443  040412  001402                      BEQ     20$             ;NO - SKIP
```

```
10444  040414  052710  000004          BIS     #UFERR,(R0)     ;SET UFE FLAG
10445
10446  040420  032713  001000  20$:    BIT     #MDS,(R3)       ;TEST MDS SET
10447  040424  001402                  BEQ     21$             ;NO - SKIP
10448  040426  052710  000010          BIS     #MDSERR,(R0)    ;SET MDE FLAG
10449
10450  040432  032713  002000  21$:    BIT     #PGE,(R3)       ;TEST PGE SET
10451  040436  001402                  BEQ     22$             ;NO - SKIP
10452  040440  052710  000020          BIS     #PGERR,(R0)     ;SET PGE FLAG
10453
10454  040444  032713  004000  22$:    BIT     #NEM,(R3)       ;TEST NEM SET
10455  040450  001402                  BEQ     23$             ;NO - SKIP
10456  040452  052710  000040          BIS     #NEMERR,(R0)    ;SET NEM FLAG
10457
10458  040456  032713  020000  23$:    BIT     #UPE,(R3)       ;TEST UPE SET
10459  040462  001402                  BEQ     24$             ;NO - SKIP
10460  040464  052710  000100          BIS     #UPERR,(R0)     ;SET UPE FLAG
10461
10462  040470  032715  000001  24$:    BIT     #ILF,(R5)       ;TEST ILF SET
10463  040474  001402                  BEQ     25$             ;NO - SKIP
10464  040476  052710  000200          BIS     #ILFERR,(R0)    ;SET ILF FLAG.
10465
10466                          ;       THE FOLLOWING CODE IS EXECUTED ONLY IF ERRORS WERE EXPECTED.
10467                          ;       THE FLAG IN $TMP13 INDICATES IF
10468                          ;       AN ERROR WAS EXPECTED AND THE CONTENTS OF TMP10,
10469                          ;       TEMP11, & TEMP12 SPECIFY WHICH ERRORS. THESE ARE COMPARED AGAINST
10470                          ;       THE ERRORS FOUND AND STORED IN GRP1ER, GRP2ER, AND GRP3ER.
10471                          ;       THE CONTENTS OF GRP1,2, & 3 ARE MODIFIED TO INDICATE ERRORS THAT
10472                          ;       OCCURRED BUT WERE NOT EXPECTED. THE CONTENTS OF $TMP10,11,
10473                          ;       & 12 ARE MODIFIED TO INDICATE EXPECTED ERRORS THAT DID NOT
10474                          ;       OCCUR. BOTH CONDITIONS CAN EXIST AT THE SAME TIME AND MUST
10475                          ;       BE REPORTED.
10476
10477  040502  005737  001250  25$:    TST     $TMP13          ;CHECK IF AN ERROR WAS EXPECTED
10478  040506  001423                  BEQ     110$            ;NO - SKIP
10479  040510  012704  047454          MOV     #GRP1ER,R4      ;GET ADDRESS OF ERROR
10480  040514  012705  001242          MOV     #$TMP10,R5      ;GET ADDRESS OF EXPECTED ERRORS
10481
10482  040520  011500          26$:    MOV     (R5),R0         ;GET EXPECTED ERROR
10483  040522  011401                  MOV     (R4),R1         ;GET GROUP ERROR FLAGS
10484  040524  020001                  CMP     R0,R1           ;ARE THEY EQUAL?
10485  040526  001003                  BNE     27$             ;NO - SKIP
10486  040530  005000                  CLR     R0              ;CLEAR EXPECTED ED
10487  040532  005001                  CLR     R1              ;CLEAR OCCURED
10488  040534  000403                  BR      28$
10489
10490  040536  010003          27$:    MOV     R0,R3           ;STORE EXPECTED ERRORS
10491  040540  040100                  BIC     R1,R0           ;RESET EXPECTED THAT OCCURRED
10492  040542  040301                  BIC     R3,R1           ;RESET OCCURRED THAT EXPECTED
10493  040544  010025          28$:    MOV     R0,(R5)+        ;STORE EXPECTED THAT DID NOT OCCUR
10494  040546  010124                  MOV     R1,(R4)+        ;STORE OCCURRED THAT WERE NOT EXPECTED
10495  040550  022705  001250          CMP     #$TMP13,R5      ;ALL GROUPS CHECKED.
10496  040554  001361                  BNE     26$             ;NO - LOOP
10497
10498                          ;       THE FOLLOWING CODE:
10499                          ;
```

```
10500                                   ;       A. DETERMINES WHICH FORMAT IS TO BE USED
10501                                   ;       B. LOADS THE ADDRESSES OF THE ASCIZ TEXT INTO THE SELECTED
10502                                   ;          ERROR TABLE ITEM AND FORMAT FIELD
10503                                   ;       C. COUNTS THE NUMBER OF ERRORS THAT MUST BE REPORTED
10504                                   ;       D. GETS DRIVE STATUS IF GROUP 1 ERROR.
10505                                   ;
10506                                   ;       THE DECISION OF WHICH ERROR IS TO BE USED
10507                                   ;       IS BASED ON THE ERROR GROUP (OR GROUPS) THAT HAVE
10508                                   ;       FLAGS SET. IF ANY BIT IS SET IN GROUP 1,2, OR 3, GROUP 1 FORMAT (ERROR 4 OR 5)
10509                                   ;       WILL BE USED; ANY SET IN GROUP 2 OR 3, GROUP 2 (ERROR 6) WILL BE USED; AND A
10510                                   ;       FLAG SET IN GROUP 3 ONLY, GROUP 3 (ERROR 7) IS USED.
10511                                   ;
10512                                   ;       THE FORMAT TO BE USED IN THE CONTROLLING FACTOR IN HOW THE
10513                                   ;       ERROR TRAP IS CHANGED IN THE MAIN CALL. IF GROUP 1 FORMAT IS
10514                                   ;       USED THE ERROR TRAP WILL BE CHANGED TO ERROR 4 OR 5(DEPENDING ON
10515                                   ;       AVAILABILITY OF DRIVE STATUS), GROUP 2 FORMAT WILL BE ERROR
10516                                   ;       6, AND GROUP 3 WILL BE ERROR 7. ONLY THE LOW ORDER BYTE
10517                                   ;       OF THE ERROR TRAP WILL BE ALTERED. THE SP WILL BE
10518                                   ;       POINTING TO THE LOCATION THAT CONTAINS THE ERROR CALL TRAP.
10519                                   ;
10520                                   ;       IF THE STATUS IS READ FROM THE DRIVE WITH NO PROBLEM,
10521                                   ;       ERROR 4 IS USED. IF ANY ERROR IS ENCOUNTERED READING
10522                                   ;       STATUS, ERROR 5 IS USED. ERROR 5 INCLUDES A WARNING MESSAGE.
10523
10524  040556  005004          110$:  CLR     R4              ;CLEAR COUNTERS
10525  040560  005005                 CLR     R5
10526  040562  012700  001224         MOV     #$TMP1,R0       ;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
10527  040566  012701  001226         MOV     #$TMP2,R1       ;WHERE ASCIZ ADDRESSES GO
10528  040572  012703  047576         MOV     #GPSUMF,R3      ;POINTERS TO GROUP SUMMARY FLAGS
10529  040576  012710  060750         MOV     #DF007A,(R0)    ;PRESET FOR GRP3 ERR MESSAGE BUILD
10530  040602  012711  001442         MOV     #DH7N,(R1)
10531  040606  013746  047460         MOV     GRP3ER,-(SP)    ;GET GROUP 3 ERRORS, PUT ON STACK
10532  040612  004437  035730         JSR     R4,BITCNT       ;GO COUNT NUMBER AT ERRORS
10533  040616  005716                 TST     (SP)            ;ANY ERRORS?
10534  040620  001403                 BEQ     29$             ;NO - SKIP
10535  040622  061605                 ADD     (SP),R5         ;ADD IN ERROR TOTAL
10536  040624  052713  000004         BIS     #GRP3ST,(R3)    ;SET BIT TO INDICATE GROUP 3 ERROR
10537
10538  040630  005726          29$:   TST     (SP)+           ;CLEAR OFF STACK
10539  040632  005737  001250         TST     $TMP13          ;ERROR EXPECTED
10540  040636  001412                 BEQ     31$             ;NO - SKIP
10541  040640  013746  001246         MOV     $TMP12,-(SP)    ;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
10542  040644  004437  035730         JSR     R4,BITCNT       ;COUNT NUMBER OF ERRORS.
10543  040650  005716                 TST     (SP)            ;WERE THERE ANY
10544  040652  001403                 BEQ     30$             ;NO - SKIP
10545  040654  052713  000040         BIS     #GP3NR,(R3)     ;SET GROUP 3 NOT RECEIVED ERROR FLAG
10546  040660  061604                 ADD     (SP),R4         ;ADD COUNT TO TOTAL THESE
10547
10548  040662  005726          30$:   TST     (SP)+           ;CLEAR OFF STACK
10549  040664  013746  047456  31$:   MOV     GRP2ER,-(SP)    ;GET GROUP 2 ERRORS FOR COUNTING
10550  040670  004437  035730         JSR     R4,BITCNT       ;COUNT BITS
10551  040674  005716                 TST     (SP)            ;ANY SET?
10552  040676  001407                 BEQ     32$             ;NO - SKIP
10553  040700  052713  000002         BIS     #GRP2ST,(R3)    ;SET FLAG FOR GROUP 2 ERRORS
10554  040704  061605                 ADD     (SP),R5         ;ADD INTO TOTAL
10555  040706  012710  060724         MOV     #DF006A,(R0)    ;STORE ADDRESS FOR BUILDING REPORT
```

```
10556  040712  012711  001432                  MOV     #DH6N,(R1)
10557
10558  040716  005726                  32$:    TST     (SP)+           ;CLEAR OFF STACK
10559  040720  005737  001250                  TST     $TMP13          ;ANY EXPECTED ERRORS
10560  040724  001416                          BEQ     34$             ;NO - SKIP
10561  040726  013746  001244                  MOV     $TMP11,-(SP)    ;GET GROUP 2 NOT RECEIVED ERRORS
10562  040732  004437  035730                  JSR     R4,BITCNT       ;COUNT NUMBER OF BITS
10563  040736  005716                          TST     (SP)            ;ANY SET?
10564  040740  001407                          BEQ     33$             ;NO - SKIP
10565  040742  052713  000020                  BIS     #GP2NR,(R3)     ;SET FLAG FOR GROUP 2 NOT RECEIVED
10566  040746  061604                          ADD     (SP),R4         ;ADD INTO TOTAL
10567  040750  012710  060724                  MOV     #DF006A,(R0)    ;STORE ADDRESS FOR BUILDING REPORT
10568  040754  012711  001432                  MOV     #DH6N,(R1)
10569
10570  040760  005726                  33$:    TST     (SP)+           ;CLEAR OFF STACK
10571  040762  013746  047454          34$:    MOV     GRP1ER,-(SP)    ;GET GROUP 1 ERROR FLAGS
10572  040766  004437  035730                  JSR     R4,BITCNT       ;COUNT THE NUMBER OF BITS
10573  040772  005716                          TST     (SP)            ;ANY SET?
10574  040774  001407                          BEQ     35$             ;NO - SKIP
10575  040776  052713  000001                  BIS     #GRP1ST,(R3)    ;SET FLAG FOR GROUP 1 ERRORS SET
10576  041002  061605                          ADD     (SP),R5         ;ADD INTO TOTAL
10577  041004  012710  060640                  MOV     #DF004A,(R0)    ;LOAD ADDRESS FOR BUILDING REPORT
10578  041010  012711  001412                  MOV     #DH4N,(R1)
10579  041014  005726                  35$:    TST     (SP)+           ;CLEAR OFF STACK
10580  041016  005737  001250                  TST     $TMP13          ;ANY EXPECTED ERRORS?
10581  041022  001416                          BEQ     60$             ;NO - SKIP
10582  041024  013746  001242                  MOV     $TMP10,-(SP)    ;GET GROUP 1 NO RECEIVED ERROR
10583  041030  004437  035730                  JSR     R4,BITCNT       ;COUNT # OF BITS
10584  041034  005716                          TST     (SP)            ;ANY SET?
10585  041036  001407                          BEQ     36$             ;NO - SKIP
10586  041040  052713  000010                  BIS     #GP1NR,(R3)     ;SET FLAG FOR GROUP 1 NOT RECEIVED
10587  041044  061604                          ADD     (SP),R4         ;ADD INTO TOTAL
10588  041046  012710  060640                  MOV     #DF004A,(R0)    ;LOAD ADDRESS FOR BUILDING REPORT
10589  041052  012711  001412                  MOV     #DH4N,(R1)
10590  041056  005726                  36$:    TST     (SP)+           ;CLEAR OFF STACK.
10591  041060  032713  000011          60$:    BIT     #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERROR
10592  041064  001414                          BEQ     52$             ;NO - SKIP
10593  041066  042713  040000                  BIC     #DRSTER,(R3)
10594  041072  004437  036442                  JSR     R4,GETDRS
10595  041076  000401                          BR      51$             ;ERROR RETURN
10596  041100  000406                          BR      52$             ;NO ERROR RETURN
10597  041102  012710  060670          51$:    MOV     #DF005A,(R0)    ;CHANGE TO FORMAT 5 - STORE ADDRESS
10598  041106  012711  001422                  MOV     #DH5N,(R1)      ;FOR BUILDING REPORT
10599  041112  052713  040000                  BIS     #DRSTER,(R3)    ;SET DRIVE STATUS ERROR
10600  041116                          52$:
10601
10602                                          ;       THE ERRORS ARE COUNTED, FLAGS SET TO INDICATE WHICH ERRORS
10603                                          ;       ARE TO BE REPORTED, AND THE ERROR FORMAT HAS BEEN SELECTED.
10604                                          ;       THE FOLLOWING CODE WILL TYPE ALL THE ERRORS, LOAD THE
10605                                          ;       PROPER HEADER MESSAGE ADDRESS IN THE ERROR ITEM TABLE
10606                                          ;       AND LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE PROPER
10607                                          ;       DF TABLE.
10608                                          ;
10609                                          ;       AT THIS TIME
10610                                          ;           R5 CONTAINS EITHER THE NUMBER OF ERRORS THAT OCCURRED BUT WERE
10611                                          ;               NOT EXPECTED OR
```

```
10612                                   ;               THE NUMBER OF ERRORS THE OCCURRED IF NONE WERE EXPECTED
10613                                   ;           R4 CONTAINS THE NUMBER OF ERRORS THAT WERE EXPECTED BUT
10614                                   ;               DID NOT OCCUR.
10615                                   ;               $TMP10 CONTAINS GROUP 1 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
10616                                   ;               $TMP11 CONTAINS GROUP 2 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
10617                                   ;               $TMP12 CONTAINS GROUP 3 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
10618                                   ;               GRP1ER CONTAINS GROUP 1 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
10619                                   ;               GRP2ER CONTAINS GROUP 2 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
10620                                   ;               GRP3ER CONTAINS GROUP 3 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
10621                                   ;           (R1)=#$TMP2 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
10622                                   ;               OF THAT MUST BE ALTERED TO IDENTIFY THE OPERATION
10623                                   ;           (R0)=#$TMP1 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
10624                                   ;               THE ERROR ITEM TABLE THAT MUST BE ALTERED TO PROVIDE A
10625                                   ;               PROPER MESSAGE TO REPORT.
10626                                   ;           (R3)=#GRSUMF CONTAIN FLAGS TO INDICATE WHICH OF THE GROUP
10627                                   ;               ERROR FLAG FIELDS HAVE ERROR BITS STORED.
10628  041116  032777  020000  14001            BIT     #SW13,@SWR      ;IS REPORT INHIBITED?
10629  041124  001402                           BEQ     37$             ;NO - SKIP
10630  041126  000137  041514                   JMP     49$             ;ELSE EXIT
10631  041132  005737  001250           37$:    TST     $TMP13          ;WERE ERRORS EXPECTED?
10632  041136  001004                           BNE     38$             ;YES - SKIP
10633
10634                                   ;       IF NO ERRORS WERE EXPECTED, $TMP10,11, &12 ARE NOT MEANINGFUL
10635
10636  041140  012771  057643  000000           MOV     #DH007,@(R1)    ;HEADER = ERROR IN OPERATION
10637  041146  000411                           BR      39$
10638  041150  012771  057464  000000   38$:    MOV     #DH005,@(R1)    ;PRESET HDRMSG = EXPECTED NOT SET
10639  041156  032713  000070                   BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS?
10640  041162  001003                           BNE     39$             ;YES - SKIP
10641  041164  012771  057561  000000           MOV     #DH006,@(R1)    ;SET MESSAGE TO UNEXPECTED ERROR SET
10642  041172  013701  001540           39$:    MOV     T.CS1,R1        ;GET CS1
10643  041176  042701  177741                   BIC     #177741,R1      ;CLEAR ALL BUT COMMAND
10644
10645  041202  016170  047600  000000           MOV     CMNDLB(R1),@(R0) ;MOVE ADDRESS OF COMMAND MESSAGE
10646                                                                   ;INTO REPORT
10647  041210  032713  000007                   BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY GPR ERRORS?
10648  041214  001462                           BEQ     46$             ;NO - SKIP GPR REPORT
10649
10650                                   ;       PRINT ALL THE ERRORS CONTAINED IN THE GPR1,2,3ER(UNEXPECTED ERRORS)
10651
10652  041216  013701  047460                   MOV     GRP3ER,R1       ;GET GROUP 3 ERROR FLAGS
10653  041222  012700  047500                   MOV     #GRP3MS,R0      ;SET POINTER TO GRP3 ERROR MESSAGES
10654  041226  005037  001252           140$:   CLR     $TMP14          ;CLEAR GROUP PRINTING INDICATOR
10655  041232  012737  000021  001254   40$:    MOV     #17.,$TMP15     ;PRESET SHIFT COUNT
10656  041240  000241                           CLC                     ;CLEAR CARRY
10657  041242  006001                   41$:    ROR     R1              ;ROTATE ERROR FLAGS
10658  041244  103406                           BCS     42$             ;WAS BIT SHIFTED OUT SET?
10659  041246  062700  000002           141$:   ADD     #2,R0           ;NO - BUMP POINTER
10660  041252  005337  001254                   DEC     $TMP15          ;DEC SHIFT COUNT
10661  041256  001371                           BNE     41$             ;LOOP IF SHIFT NOT ZERO
10662  041260  000411                           BR      44$
10663  041262  011037  041274           42$:    MOV     (R0),43$        ;GET ERROR MESSAGE ADDRESS FROM TABLE
10664  041266  104401  001273                   TYPE    ,$CRLF          ;TYPE CRLF
10665  041272  104401                           TYPE                    ;TYPE ERROR MESSAGE
10666  041274  000000                   43$:    .WORD                   ;MESSAGE ADDRESS GOES HERE
10667  041276  005305                           DEC     R5              ;DECEREMENT TOTAL ERROR COUNT.
```

```
10668  041300  001362                    BNE     141$            ;LOOP IF ZERO
10669  041302  000427                    BR      46$             ;ELSE EXIT GPR ERROR PRINT LOOP
10670
10671  041304  005713            44$:    TST     (R3)            ;TEST GPSUMF FLAG FOR PRINTING ERROR NOT RECEIVED
10672  041306  100455                    BMI     47$             ;YES - SKIP
10673  041310  005737  001252            TST     $TMP14          ;PRINTING GROUP 3?
10674  041314  001007                    BNE     45$             ;NO -SKIP
10675  041316  013701  047456            MOV     GRP2ER,R1       ;ELSE SET TO GROUP 2, GET GRP2ER
10676  041322  012700  047520            MOV     #GRP2MS,R0      ;& SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
10677  041326  005237  001252            INC     $TMP14          ;BUMP TO INDICATE PRINTING GROUP 2
10678  041332  000737                    BR      40$             ;GO RESTART PRINT LOOP
10679  041334  022737  000002  001252  45$:    CMP     #2,$TMP14       ;PRINTING GROUP 1?
10680  041342  001407                    BEQ     46$             ;YES - EXIT GPR ERROR PRINT LOOP.
10681  041344  013701  047454            MOV     GRP1ER,R1       ;ELSE SET TO GROUP 1, GET GROUP 1 ERROR
10682  041350  012700  047536            MOV     #GRP1MS,R0      ;SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
10683  041354  005237  001252            INC     $TMP14          ;BUMP TO INDICATE PRINTING GROUP 1
10684  041360  000724                    BR      40$             ;RESTART PRINT LOOP.
10685
10686  041362  005737  001250    46$:    TST     $TMP13          ;EXPECTING ERRORS?
10687  041366  001452                    BEQ     49$             ;NO - SKIP
10688
10689                           ;       PRINT ALL ERRORS CONTAINED IN $TMP10, 11, 12(NOT RECEIVED ERRORS)
10690
10691  041370  005713                    TST     (R3)            ;TEST IF PRINTING NOT RECEIVED ERRORS
10692  041372  100423                    BMI     47$             ;YES - SKIP
10693  041374  032713  000070            BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS
10694  041400  001445                    BEQ     49$             ;NO - SKIP
10695  041402  032713  000007            BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY NOT RECEIVED ERRORS?
10696  041406  001404                    BEQ     146$            ;NO - SKIP LABEL FOR UNEXPECTED ERRORS
10697  041410  104401  001273            TYPE    ,$CRLF          ;TYPE CRLF
10698  041414  104401  057561            TYPE    ,DH006          ;TYPE HEADER FOR PREVIOUS ERRORS
10699  041420  052737  100000  047576  146$:    BIS     #REPNR,GPSUMF   ;SET PRINTING NOT RECEIVED ERRORS SWITCH
10700  041426  010405                    MOV     R4,R5           ;MOVE TOTAL ERRORS TO R5
10701  041430  013701  001246            MOV     $TMP12,R1       ;GET GRP3 NOT RECEIVED ERRORS
10702  041434  012700  047500            MOV     #GRP3MS,R0      ;SET POINTER TO GROUP 3 MESSAGES
10703  041440  000672                    BR      140$            ;GO START PRINT LOOP
10704  041442  005737  001252    47$:    TST     $TMP14          ;PRINTING GROUP 3?
10705  041446  001007                    BNE     48$             ;NO - SKIP
10706  041450  013701  001244            MOV     $TMP11,R1       ;ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
10707  041454  012700  047520            MOV     #GRP2MS,R0      ;& SET POINTER TO GROUP 2 MESSAGE TABLE
10708  041460  005237  001252            INC     $TMP14          ;BUMP TO INDICATE GROUP 2 PRINTING
10709  041464  000662                    BR      40$             ;GO START PRINT LOOP
10710  041466  022737  000002  001252  48$:    CMP     #2,$TMP14       ;PRINTING GROUP 1?
10711  041474  001407                    BEQ     49$             ;YES - EXIT LOOP
10712  041476  013701  001242            MOV     $TMP10,R1       ;SET POINTER TO GROUP 1 MESSAGE
10713  041502  012700  047536            MOV     #GRP1MS,R0      ;TABLE AND GET GROUP 1 ERRORS.
10714  041506  005237  001252            INC     $TMP14          ;BUMP TO INDICATE GROUP 1 PRINTING
10715  041512  000647                    BR      40$             ;START LOOP AGAIN.
10716
10717  041514  032713  000077    49$:    BIT     #77,(R3)        ;TEST IF ANY ERRORS TO BE REPORTED
10718                                                           ;  GRP1ST!GRP2ST!GRP3ST
10719                                                           ;  GP1NR!GP2NR!GP3NR
10720  041520  001004                    BNE     61$             ;YES - SKIP
10721  041522  104414                    RESREG                  ;ELSE EXIT
10722  041524  062716  000002            ADD     #2,(SP)         ;BUMP FOR GOOD RETURN
10723  041530  000002                    RTI
```

```
10724
10725  041532  112776  000007  000000  61$:    MOVB    #7,@(SP)          ;PRESET FOR GROUP 3 ERROR RETURN.
10726  041540  032713  000022            BIT     #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
10727  041544  001403                    BEQ     50$               ;NO - SKIP
10728  041546  112776  000006  000000    MOVB    #6,@(SP)          ;ELSE SET FOR GROUP 2 ERROR RETURN
10729
10730  041554  032713  000011            50$:    BIT     #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
10731  041560  001411                    BEQ     53$               ;NO - SKIP
10732  041562  112776  000004  000000    MOVB    #4,@(SP)          ;ELSE SET FOR GROUP 1 ERROR RETURN.
10733  041570  032713  040000            BIT     #DRSTER,(R3)      ;CHECK IF ERROR GETTING DRIVE STATUS
10734  041574  001403                    BEQ     53$               ;NO - SKIP
10735  041576  112776  000005  000000    MOVB    #5,@(SP)          ;ELSE CHANGE RETURN FORM GROUP 1
10736
10737  041604  005737  001264            53$:    TST     $ESCAPE           ;CHECK IF ESCAPE ALREADY SET
10738  041610  001011                    BNE     54$               ;YES - SKIP
10739  041612  013700  001302            MOV     $TESTN,R0         ;SET UP $ESCAPE TO FORCE
10740  041616  006300                    ASL     R0                ;ABORT TO PRESENT TEST AFTER
10741  041620  016037  033032  001264    MOV     $SW08TB(R0),$ESCAPE ;ERROR IS REPORTED
10742  041626  162737  000002  001264    SUB     #2,$ESCAPE        ;BUT GO TO NEXT SCOPE STATEMENT
10743  041634  104414                    54$:    RESREG
10744  041636  000002                    RTI                       ;RETURN
10745
10746                                    ;;*****************************************************************
10747                                    .SBTTL  BAD SECTOR CHECK
10748                                    ;*      THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
10749                                    ;*      CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN
10750                                    ;*      THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN
10751                                    ;*      L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD
10752                                    ;*      THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER
10753                                    ;*      IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS
10754                                    ;*      IS PLACED ON THE STACK AFTER THE ENTIRE
10755                                    ;*      FIELD IS SEARCHED.
10756                                    ;*
10757                                    ;*      CALL:   JSR     R4,BDSRCK
10758                                    ;*              <ADDRESS OF FIELD TO BE SEARCHED>
10759                                    ;;*****************************************************************
10760
10761  041640  012637  001236            BDSRCK: MOV     (SP)+,$TMP6       ;STORE OLD R4 CONTENTS
10762  041644  010437  001240            MOV     R4,$TMP7          ;GET RETURN ADDRESS
10763  041650  011404                    MOV     (R4),R4           ;GET POINTER TO FIELD TO BE CHECKED
10764  041652  005037  001234            CLR     $TMP5             ;CLEAR A COUNTER
10765  041656  005714                    1$:     TST     (R4)              ;TEST IF FIELD HAS NO (OR NO MORE)ENTRIES
10766  041660  100417                    BMI     4$                ;YES - EXIT
10767  041662  023724  001614            CMP     L.DCYL,(R4)+      ;IS THIS ENTRY FOR THIS CYLINDER?
10768  041666  001012                    BNE     3$                ;NO - SKIP
10769  041670  005204                    INC     R4                ;BUMP TO TRACK
10770  041672  123714  001607            CMPB    L.DT,(R4)         ;IS ENTRY FOR THIS TRACK?
10771  041676  001005                    BNE     2$                ;NO - SKIP
10772  041700  005046                    CLR     -(SP)             ;CLEAR STACK LOCATION
10773  041702  114416                    MOVB    -(R4),(SP)        ;PUT SECTOR NUMBER ON STACK
10774  041704  005237  001234            INC     $TMP5             ;BUMP COUNTER
10775  041710  000401                    BR      3$                ;BRANCH
10776
10777  041712  005304                    2$:     DEC     R4                ;DECREMENT POINTER TO WORD ALIGN
10778  041714  005724                    3$:     TST     (R4)+             ;BUMP TO NEXT ENTRY
10779  041716  000757                    BR      1$                ;TEST NEXT ENTRY
```

```
10780
10781  041720  013746  001234          4$:     MOV     $TMP5,-(SP)     ;PUT COUNT ON STACK
10782  041724  013746  001236                  MOV     $TMP6,-(SP)     ;PUT OLD R4 CONTENTS BACK ON STACK
10783  041730  013704  001240                  MOV     $TMP7,R4        ;SET UP RETURN
10784  041734  005724                          TST     (R4)+           ;BUMP PAST PARAMETER
10785  041736  000204                          RTS     R4              ;RETURN
10786
10787                                  ;;*****************************************************************
10788                                  .SBTTL  DATA GENERATION AND COMPARE ROUTINE
10789                                  ;*  CALLS:      JSR     R4,GENCOM
10790                                  ;*              CONTROL WORD
10791                                  ;*
10792                                  ;*              JSR     R4,GENCOM
10793                                  ;*              CONTROL WORD
10794                                  ;*              LENGTH
10795                                  ;*
10796                                  ;*              JSR     R4,GENCOM
10797                                  ;*              CONTROL WORD
10798                                  ;*              RELOCATION CONSTANT
10799                                  ;*              LENGTH
10800                                  ;*
10801                                  ;*  RETURN:     RTS     R4
10802                                  ;*      R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
10803                                  ;*      THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
10804                                  ;*      CONTROL WORD. THIS IS UNCONDITIONAL.
10805                                  ;*
10806                                  ;*      THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF
10807                                  ;*      THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED.
10808                                  ;*      IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE
10809                                  ;*      ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.
10810                                  ;*
10811                                  ;*      THE THIRD CALL IS IDENTICAL TO THE SECOND.
10812                                  ;*
10813                                  ;*  DEFINITION OF CONTROL WORD:
10814                                  ;*      BIT 15 - DO COMPARE OPERATION OF IBUFF (SOURCE) TO OBUFF
10815                                  ;*               (DESTINATION). EXPECTED VALUES ARE IN OBUFF (DESTINATION.
10816                                  ;*      BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
10817                                  ;*      BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
10818                                  ;*      BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
10819                                  ;*      BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
10820                                  ;*      BIT 10 - CLEAR IBUFF TO PATTERN SELECTED.
10821                                  ;*      BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
10822                                  ;*      BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
10823                                  ;*      BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
10824                                  ;*      BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES
10825                                  ;*               NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES.
10826                                  ;*               OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16.
10827                                  ;*
10828                                  ;*  EXPLANATION OF CALLS:
10829                                  ;*      THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR
10830                                  ;*      BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.
10831                                  ;*
10832                                  ;*      THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED
10833                                  ;*      FOR DATA GENERATION OR COMPARE AND FOR IBUFF INITALIZATION.
10834                                  ;*
10835                                  ;*      THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS
```

```
10836                   ;*      USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.
10837                   ;*
10838                   ;*
10839                   ;*  DESCRIPTION:
10840                   ;*
10841                   ;*      THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:
10842                   ;*        A.    BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE
10843                   ;*              ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8)
10844                   ;*              OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR
10845                   ;*               EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE
10846                   ;*              SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK
10847                   ;*              ARE LISTED THEREIN (BIT 9).
10848                   ;*
10849                   ;*        B.    COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT 15). THE
10850                   ;*              CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A
10851                   ;*              HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH
10852                   ;*              WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8)
10853                   ;*              WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT
10854                   ;*              BIT THAT WAS LAST SPECIFIED IN L.CS1. THE HEADERS
10855                   ;*              MAY BE BUILT BEFORE THE COMPARE AS PART OF THE
10856                   ;*              OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE
10857                   ;*              GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
10858                   ;*
10859                   ;*        C.    RESUME COMPARE OPERATION. IF A COMPARE OPERATION
10860                   ;*              DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER
10861                   ;*              BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE
10862                   ;*              RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14
10863                   ;*              SET IN THE CONTROL WORD.
10864                   ;*
10865                   ;*        D.    DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT.
10866                   ;*              MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER
10867                   ;*              SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS
10868                   ;*              MANNER, DATA GENERATION CAN BE MADE TO
10869                   ;*              PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE
10870                   ;*              DATA COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO
10871                   ;*              ANY AREA OF AVAILABLE MEMORY.
10872
10873   041740                   GENCOM:
10874   041740  010046                   MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10875   041742  010146                   MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10876   041744  010346                   MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10877   041746  010546                   MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10878   041750  012400                   MOV     (R4)+,R0        ;GET PARAMETER WORD
10879   041752  012737  056210  001520   MOV     #EM54,EM15N     ;PRESET FOR HEADER COMPARE ERROR
10880   041760  032700  000200          BIT     #BIT7,R0        ;HEADER OPERATION SPECIFIED?
10881   041764  001005                   BNE     18$             ;YES - SKIP
10882   041766  012737  056237  001520   MOV     #EM55,EM15N     ;CHANGE FOR DATA COMPARE ERROR
10883   041774  000137  042526          JMP     17$             ;ELSE JUMP TO DATA ROUTINE
10884   042000                   18$:
10885   042000  010446                   MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10886   042002  032700  001400          BIT     #BIT8!BIT9,R0   ;MUST HEADERS BE BUILT?
10887   042006  001002                   BNE     19$             ;YES - SKIP
10888   042010  000137  042310          JMP     11$             ;ELSE JUMP TO HEADER COMPARE
10889   042014  113701  001607   19$:    MOVB    L.DT,R1         ;START HEADER BUILD ROUTINE
10890   042020  013703  001614          MOV     L.DCYL,R3       ;GET TRACK AND CYL
10891   042024  012705  000005          MOV     #5,R5           ;SET COUNT TO SHIFT TRACK FOR HDR WORD
```

```
10892
10893  042030  006301                  1$:     ASL     R1              ;SHIFT TRACK
10894  042032  005305                          DEC     R5              ;DECREMENT TRACK
10895  042034  001375                          BNE     1$              ;LOOP UNTIL COUNT 0
10896
10897  042036  012704  000026                  MOV     #26,R4          ;PRESET FOR 26 SECTOR MODE
10898  042042  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS IT 24 SECTOR MODE?
10899  042050  001404                          BEQ     2$              ;NO - SKIP
10900  042052  012704  000024                  MOV     #24,R4          ;CHANGE COUNT FOR 24 SECTOR MODE
10901  042056  052701  001000                  BIS     #BIT9,R1        ;SET 24 SECTOR MODE BIT IN WRD 2 OF HDR
10902
10903  042062  052701  140000          2$:     BIS     #BIT15!BIT14,R1 ;SET BS BITS TO INDICATE GOOD SECTOR
10904  042066  012705  063414                  MOV     #OBUFF,R5       ;SET POINTER TO ADDRESS WHERE HEADERS GO
10905  042072  010325                  3$:     MOV     R3,(R5)+        ;INSERT CYLINDER
10906  042074  010125                          MOV     R1,(R5)+        ;INSERT TRACK AND SECTOR
10907  042076  010337  001224                  MOV     R3,$TMP1        ;CALCULATE HVRC WORD
10908  042102  010115                          MOV     R1,(R5)
10909  042104  040137  001224                  BIC     R1,$TMP1
10910  042110  040315                          BIC     R3,(R5)
10911  042112  053725  001224                  BIS     $TMP1,(R5)+     ;COMPLETE HVRC WORD INSERTION
10912
10913  042116  005304                          DEC     R4              ;DECREMENT HEADER COUNT
10914  042120  001402                          BEQ     4$              ;DONE? - YES, SKIP
10915  042122  005201                          INC     R1              ;BUMP SECTOR
10916  042124  000762                          BR      3$              ;LOOP
10917
10918  042126  032700  001000          4$:     BIT     #BIT9,R0        ;MUST HEADERS BE CORRECTED FOR TABLE ENTRIES?
10919  042132  001003                          BNE     5$              ;YES - SKIP
10920  042134  005700                  10$:    TST     R0              ;IS THIS A COMPARE OPERATION?
10921  042136  100464                          BMI     11$             ;YES-GO DO HDR COMPARE
10922  042140  000534                          BR      50$             ;ELSE GET OUT
10923
10924  042142  013737  001640  042200  5$:     MOV     BSF26P,6$       ;PRESET FOR BS FACTORY LIST
10925  042150  012737  100000  001224          MOV     #BIT15,$TMP1    ;SET BIT TO BE RESET IN BAD HEADER
10926  042156  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS THIS 26 SECTOR MODE?
10927  042164  001403                          BEQ     8$              ;YES - SKIP
10928  042166  013737  001636  042200          MOV     BSF24P,6$       ;ELSE CHANGE FOR 24 SECTOR MODE
10929
10930  042174  004437  041640          8$:     JSR     R4,BDSRCK       ;GO CHECK FOR BAD SECTOR THIS ADDRESS
10931  042200  000000                  6$:     .WORD   0               ;POINTER TO FILE TO BE CHECKED GOES HERE
10932  042202  012605                          MOV     (SP)+,R5        ;GET # OF BAD SECTORS THIS PACK ADDRESS
10933  042204  001417                          BEQ     9$              ;SKIP IF ZERO
10934
10935  042206  011601                  7$:     MOV     (SP),R1         ;GET 1ST BAD SECTOR NUMBER
10936  042210  006301                          ASL     R1              ;MULTIPLY SECTOR NUMBER BY 6 TO
10937  042212  006301                          ASL     R1              ;LOACATE SECTOR TO BE MARKED BAD
10938  042214  061601                          ADD     (SP),R1
10939  042216  062601                          ADD     (SP)+,R1
10940  042220  062701  000002                  ADD     #2,R1           ;ADD 2 FOR 2ND WORD THAT SECTOR
10941  042224  043761  001224  063414          BIC     $TMP1,OBUFF(R1) ; CLEAR BIT FOR BAD SECTOR IN HDR
10942  042232  043761  001224  063416          BIC     $TMP1,OBUFF+2(R1) ; CORRECT THE HVRC BIT
10943  042240  005305                          DEC     R5              ;DECREMENT BAD SECTOR COUNT
10944  042242  001361                          BNE     7$              ;LOOP IF NOT ZERO
10945
10946  042244  032737  100000  001224  9$:     BIT     #BIT15,$TMP1    ;WERE WE DOING BS FACTORY LIST?
10947  042252  001730                          BEQ     10$             ;NO - GO CHECK IF COMPARE MUST BE DONE
```

```
10948  042254  012737  040000  001224          MOV     #BIT14,$TMP1    ;ELSE SET BIT TO BE RESET IN BAD HDR
10949  042262  013737  001644  042200          MOV     BSC26P,6$       ;PRESET POINTER FOR 26 SECTOR MODE
10950  042270  032737  010000  001600          BIT     #CFMT,L.CS1     ;TEST IF WE ARE DOING 26 SECTOR MODE
10951  042276  001736                          BEQ     8$              ;YES - SKIP TO START CHECK
10952  042300  013737  001642  042200          MOV     BSS24P,6$       ;CHANGE POINTER TOR 24 SECTOR MODE
10953  042306  000732                          BR      8$              ;SKIP TO START CHECK
10954
10955                                  ;       START OF COMPARE
10956  042310  012701  000102          11$:    MOV     #102,R1         ;PRESET FOR 102 WORDS OF HEADER
10957  042314  032737  010000  001600          BIT     #CFMT,L.CS1     ;CHECK IF 26 SECTOR MODE
10958  042322  001402                          BEQ     12$             ;YES - SKIP
10959  042324  012701  000074                  MOV     #74,R1          ;CHANGE TO 74 WORDS OF HEADER
10960
10961  042330  012704  061414          12$:    MOV     #IBUFF,R4       ;SET START OF HEADERS TO BE COMPARED
10962  042334  012705  063414                  MOV     #OBUFF,R5       ;SET START OF GOOD HEADERS
10963  042340  005003                          CLR     R3              ;CLEAR COUNTER
10964  042342  032700  040000                  BIT     #BIT14,R0       ;IS THIS A CONTINUATION OF EARLIER COMPARE
10965  042346  001412                          BEQ     13$             ;NO - SKIP
10966  042350  013705  001700          28$:    MOV     DESHLD,R5       ;GET VALUES WHERE PREVIOUS CHECK STOPPED
10967  042354  013704  001702                  MOV     SRCHLD,R4       ;  DESTINATION AND SOURCE
10968  042360  013703  001704                  MOV     WRDNUM,R3       ;  WORD NUMBER IN ERROR
10969  042364  013701  001706                  MOV     WRDCNT,R1       ;  WORD COUNT LEFT IN COMPARE
10970  042370  005701                          TST     R1              ;TEST IF WORD COUNT LEFT = 0
10971  042372  001417                          BEQ     50$             ;YES - EXIT
10972
10973  042374  032700  030000          13$:    BIT     #BIT12!BIT13,R0 ;MEM MANAGE REQUIRED?
10974  042400  001402                          BEQ     25$             ;NO - SKIP
10975  042402  005237  177572                  INC     @#SR0           ;TURN IT ON
10976  042406  022425                  25$:    CMP     (R4)+,(R5)+     ;COMPARE THE WORDS
10977  042410  001012                          BNE     14$             ;SKIP IF NOT EQUAL
10978  042412  005203                          INC     R3              ;BUMP WORD NUMBER IN ERROR
10979  042414  005301                          DEC     R1              ;DEC WORD COUNT LEFT IN COMPARE
10980  042416  001373                          BNE     25$             ;LOOP IF NOT ZERO
10981  042420  032700  030000                  BIT     #BIT12!BIT13,R0 ;MEM MANAGE IN USE?
10982  042424  001402                          BEQ     50$             ;NO - SKIP
10983  042426  005337  177572                  DEC     @#SR0           ;TURN IT OFF
10984  042432                          50$:
10985  042432  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10986  042434  000427                          BR      16$
10987
10988                                  ;       ERROR REPORT PREP AND PARAMETER STORAGE FOR CONTINUATION
10989
10990  042436  010537  001700          14$:    MOV     R5,DESHLD       ;STORE DESTINATION
10991  042442  010437  001702                  MOV     R4,SRCHLD       ;       SOURCE
10992  042446  014537  001202                  MOV     -(R5),$REG10    ;LOAD GOOD WORD FOR REPORT
10993  042452  014437  001204                  MOV     -(R4),$REG11    ;       BAD WORD
10994  042456  010337  001206                  MOV     R3,$REG12       ;       WORD NUMBER
10995  042462  005301                          DEC     R1              ;DEC COUNT LEFTFOR CONTINUATION
10996  042464  005203                          INC     R3              ;BUMP BAD WORD NUMBER
10997  042466  010137  001706                  MOV     R1,WRDCNT       ;STORE COUNT LEFT
10998  042472  010337  001704                  MOV     R3,WRDNUM       ;       WORD NUM IN ERROR
10999  042476  032700  030000                  BIT     #BIT12!BIT13,R0 ;MEM MANAGE IS USE?
11000  042502  001402                          BEQ     15$             ;NO - SKIP
11001  042504  005337  177572                  DEC     @#SR0           ;TURN IT OFF
11002
11003  042510                          15$:
```

```
11004  042510  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
11005  042512  005724                        TST     (R4)+           ;ERROR RETURN
11006
11007  042514                        16$:
11008  042514  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
11009  042516  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
11010  042520  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
11011  042522  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
11012  042524  000204                        RTS     R4
11013
11014                            ;       DATA PATTERN PROCESSING ROUTINE
11015
11016  042526  032700  040000    17$:    BIT     #BIT14,R0       ;CONTINUE WITH COMPARE?
11017  042532  001402                    BEQ     29$             ;NO - SKIP
11018  042534  010446                    MOV     R4,-(SP)        ;STORE RETURN
11019  042536  000704                    BR      28$             ;GO CONTINUE COMPARE
11020
11021  042540  012705  063414    29$:    MOV     #OBUFF,R5       ;GET DESTINATION
11022  042544  012703  061414            MOV     #IBUFF,R3       ;GET SOURCE
11023  042550  032700  030000            BIT     #BIT12!BIT13,R0 ;USE MEM MANAGE?
11024  042554  001412                    BEQ     21$             ;NO - SKIP
11025
11026  042556  012437  172354            MOV     (R4)+,@#KIPAR6  ;LOAD PAR FOR RELOCATION
11027  042562  032700  010000            BIT     #BIT12,R0       ;RELOCATE SOURCE?
11028  042566  001403                    BEQ     20$             ;NO - SKIP
11029  042570  012705  140070            MOV     #140070,R5      ;SET DESTINATION TO USE PAR6 + OFFSET
11030  042574  000402                    BR      21$             ;SKIP
11031  042576  012703  140070    20$:    MOV     #140070,R3      ;SET SOURCE TO USE PAR6 + OFFSET
11032
11033  042602  012401                21$:    MOV     (R4)+,R1        ;STORE COUNT
11034  042604  010446                        MOV     R4,-(SP)        ;STORE RETURN
11035  042606  010304                        MOV     R3,R4           ;PUT IN IBUFF POINTER
11036  042610  005003                        CLR     R3              ;CLEAR R3 FOR WORD NUMBER COUNTER
11037  042612  032700  000077                BIT     #77,R0          ;ANY DATA PATTERN SPECIFIED?
11038  042616  001666                        BEQ     13$             ;NO - GO DO COMPARE
11039
11040                            ;       START OF GENERATION ROUTINE
11041
11042  042620  010537  001700                MOV     R5,DESHLD       ;STORE PARAMETERS FOR COMPARE
11043  042624  010437  001702                MOV     R4,SRCHLD
11044  042630  010337  001704                MOV     R3,WRDNUM
11045  042634  010137  001706                MOV     R1,WRDCNT
11046
11047                            ;       CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
11048                            ;       MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
11049                            ;       WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
11050
11051  042640  032700  030000                BIT     #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
11052  042644  001402                        BEQ     33$             ;NO - SKIP
11053  042646  005237  177572                INC     @#SR0           ;TURN IT ON
11054  042652  032700  002000    33$:    BIT     #BIT10,R0       ;GENERATE PATTERN IN IBUFF?
11055  042656  001403                    BEQ     32$             ;NO - SKIP
11056  042660  010446                    MOV     R4,-(SP)        ;ELSE SWAP R4 AND R5
11057  042662  010504                    MOV     R5,R4
11058  042664  012605                    MOV     (SP)+,R5
11059
```

```
11060  042666  122700  000001  32$:    CMPB    #1,R0           ;PATTERN 1 (ALL ZEROS)?
11061  042672  001004                   BNE     55$             ;NO - SKIP
11062  042674  005025          30$:     CLR     (R5)+           ;CLEAR WORD IN BUFF
11063  042676  005301                   DEC     R1              ;DEC WORD COUNT
11064  042700  001375                   BNE     30$             ;LOOP UNTIL WORD COUNT ZERO
11065  042702  000550                   BR      22$             ;EXIT BUILD
11066
11067  042704  122700  000007  55$:     CMPB    #7,R0           ;PATTERN 7 (ALL ONES)?
11068  042710  001005                   BNE     56$             ;NO - SKIP
11069  042712  012725  177777  31$:     MOV     #-1,(R5)+       ;LOAD WORD IN BUFF
11070  042716  005301                   DEC     R1              ;DEC WORD COUNT
11071  042720  001374                   BNE     31$             ;LOOP UNTIL WORD COUNT ZERO
11072  042722  000540                   BR      22$             ;EXIT BUILD
11073
11074  042724  122700  000002  56$:     CMPB    #2,R0           ;PATTERN 2 SET UP
11075  042730  001003                   BNE     57$
11076  042732  012703  046624           MOV     #PAT02,R3
11077  042736  000504                   BR      70$
11078
11079  042740  122700  000003  57$:     CMPB    #3,R0           ;PATTERN 3 SET UP
11080  042744  001003                   BNE     58$
11081  042746  012703  046664           MOV     #PAT03,R3
11082  042752  000476                   BR      70$
11083
11084  042754  122700  000004  58$:     CMPB    #4,R0           ;PATTERN 4 SET UP
11085  042760  001003                   BNE     59$
11086  042762  012703  046724           MOV     #PAT04,R3
11087  042766  000470                   BR      70$
11088
11089  042770  122700  000005  59$:     CMPB    #5,R0           ;PATTERN 5 SET UP
11090  042774  001003                   BNE     60$
11091  042776  012703  046764           MOV     #PAT05,R3
11092  043002  000462                   BR      70$
11093
11094  043004  122700  000006  60$:     CMPB    #6,R0           ;PATTERN 6 SET UP
11095  043010  001003                   BNE     61$
11096  043012  012703  047024           MOV     #PAT06,R3
11037  043016  000454                   BR      70$
11098
11099  043020  122700  000010  61$:     CMPB    #10,R0          ;PATTERN 10 SET UP
11100  043024  001003                   BNE     62$
11101  043026  012703  047064           MOV     #PAT10,R3
11102  043032  000446                   BR      70$
11103
11104  043034  122700  000011  62$:     CMPB    #11,R0          ;PATTERN 11 SET UP
11105  043040  001003                   BNE     63$
11106  043042  012703  047124           MOV     #PAT11,R3
11107  043046  000440                   BR      70$
11108
11109  043050  122700  000012  63$:     CMPB    #12,R0          ;PATTERN 12 SET UP
11110  043054  001003                   BNE     64$
11111  043056  012703  047164           MOV     #PAT12,R3
11112  043062  000432                   BR      70$
11113
11114  043064  122700  000013  64$:     CMPB    #13,R0          ;PATTERN 13 SET UP
11115  043070  001003                   BNE     65$
```

```
11116  043072  012703  047224          MOV     #PAT13,R3
11117  043076  000424                  BR      70$
11118
11119  043100  122700  000014  65$:    CMPB    #14,R0          ;PATTERN 14 SET UP
11120  043104  001003                  BNE     66$
11121  043106  012703  047264          MOV     #PAT14,R3
11122  043112  000416                  BR      70$
11123
11124  043114  122700  000015  66$:    CMPB    #15,R0          ;PATTERN 15 SET UP
11125  043120  001003                  BNE     67$
11126  043122  012703  047324          MOV     #PAT15,R3
11127  043126  000410                  BR      70$
11128
11129  043130  122700  000016  67$:    CMPB    #16,R0          ;PATTERN 16 SET UP
11130  043134  001003                  BNE     68$
11131  043136  012703  047364          MOV     #PAT16,R3
11132  043142  000402                  BR      70$
11133
11134  043144  012703  047364  68$:    MOV     #PAT16,R3       ;SET UP FOR 16
11135
11136  043150  032700  004000  70$:    BIT     #BIT11,R0       ;FIRST WORD REPEAT?
11137  043154  001020                  BNE     73$             ;YES - SKIP
11138  043156  010446                  MOV     R4,-(SP)        ;STORE R4
11139  043160  010046                  MOV     R0,-(SP)        ;STORE R0
11140  043162  012700  000020          MOV     #16.,R0         ;PRESET COUNT FOR PATTERN LENGTH
11141  043166  010504                  MOV     R5,R4           ;STORE START OF BUFF
11142
11143  043170  012325          71$:    MOV     (R3)+,(R5)+     ;MOV WORD TO BUFF
11144  043172  005301                  DEC     R1              ;DEC WORD COUNT
11145  043174  001405                  BEQ     74$             ;EXIT IF ZERO
11146  043176  005300                  DEC     R0              ;DEC PAT LENGTH COUNT
11147  043200  001373                  BNE     71$             ;LOOP IF NOT ZERO
11148
11149  043202  012425          72$:    MOV     (R4)+,(R5)+     ;REPEAT PATTERN IN BUFFER
11150  043204  005301                  DEC     R1              ;DEC WORD COUNT
11151  043206  001375                  BNE     72$             ;LOOP UNTIL WORD COUNT ZERO
11152
11153  043210  012600          74$:    MOV     (SP)+,R0        ;RESTORE R0
11154  043212  012604                  MOV     (SP)+,R4        ;RESTORE R4
11155  043214  000403                  BR      22$             ;EXIT BUILD
11156
11157  043216  011325          73$:    MOV     (R3),(R5)+      ;MOV THE SAME WORD INTO BUFFER
11158  043220  005301                  DEC     R1              ;DEC WORD COUNT
11159  043222  001375                  BNE     73$             ;LOOP UNTIL ZERO
11160
11161  043224  032700  030000  22$:    BIT     #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
11162  043230  001402                  BEQ     34$             ;NO - SKIP
11163  043232  005337  177572          DEC     @#SR0           ;TURN OFF MEM MANAGEMENT
11164  043236  005700          34$:    TST     R0              ;IS COMPARE REQUIRED?
11165  043240  100012                  BPL     23$             ;NO - SKIP
11166  043242  013705  001700          MOV     DESHLD,R5       ;RESTORE COMPARE PARAMETERS
11167  043246  013704  001702          MOV     SRCHLD,R4
11168  043252  013703  001'04          MOV     WRDNUM,R3
11169  043256  013701  001'06          MOV     WRDCNT,R1
11170  043262  000137  042374          JMP     13$             ;GO START COMPARE
11171  043266                  23$:
```

```
11172  043266  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11173  043270  000137  042514                  JMP     16$             ;GO TO EXIT
11174
11175                                  ;;*****************************************************************
11176                                  .SBTTL  PHASE LOCK LOOP CLOCK ADJUSTMENT ROUNTINE
11177                                  ;*      THIS ROUTINE IS ENTERED VIA A START AT LOCATION 220(8). THE
11178                                  ;*      PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
11179                                  ;*      PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
11180                                  ;*      THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
11181                                  ;*      RESETS DIAGNOSTIC MODE BIT IN MR1. INSTRUCTIONS ON WHERE TO
11182                                  ;*      SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
11183                                  ;*
11184                                  ;*      THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
11185
11186  043274  104401  052650          ADJCLK: TYPE    ,OPR019         ;TYPE ADJUSTMENT INSTRUCTIONS
11187
11188  043300  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
11189  043306  013762  001626  000010          MOV     DRVNUM,RKCS2(R2)        ;SELECT DRIVE
11190  043314  012762  000001  000000          MOV     #1,RKCS1(R2)
11191  043322  032762  000200  000000  5$:     BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
11192  043330  001774                          BEQ     5$
11193  043332  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
11194  043340  001402                          BEQ     1$              ;NO - SKIP
11195  043342  005077  136342                  CLR     @KWLADD         ;CLEAR INTERRUPT ENABLE
11196
11197  043346  012762  000040  000026  1$:     MOV     #DMD,RKMR1(R2)  ;SET DIAG MODE
11198  043354  012701  000014                  MOV     #12.,R1         ;SET A COUNT
11199  043360  005301                  2$:     DEC     R1              ;DEC COUNT
11200  043362  001376                          BNE     2$              ;LOOP UNTIL ZERO
11201  043364  012762  000000  000026          MOV     #0,RKMR1(R2)    ;CLEAR MR1
11202  043372  012701  000014                  MOV     #12.,R1         ;SET COUNT
11203  043376  005301                  3$:     DEC     R1              ;DEC COUNT
11204  043400  001376                          BNE     3$              ;LOOP UNTIL ZERO
11205  043402  000761                          BR      1$              ;RESTART LOOP
11206
11207                                  .SBTTL  CONTROLLED HALT SUBROUTINE
11208                                  ;*      THIS ROUTINE IS ENTERED WHEN A CONTROL C IS TYPED. THE
11209                                  ;*      SUBSYSTEM IS CLEARED, THE DRIVE IS RECALIBRATED, AND, IF
11210                                  ;*      NECESSARY, CERTAIN CYLINDERS ARE REFORMATED. THE REFORMATTING
11211                                  ;*      IS CONTROLLED BY THE LOCATION REFMT WHICH CONTAINS THE ADDRESS
11212                                  ;*      OF THE CYLINDER TO BE REFORMATTED.
11213
11214  043404  012737  000112  001302  CTRHLT: MOV     #STN,STESTN             ;SET UP FOR HALT FAIL
11215
11216  043412  104416                          TSS.NIT                 ;CLEAR SUBSYSTEM
11217  043414  104003                          ERROR   3               ;BAD INIT ERROR
11218
11219  043416  113700  001102                  MOVB    STSTNM,R0       ;GET CURRENT TEST NUMBER
11220  043422  042700  177400                  BIC     #177400,R0      ;CLEAR UNUSED BITS
11221  043426  022700  000003                  CMP     #3,R0           ;TEST IF TEST NUMBER 3
11222  043432  001464                          BEQ     PROGEND         ;GO TO HALT PROG
11223  043434  004437  035530                  JSR     R4,LRLOAD       ;LOAD "L" REGS
11224  043440  000113                          RECAL                   ;RECAL
11225  043442  000000                          0                       ;0 WORDS
11226  043444  000000                          0                       ;0 IS BUFF ADDRESS
11227  043446     000                          .BYTE   0               ;SECTOR 0
```

```
11228  043447  000                        .BYTE   0               ;TRACK 0
11229  043450  000000                     0                       ;CYLINDER 0
11230
11231  043452  104417                     TLOADRK                 ;LOAD RK REGS
11232  043454  104423                     TWAT16                  ;WAIT FOR INTERRUPT
11233  043456  104002                     ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
11234
11235  043460  104421                     TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11236  043462  10400~                     ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11237
11238  043464  005037  001662             CLR     INTSET          ;CLEAR INTERRUPT FLAG
11239  043470  104437                     TWAT8S                  ;WAIT FOR SECOND INTERRUPT
11240  043472  104002                     ERROR   2
11241
11242  043474  104421                     TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11243  043476  104004                     ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11244
11245  043500  104416                     TSSINIT                 ;CLEAR SUBSYSTEM
11246  043502  104003                     ERROR   3               ;BAD INIT ERROR
11247
11248  043504  005737  001676             TST     REFMT           ;TEST IF REFORMAT REQUIRED
11249  043510  001435                     BEQ     PROGEND         ;NO - GO TO HALT
11250  043512  104401  053013             TYPE    ,OPR020         ;TYPE MESSAGE
11251
11252  043516  004437  035530             JSR     R4,LRLOAD       ;LOAD "L" REGS
11253  043522  000127                     WRHEAD                  ;WRHEAD
11254  043524  177676                     -102                    ;-102 WORDS
11255  043526  063414                     OBUFF                   ;OBUFF IS BUFF ADDRESS
11256  043530  000                        .BYTE   0               ;SECTOR 0
11257  043531  000                        .BYTE   0               ;TRACK 0
11258  043532  000312                     312                     ;CYLINDER 312
11259
11260  043534  005737  001676             TST     REFMT           ;TEST IF CYL 0
11261  043540  100002                     BPL     5$              ;NO - SKIP
11262  043542  005037  001614             CLR     L.DCYL          ;ELSE LOAD FOR CYL 0
11263  043546  004437  041740     5$:     JSR     R4,GENCOM       ;GENERATE HEADERS
11264  043552  001200                     1200
11265
11266  043554  104417                     TLOADRK                 ;LOAD RK REGS
11267  043556  104434                     TWAT159                 ;WAIT FOR INTERRUPT
11268  043560  104002                     ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
11269
11270  043562  104421                     TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11271  043564  104004                     ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11272
11273  043566  122737  000002  001607     CMPB    #2,L.DT         ;TEST IF TRACK 2 FORMATTED
11274  043574  001403                     BEQ     PROGEND         ;YES - SKIP
11275  043576  105237  001607             INCB    L.DT            ;ELSE BUMP TRACK
11276  043602  000761                     BR      5$              ;DO NEXT TRACK
11277
11278  043604  104401  053107     PROGEND:        TYPE    ,OPR021 ;TYPE HALT MESSAGE
11279  043610  012706  001100             MOV     #STACK,SP       ;CLEAR STACK
11280  043614  105037  001103             CLRB    $ERFLG          ;CLEAR ERROR FLAG
11281  043620  005037  001264             CLR     $ESCAPE         ;CLEAR ESCAPE
11282  043624  005737  000042             TST     @#42            ;TEST IF MONITOR PRESENT
11283  043630  001404                     BEQ     10$             ;NO - SKIP
```

```
11284  043632  005037  032016                  CLR     $EOPCT          ;SET FOR END OF PROGRAM
11285  043636  000137  031770                  JMP     $EOP            ;GO TO END OF PASS
11286
11287  043642  000000                  10$:    HALT                    ;HALT PROGRAM
11288  043644  000137  001754                  JMP     START1          ;GO TO RESTART IF CONTINUE
11289
11290                                  .SBTTL  HALT FAIL ROUTINE
11291                                  ;*      THIS ROUTINE IS ENTERED IF A HARDWARE ERROR IS DETECTED WHEN
11292                                  ;*      THE CARTRIDGE IS BEING REFORMATTED PRIOR TO HALT.
11293  043650  000240                  ABTFAIL:        NOP
11294  043652  104401  053150                  TYPE    ,OPR022         ;TYPE HALT FAIL MESSAGE
11295  043656  000137  043604                  JMP     PROGEND         ;GO STOP PROGRAM
11296                                  .SBTTL  TYPE ROUTINE
11297
11298                                  ;;*****************************************************************
11299                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11300                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11301                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11302                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11303                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11304                                  ;*
11305                                  ;*CALL:
11306                                  ;*1) USING A TRAP INSTRUCTION
11307                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11308                                  ;*OR
11309                                  ;*      TYPE
11310                                  ;*      MESADR
11311                                  ;*
11312
11313  043662  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
11314  043666  100002                  BPL     1$              ;;BR IF YES
11315  043670  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
11316  043672  000430                  BR      3$              ;;LEAVE
11317  043674  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
11318  043676  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
11319  043702  122737  000001  001316  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
11320  043710  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
11321  043712  132737  000100  001317  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
11322  043720  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
11323  043722  010037  043732          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
11324  043726  004737  033266          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
11325  043732  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
11326  043734  132737  000040  001317  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
11327  043742  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
11328  043744  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
11329  043746  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
11330  043750  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
11331  043752  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
11332  043754  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
11333  043760  000002                  RTI                     ;;RETURN
11334  043762  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
11335  043766  001430                  BEQ     8$
11336  043770  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
11337  043774  001006                  BNE     5$
11338  043776  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
11339  044000  104401                  TYPE                    ;;TYPE A CR AND LF
```

```
11340  044002  001273                          $CRLF
11341  044004  105037  044140                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
11342  044010  000755                          BR      2$              ;;GET NEXT CHARACTER
11343  044012  004737  044074          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
11344  044016  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
11345  044022  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
11346  044024  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
11347                                                                  ;;AND THE NULL CHAR.
11348  044030  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
11349  044034  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
11350  044036  004737  044074                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
11351  044042  105337  044140                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
11352  044046  000770                          BR      7$              ;;LOOP
11353
11354                                  ;HORIZONTAL TAB PROCESSOR
11355
11356  044050  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
11357  044054  004737  044074          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
11358  044060  132737  000007  044140          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
11359  044066  001372                          BNE     9$              ;;TAB STOP
11360  044070  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
11361  044072  000724                          BR      2$              ;;GET NEXT CHARACTER
11362  044074  105777  135050          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
11363  044100  100375                          BPL     $TYPEC
11364  044102  116677  000002  135042          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11365  044110  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
11366  044116  001003                          BNE     1$              ;;BRANCH IF NO
11367  044120  105037  044140                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
11368  044124  000406                          BR      $TYPEX          ;;EXIT
11369  044126  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
11370  044134  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
11371  044136  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
11372  044140  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
11373  044142  000207          $TYPEX: RTS     PC
11374
11375                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11376
11377                                  ;;******************************************************************
11378                                  ;*THIS ROUTINE IS USED TO CHANGE A 16 BIT BINARY NUMBER TO A 5-DIGIT
11379                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
11380                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
11381                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
11382                                  ;*REPLACED WITH SPACES.
11383                                  ;*CALL:
11384                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
11385                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
11386
11387  044144                          $TYPDS:
11388  044144  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11389  044146  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11390  044150  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11391  044152  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11392  044154  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11393  044156  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
11394  044162  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
11395  044166  100004                          BPL     1$              ;;BR IF INPUT IS POS.
```

```
11396  044170  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
11397  044172  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
11398  044200  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
11399  044202  012703  044360                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
11400  044206  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
11401  044212  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
11402  044214  016001  044350                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
11403  044220  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
11404  044222  002402                          BLT     4$              ;;BR IF DONE
11405  044224  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
11406  044226  000774                          BR      3$
11407  044230  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
11408  044232  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
11409  044234  001002                          BNE     5$              ;;FALL THROUGH IF 0
11410  044236  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
11411  044240  100407                          BMI     7$              ;;BR IF YES
11412  044242  106316                  5$:     ASLB    (SP)            ;;MSD?
11413  044244  103003                          BCC     6$              ;;BR IF NO
11414  044246  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
11415  044254  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
11416  044260  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
11417  044264  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
11418  044266  005720                          TST     (R0)+           ;;JUST INCREMENTING
11419  044270  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
11420  044274  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
11421  044276  003002                          BGT     8$              ;;GO TO EXIT
11422  044300  010502                          MOV     R5,R2           ;;GET THE LSD
11423  044302  000764                          BR      6$              ;;GO CHANGE TO ASCII
11424  044304  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
11425  044306  100003                          BPL     9$              ;;BR IF NO
11426  044310  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
11427  044316  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
11428  044320  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
11429  044322  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11430  044324  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11431  044326  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11432  044330  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11433  044332  104401  044360                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
11434  044336  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
11435  044344  012616                          MOV     (SP)+,(SP)
11436  044346  000002                          RTI                     ;;RETURN TO USER
11437  044350  023420                  $DTBL:  10000.
11438  044352  001750                          1000.
11439  044354  000144                          100.
11440  044356  000012                          10.
11441  044360  000004                  $DBLK:  .BLKW   4
11442                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
11443
11444                                  ;;*****************************************************************
11445                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
11446                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
11447                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
11448                                  ;*CALL:
11449                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11450                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
11451                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
```

```
11452                                   ;*      .BYTE   M               ;;M=1 OR 0
11453                                   ;*                              ;;1=TYPE LEADING ZEROS
11454                                   ;*                              ;;0=SUPPRESS LEADING ZEROS
11455                                   ;*
11456                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
11457                                   ;*$TYPOS OR $TYPOC
11458                                   ;*CALL:
11459                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11460                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
11461                                   ;*
11462                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
11463                                   ;*CALL:
11464                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11465                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
11466
11467  044370  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
11468  044374  116637  000001  044613          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
11469  044402  112637  044615                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
11470  044406  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
11471  044412  000406                          BR      $TYPON
11472  044414  112737  000001  044613  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
11473  044422  112737  000006  044615          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
11474  044430  112737  000005  044612  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
11475  044436  010346                          MOV     R3,-(SP)        ;;SAVE R3
11476  044440  010446                          MOV     R4,-(SP)        ;;SAVE R4
11477  044442  010546                          MOV     R5,-(SP)        ;;SAVE R5
11478  044444  113704  044615                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
11479  044450  005404                          NEG     R4
11480  044452  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
11481  044456  110437  044614                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
11482  044462  113704  044613                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
11483  044466  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
11484  044472  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
11485  044474  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
11486  044476  000404                          BR      3$              ;;GO DO MSB
11487  044500  006105                  2$:     ROL     R5              ;;FORM THIS JIGIT
11488  044502  006105                          ROL     R5
11489  044504  006105                          ROL     R5
11490  044506  010503                          MOV     R5,R3
11491  044510  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
11492  044512  105337  044614                  DECB    $OMODE          ;;TYPE THIS DIGIT?
11493  044516  100016                          BPL     7$              ;;BR IF NO
11494  044520  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
11495  044524  001002                          BNE     4$              ;;TEST FOR 0
11496  044526  005704                          TST     R4              ;;SUPPRESS THIS 0?
11497  044530  001403                          BEQ     5$              ;;BR IF YES
11498  044532  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11499  044534  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11500  044540  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
11501  044544  110337  044610                  MOVB    R3,8$           ;;SAVE FOR TYPING
11502  044550  104401  044610                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
11503  044554  105337  044612  7$:     DECB    $OCNT           ;;COUNT BY 1
11504  044560  003347                          BGT     2$              ;;BR IF MORE TO DO
11505  044562  002402                          BLT     6$              ;;BR IF DONE
11506  044564  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11507  044566  000744                          BR      2$              ;;GO DO THE LAST DIGIT
```

```
11508  044570  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
11509  044572  012604                          MOV     (SP)+,R4        ;;RESTORE R4
11510  044574  012603                          MOV     (SP)+,R3        ;;RESTORE R3
11511  044576  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
11512  044604  012616                          MOV     (SP)+,(SP)
11513  044606  000002                          RTI                     ;;RETURN
11514  044610     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11515  044611     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11516  044612     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
11517  044613     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
11518  044614  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
11519                                  .SBTTL  TTY INPUT ROUTINE
11520
11521                                  ;;**************************************************
11522                                  .ENABL  LSB
11523  044616  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
11524  044620  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
11525  044622  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
11526  044624  000001                  $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
11527          044625                  $TKQEND=.
11528          044626                  .EVEN
11529
11530                                  ;*TK INITIALIZE ROUTINE
11531                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
11532                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
11533                                  ;
11534                                  ;*CALL:
11535                                  ;*      JSR     PC,$TKINT
11536                                  ;*      RETURN
11537                                  ;
11538  044626  005037  044616          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
11539  044632  012737  044624  044620          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
11540  044640  013737  044620  044622          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
11541  044646  012737  044676  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
11542  044654  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
11543  044662  005777  134260                  TST     @$TKB           ;;CLEAR DONE FLAG
11544  044666  012777  000100  134250          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
11545  044674  000207                          RTS     PC              ;;RETURN TO CALLER
11546
11547                                  ;*TK SERVICE ROUTINE
11548                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
11549                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
11550                                  ;*IT IN THE QUEUE.
11551                                  ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
11552                                  ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)
11553
11554  044676  117746  134244          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
11555  044702  042716  177600                  BIC     #↑C177,(SP)     ;;STRIP THE JUNK
11556  044706  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
11557  044712  001007                          BNE     1$              ;;BRANCH IF NO
11558  044714  104401  046012                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
11559  044720  004737  044626                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
11560  044724  005726                          TST     (SP)+           ;;CLEAN UP STACK
11561  044726  000137  043404                  JMP     CTRHLT          ;;CONTROL C RESTART
11562  044732  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
11563  044736  001004                          BNE     2$              ;;BRANCH IF NO
```

```
11564  044740  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
11565  044746  001500                          BEQ     6$              ;;GO TO SWR CHANGE
11566
11567  044750                          2$:
11568  044750  022737  000001  044616          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
11569  044756  001004                          BNE     3$              ;;BRANCH IF NO
11570  044760  104401  001266                  TYPE    ,$BELL          ;;RING THE TTY BELL
11571  044764  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
11572  044766  000451                          BR      5$              ;;EXIT
11573  044770  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
11574  044774  001021                          BNE     32$             ;;BRANCH IF NO
11575  044776  005077  134142                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11576  045002  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
11577  045004  105777  134134          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
11578  045010  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
11579  045012  117746  134130                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
11580  045016  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11581  045022  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
11582  045026  001366                          BNE     31$             ;;BRANCH IF NO
11583  045030  012777  000100  134106          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
11584  045036  000002                          RTI                     ;;RETURN
11585  045040  005237  044616          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
11586  045044  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
11587  045050  002405                          BLT     4$              ;;BRANCH IF YES
11588  045052  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
11589  045056  003002                          BGT     4$              ;;BRANCH IF YES
11590  045060  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
11591  045064  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
11592  045070  005237  044620                  INC     $TKQIN          ;;UPDATE THE POINTER
11593  045074  023727  044620  044625          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
11594  045102  001003                          BNE     5$              ;;BRANCH IF NO
11595  045104  012737  044624  044620          MOV     #$TKQSR,$TKQIN  ;;RESET THE POINTER
11596  045112  000002                  5$:     RTI                     ;;RETURN
11597
11598                                  ;;**********************************************************************
11599                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11600                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
11601                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11602                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11603  045114  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
11604  045122  001124                          BNE     15$             ;;EXIT IF NOT
11605  045124  105777  134014                  TSTB    @$TKS           ;;IS A CHAR WAITING?
11606  045130  100121                          BPL     15$             ;;IF NOT, EXIT
11607  045132  117746  134010                  MOVB    @$TKB,-(SP)     ;;YES
11608  045136  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
11609  045142  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
11610  045146  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
11611                                                                  ;;AND EXIT
11612
11613                                  ;;**********************************************************************
11614                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11615                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11616                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11617  045150  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
11618  045156  001674                          BEQ     2$              ;;BRANCH IF YES
11619  045160  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
```

```
11620  045162  004737  044626           JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
11621  045166  005077  133752           CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11622  045172  112737  000001  001135   MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
11623
11624  045200  104401  046024           TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (↑G)
11625  045204  104401  046031  $GTSWR:  TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
11626  045210  013746  000176           MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
11627  045214  104402                   TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11628  045216  104401  046042           TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
11629  045222  005046           19$:    CLR     -(SP)           ;;CLEAR COUNTER
11630  045224  005046                   CLR     -(SP)           ;;THE NEW SWR
11631  045226  105777  133712   7$:     TSTB    @$TKS           ;;CHAR THERE?
11632  045232  100375                   BPL     7$              ;;IF NOT TRY AGAIN
11633
11634  045234  117746  133706           MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
11635  045240  042716  177600           BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
11636
11637  045244  021627  000003           CMP     (SP),#3         ;;IS IT A CONTROL-C?
11638  045250  001015                   BNE     9$              ;;BRANCH IF NOT
11639  045252  104401  046012           TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (↑C)
11640  045256  062706  000006           ADD     #6,SP           ;;CLEAN UP STACK
11641  045262  123727  001135  000001   CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
11642  045270  001003                   BNE     8$              ;;BRANCH IF NO
11643  045272  012777  000100  133644   MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
11644  045300  000137  043404   8$:     JMP     CTRHLT          ;;CONTROL-C RESTART
11645
11646
11647  045304  021627  000025   9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
11648  045310  001005                   BNE     10$             ;;BRANCH IF NOT
11649  045312  104401  046017           TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (↑U)
11650  045316  062706  000006   20$:     ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
11651  045322  000737                   BR      19$             ;;LET'S TRY IT AGAIN
11652
11653
11654  045324  021627  000015   10$:     CMP     (SP),#15        ;;IS IT A <CR>?
11655  045330  001022                   BNE     16$             ;;BRANCH IF NO
11656  045332  005766  000004           TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
11657  045336  001403                   BEQ     11$             ;;BRANCH IF YES
11658  045340  016677  000002  133572   MOV     2(SP),@SWR      ;;SAVE NEW SWR
11659  045346  062706  000006   11$:     ADD     #6,SP           ;;CLEAR UP STACK
11660  045352  104401  001273   14$:     TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
11661  045356  123727  001135  000001   CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
11662  045364  001003                   BNE     15$             ;;BRANCH IF NOT
11663  045366  012777  000100  133550   MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
11664  045374  000002           15$:     RTI                     ;;RETURN
11665  045376  004737  044074   16$:     JSR     PC,$TYPEC       ;;ECHO CHAR
11666  045402  021627  000060           CMP     (SP),#60        ;;CHAR < 0?
11667  045406  002420                   BLT     18$             ;;BRANCH IF YES
11668  045410  021627  000067           CMP     (SP),#67        ;;CHAR > 7?
11669  045414  003015                   BGT     18$             ;;BRANCH IF YES
11670  045416  042726  000060           BIC     #60,(SP)+       ;;STRIP-OFF ASCII
11671  045422  005766  000002           TST     2(SP)           ;;IS THIS THE FIRST CHAR
11672  045426  001403                   BEQ     17$             ;;BRANCH IF YES
11673  045430  006316                   ASL     (SP)            ;;NO, SHIFT PRESENT
11674  045432  006316                   ASL     (SP)            ;;   CHAR OVER TO MAKE
11675  045434  006316                   ASL     (SP)            ;;   ROOM FOR NEW ONE.
```

```
11676  045436  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
11677  045442  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
11678  045446  000667                          BR      7$              ;;GET THE NEXT ONE
11679  045450  104401  001272          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
11680  045454  000720                          BR      20$             ;;SIMULATE CONTROL-U
11681                                  .DSABL  LSB
11682
11683
11684                                  ;;**************************************************************
11685                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11686                                  ;*CALL:
11687                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
11688                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
11689                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
11690                                  ;
11691
11692  045456  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
11693  045460  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
11694  045466  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
11695  045472  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
11696  045474  012746  045502                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
11697  045500  000002                          RTI                     ;;POP NEW PC AND PS
11698  045502                          64$:
11699  045502  005737  044616          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
11700  045506  001775                          BEQ     1$
11701  045510  005337  044616                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
11702  045514  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
11703  045522  005237  044622                  INC     $TKQOUT         ;;UPDATE THE POINTER
11704  045526  023727  044622  044625          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
11705  045534  001003                          BNE     2$              ;;BRANCH IF NO
11706  045536  012737  044624  044622          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
11707  045544  000002          2$:     RTI                             ;;RETURN
11708                                  ;;**************************************************************
11709                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
11710                                  ;*CALL:
11711                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
11712                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11713                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
11714
11715  045546  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
11716  045550  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
11717  045552  012703  046002          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
11718  045556  022703  046012          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
11719  045562  101456                          BLOS    4$              ;;BR IF YES
11720  045564  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
11721  045566  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
11722  045570  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
11723  045574  001022                          BNE     5$              ;;BR IF NO
11724  045576  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
11725  045600  001007                          BNE     6$              ;;BR IF NO
11726  045602  112737  000134  046000          MOVB    #'\,9$          ;;TYPE A BACK SLASH
11727  045610  104401  046000                  TYPE    ,9$
11728  045614  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
11729  045620  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
11730  045622  020327  046002                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
11731  045626  103434                          BLO     4$              ;;BR IF YES
```

```
11732  045630  111337  046000          MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
11733  045634  104401  046000          TYPE    ,9$             ;;GO TYPE
11734  045640  000746                  BR      2$              ;;GO READ ANOTHER CHAR.
11735  045642  005716          5$:     TST     (SP)            ;;RUBOUT KEY SET?
11736  045644  001406                  BEQ     7$              ;;BR IF NO
11737  045646  112737  000134  046000  MOVB    #'\,9$          ;;TYPE A BACK SLASH
11738  045654  104401  046000          TYPE    ,9$
11739  045660  005016                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
11740  045662  122713  000025  7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
11741  045666  001003                  BNE     8$              ;;BR IF NO
11742  045670  104401  046017          TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
11743  045674  000726                  BR      1$              ;;GO START OVER
11744  045676  122713  000022  8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
11745  045702  001011                  BNE     3$              ;;BRANCH IF NO
11746  045704  105013                  CLRB    (R3)            ;;CLEAR THE CHARACTER
11747  045706  104401  001273          TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
11748  045712  104401  046002          TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
11749  045716  000717                  BR      2$              ;;GO PICKUP ANOTHER CHACTER
11750  045720  104401  001272  4$:     TYPE    ,$QUES          ;;TYPE A '?'
11751  045724  000712                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
11752  045726  111337  046000  3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
11753  045732  104401  046000          TYPE    ,9$
11754  045736  122723  000015          CMPB    #15,(R3)+       ;;CHECK FOR RETURN
11755  045742  001305                  BNE     2$              ;;LOOP IF NOT RETURN
11756  045744  105063  177777          CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
11757  045750  104401  001274          TYPE    ,$LF            ;;TYPE A LINE FEED
11758  045754  005726                  TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
11759  045756  012603                  MOV     (SP)+,R3        ;;RESTORE R3
11760  045760  011646                  MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
11761  045762  016666  000004  000002  MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
11762  045770  012766  046002  000004  MOV     #$TTYIN,4(SP)
11763  045776  000002                  RTI                     ;;RETURN
11764  046000     000          9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
11765  046001     000                  .BYTE   0               ;;TERMINATOR
11766  046002  000010          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
11767  046012  041536  005015     000  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
11768  046017     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
11769  046024  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
11770  046031     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
11771  046036  036440  000040
11772  046042  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
11773  046050  020075     000
11774          046054                  .EVEN
11775                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
11776
11777                                  ;;*****************************************************************
11778                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
11779                                  ;*CHANGE IT TO BINARY.
11780                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
11781                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11782                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11783                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11784                                  ;*CALL:
11785                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
11786                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
11787                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
```

```
11788
11789  046054  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
11790  046056  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
11791  046064  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11792  046066  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11793  046070  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11794  046072  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
11795  046074  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
11796  046076  010037  046202                  MOV     R0,5$           ;;AND SAVE IT
11797  046102  005001                          CLR     R1              ;;CLEAR DATA WORD
11798  046104  005002                          CLR     R2
11799  046106  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
11800  046110  001420                          BEQ     3$              ;;IF ZERO GET OUT
11801  046112  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
11802  046116  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
11803  046120  122716  000067                  CMPB    #'7,(SP)
11804  046124  002423                          BLT     4$
11805  046126  006301                          ASL     R1              ;;*2
11806  046130  006102                          ROL     R2
11807  046132  006301                          ASL     R1              ;;*4
11808  046134  006102                          ROL     R2
11809  046136  006301                          ASL     R1              ;;*8
11810  046140  006102                          ROL     R2
11811  046142  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
11812  046146  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
11813  046150  000756                          BR      2$              ;;LOOP
11814  046152  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
11815  046154  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
11816  046160  010237  046212                  MOV     R2,$HIOCT
11817  046164  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11818  046166  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11819  046170  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11820  046172  000002                          RTI                     ;;RETURN
11821  046174  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
11822  046176  105010                          CLRB    (R0)            ;;SET A TERMINATOR
11823  046200  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
11824  046202  000000                  5$:     .WORD   0
11825  046204  104401  001272                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
11826  046210  000730                          BR      1$              ;;TRY AGAIN
11827  046212  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
11828                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
11829
11830                                  ;;******************************************************************
11831                                  ;*SAVE R0-R5
11832                                  ;*CALL:
11833                                  ;*      SAVREG
11834                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
11835                                  ;*
11836                                  ;*TOP---(+16)
11837                                  ;* +2---(+18)
11838                                  ;* +4---R5
11839                                  ;* +6---R4
11840                                  ;* +8---R3
11841                                  ;*+10---R2
11842                                  ;*+12---R1
11843                                  ;*+14---R0
```

```
 11844
 11845  046214                                 $SAVREG:
 11846  046214  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 11847  046216  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 11848  046220  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 11849  046222  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 11850  046224  010446                                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 11851  046226  010546                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 11852  046230  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 11853  046234  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 11854  046240  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 11855  046244  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 11856  046250  000002                                 RTI
 11857
 11858                                         ;*RESTORE R0-R5
 11859                                         ;*CALL:
 11860                                         ;*      RESREG
 11861  046252                                 $RESREG:
 11862  046252  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 11863  046256  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 11864  046262  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 11865  046266  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 11866  046272  012605                                 MOV     (SP)+,R5        ;;POP STACK INTO R5
 11867  046274  012604                                 MOV     (SP)+,R4        ;;POP STACK INTO R4
 11868  046276  012603                                 MOV     (SP)+,R3        ;;POP STACK INTO R3
 11869  046300  012602                                 MOV     (SP)+,R2        ;;POP STACK INTO R2
 11870  046302  012601                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
 11871  046304  012600                                 MOV     (SP)+,R0        ;;POP STACK INTO R0
 11872  046306  000002                                 RTI
 11873                                         .SBTTL  POWER DOWN AND UP ROUTINES
 11874
 11875                                         ;;**************************************************************
 11876                                         ;POWER DOWN ROUTINE
 11877  046310  012737  046450  000024         $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 11878  046316  012737  000340  000026                 MOV     #340,@#PWRVEC+2 ;;PRIO:7
 11879  046324  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 11880  046326  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 11881  046330  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 11882  046332  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 11883  046334  010446                                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 11884  046336  010546                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 11885  046340  017746  132574                         MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
 11886  046344  010637  046454                         MOV     SP,$SAVR6       ;;SAVE SP
 11887  046350  012737  046362  000024                 MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 11888  046356  000000                                 HALT
 11889  046360  000776                                 BR      .-2             ;;HANG UP
 11890
 11891                                         ;;**************************************************************
 11892                                         ;POWER UP ROUTINE
 11893  046362  012737  046450  000024         $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 11894  046370  013706  046454                         MOV     $SAVR6,SP       ;;GET SP
 11895  046374  005037  046454                         CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 11896  046400  005237  046454         1$:             INC     $SAVR6          ;;WAIT FOR THE INC
 11897  046404  001375                                 BNE     1$              ;;OF  WORD
 11898  046406  012677  132526                         MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
 11899  046412  012605                                 MOV     (SP)+,R5        ;;POP STACK INTO R5
```

```
11900  046414  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11901  046416  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11902  046420  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11903  046422  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11904  046424  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11905  046426  012737  046310  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
11906  046434  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
11907  046442  104401                          TYPE                    ;REPORT THE POWER FAILURE
11908  046444  046456                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
11909  046446  000002                          RTI
11910  046450  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
11911  046452  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
11912  046454  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
11913  046456  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
11914  046464  000122
11915                                          .EVEN
11916                                  .SBTTL  TRAP DECODER
11917
11918                                  ;;**************************************************************
11919                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
11920                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
11921                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
11922                                  ;*GO TO THAT ROUTINE.
11923
11924  046466  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
11925  046470  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
11926  046474  005740                          TST     -(R0)           ;;BACKUP BY 2
11927  046476  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
11928  046500  006300                          ASL     R0              ;;POSITION FOR INDEXING
11929  046502  016000  046522                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
11930  046506  000200                          RTS     R0              ;;GO TO ROUTINE
11931
11932
11933                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
11934
11935  046510  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
11936  046512  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
11937  046520  000002                          RTI                     ;;RESTORE THE PSW
11938
11939                                  .SBTTL  TRAP TABLE
11940
11941                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
11942                                  ;*BY THE "TRAP" INSTRUCTION.
11943
11944                                  ;       ROUTINE
11945                                  ;       -------
11946  046522  046510                  $TRPAD: .WORD   $TRAP2
11947  046524  043662                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
11948  046526  044414                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
11949  046530  044370                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
11950  046532  044430                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
11951  046534  044144                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
11952
11953  046536  045204                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
11954
11955  046540  045114                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
```

```
 11956  046542  045456                $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 11957  046544  045546                $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 11958  046546  046054                $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 11959  046550  046214                $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
 11960  046552  046252                $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
 11961  046554  035056                SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
 11962  046556  036656                SSINIT  ;;CALL=TSSINIT  TRAP+16(104416) INITIALIZE SUBSYSTEM
 11963  046560  035570                LOADRK  ;;CALL=TLOADRK  TRAP+17(104417) LOAD RK611 FOR OPERATION
 11964  046562  035652                GETRK   ;;CALL=TGETRK   TRAP+20(104420) GET RK611 REGISTERS
 11965  046564  037642                CHKOP   ;;CALL=TCHKOP   TRAP+21(104421) CHECK OPREATION FOR ANY ERRORS
 11966  046566  037610                CHKWE   ;;CALL=TCHKWE   TRAP+22(104422) CHECK OPERATION FOR EXPECTED ERRORS
 11967  046570  035364                IWAT16  ;;CALL=TWAT16   TRAP+23(104423) WAIT 16 MS
 11968  046572  035354                IWAT32  ;;CALL=TWAT32   TRAP+24(104424) WAIT 32 MS
 11969  046574  035344                IWAT48  ;;CALL=TWAT48   TRAP+25(104425) WAIT 48 MS
 11970  046576  035334                IWAT64  ;;CALL=TWAT64   TRAP+26(104426) WAIT 64 MS
 11971  046600  035324                IWAT80  ;;CALL=TWAT80   TRAP+27(104427) WAIT 80 MS
 11972  046602  035314                IWAT96  ;;CALL=TWAT96   TRAP+30(104430) WAIT 96 MS
 11973  046604  035304                IWAT112 ;;CALL=TWAT112  TRAP+31(104431) WAIT 112 MS
 11974  046606  035274                IWAT128 ;;CALL=TWAT128  TRAP+32(104432) WAIT 128 MS
 11975  046610  035264                IWAT144 ;;CALL=TWAT144  TRAP+33(104433) WAIT 144 MS
 11976  046612  035254                IWAT159 ;;CALL=TWAT159  TRAP+34(104434) WAIT 160 MS
 11977  046614  035244                IWAT1S  ;;CALL=TWAT1S   TRAP+35(104435) WAIT FOR 1 SECOND
 11978  046616  035234                IWAT2S  ;;CALL=TWAT2S   TRAP+36(104436) WAIT FOR 2 SECONDS
 11979  046620  035214                IWAT8S  ;;CALL=TWAT8S   TRAP+37(104437) WAIT FOR 8 SECONDS
 11980  046622  035224                IWATIM  ;;CALL=TWATIM   TRAP+40(104440) WAIT FOR 1 MIN
 11981          000102        $TERM=.-$TRPAD
```

```
 11982                                  .SBTTL  DATA PATTERNS
 11983                                  ;DATA PATTERN 1
 11984                                  ;       PATTERN IS ALL ZEROS
 11985                                  ;
 11986                                  ;DATA PATTERN 2
 11987                                  ;       HI-LO FREQ. MIX
 11988  046624                          PAT02:
 11989  046624  177777                         177777
 11990  046626  177777                         177777
 11991  046630  177777                         177777
 11992  046632  052525                         052525
 11993  046634  052525                         052525
 11994  046636  052525                         052525
 11995  046640  177777                         177777
 11996  046642  177777                         177777
 11997  046644  052525                         052525
 11998  046646  052525                         052525
 11999  046650  177777                         177777
 12000  046652  052525                         052525
 12001  046654  177252                         177252
 12002  046656  177252                         177252
 12003  046660  172765                         172765
 12004  046662  172765                         172765
 12005                                  ;
 12006                                  ;DATA PATERN 3
 12007                                  ;       HI FREQ. PHASE MIX
 12008  046664                          PAT03:
 12009  046664  000000                         000000
 12010  046666  000000                         000000
 12011  046670  000000                         000000
 12012  046672  177777                         177777
 12013  046674  177777                         177777
 12014  046676  177777                         177777
 12015  046700  000000                         000000
 12016  046702  000000                         000000
 12017  046704  177777                         177777
 12018  046706  177777                         177777
 12019  046710  000000                         000000
 12020  046712  177777                         177777
 12021  046714  000000                         000000
 12022  046716  177777                         177777
 12023  046720  000000                         000000
 12024  046722  177777                         177777
 12025                                  ;
 12026                                  ;DATA PATTERN 4
 12027                                  ;       LO FREQ. PHASE MIX
 12028  046724                          PAT04:
 12029  046724  052525                         052525
 12030  046726  052525                         052525
 12031  046730  052525                         052525
 12032  046732  125252                         125252
 12033  046734  125252                         125252
 12034  046736  125252                         125252
 12035  046740  052525                         052525
 12036  046742  052525                         052525
 12037  046744  125252                         125252
```

```
12038  046746  125252                    125252
12039  046750  052525                    052525
12040  046752  125252                    125252
12041  046754  052525                    052525
12042  046756  125252                    125252
12043  046760  052525                    052525
12044  046762  125252                    125252
12045
12046                                    ;DATA PATTERN 5
12047                                    ;       MAX PRECOMP. PHASE MIX
12048  046764                            PAT05:
12049  046764  133333                    133333
12050  046766  066666                    066666
12051  046770  155555                    155555
12052  046772  155555                    155555
12053  046774  133333                    133333
12054  046776  066666                    066666
12055  047000  066666                    066666
12056  047002  155555                    155555
12057  047004  155555                    155555
12058  047006  133333                    133333
12059  047010  133333                    133333
12060  047012  133333                    133333
12061  047014  133333                    133333
12062  047016  133333                    133333
12063  047020  133333                    133333
12064  047022  133333                    133333
12065
12066                                    ;DATA PATTERN 6
12067                                    ;       ROTATING BOUNDARY PULSE PRECOMP.
12068  047024                            PAT06:
12069  047024  121105                    121105
12070  047026  150442                    150442
12071  047030  064221                    064221
12072  047032  132110                    132110
12073  047034  055044                    055044
12074  047036  026422                    026422
12075  047040  013211                    013211
12076  047042  105504                    105504
12077  047044  042642                    042642
12078  047046  021321                    021321
12079  047050  110550                    110550
12080  047052  044264                    044264
12081  047054  022132                    022132
12082  047056  011055                    011055
12083  047060  104426                    104426
12084  047062  042213                    042213
12085
12086                                    ;DATA PATTERN 7
12087                                    ;       FIELD OF ALL ONES
12088
12089                                    ;DATA PATTERN 10
12090                                    ;       ROTATING CELL PULSE PRECOMP.
12091  047064                            PAT10:
12092  047064  026455                    026455
12093  047066  113226                    113226
```

```
12094  047070  045513                    045513
12095  047072  122645                    122645
12096  047074  151322                    151322
12097  047076  064551                    064551
12098  047100  132264                    132264
12099  047102  055132                    055132
12100  047104  026455                    026455
12101  047106  113226                    113226
12102  047110  045513                    045513
12103  047112  122645                    122645
12104  047114  151322                    151322
12105  047116  064551                    064551
12106  047120  132264                    132264
12107  047122  055132                    055132
12108
12109                                  ;DATA PATTERN 11
12110                                  ;       SHIFTED 1 IN A FIELD OF ZEROS
12111  047124                          PAT11:
12112  047124  000001                    000001
12113  047126  000002                    000002
12114  047130  000004                    000004
12115  047132  000010                    000010
12116  047134  000020                    000020
12117  047136  000040                    000040
12118  047140  000100                    000100
12119  047142  000200                    000200
12120  047144  000400                    000400
12121  047146  001000                    001000
12122  047150  002000                    002000
12123  047152  004000                    004000
12124  047154  010000                    010000
12125  047156  020000                    020000
12126  047160  040000                    040000
12127  047162  100000                    100000
12128
12129                                  ;DATA PATTERN 12
12130                                  ;       SHIFTED 0 IN A FIELD OF ONES
12131  047164                          PAT12:
12132  047164  177776                    177776
12133  047166  177775                    177775
12134  047170  177773                    177773
12135  047172  177767                    177767
12136  047174  177757                    177757
12137  047176  177737                    177737
12138  047200  177677                    177677
12139  047202  177577                    177577
12140  047204  177377                    177377
12141  047206  176777                    176777
12142  047210  175777                    175777
12143  047212  173777                    173777
12144  047214  167777                    167777
12145  047216  157777                    157777
12146  047220  137777                    137777
12147  047222  077777                    077777
12148
12149                                  ;DATA PATTERN 13
```

```
12150
12151  047224
12152  047224  052525
12153  047226  052525
12154  047230  052525
12155  047232  052525
12156  047234  052525
12157  047236  052525
12158  047240  052525
12159  047242  052525
12160  047244  052525
12161  047246  052525
12162  047250  052525
12163  047252  052525
12164  047254  052525
12165  047256  052525
12166  047260  052525
12167  047262  052525
12168
12169
12170
12171  047264
12172  047264  125252
12173  047266  125252
12174  047270  125252
12175  047272  125252
12176  047274  125252
12177  047276  125252
12178  047300  125252
12179  047302  125252
12180  047304  125252
12181  047306  125252
12182  047310  125252
12183  047312  125252
12184  047314  125252
12185  047316  125252
12186  047320  125252
12187  047322  125252
12188
12189
12190
12191  047324
12192  047324  000001
12193  047326  000003
12194  047330  000007
12195  047332  000017
12196  047334  000037
12197  047336  000077
12198  047340  000177
12199  047342  000377
12200  047344  000777
12201  047346  001777
12202  047350  003777
12203  047352  007777
12204  047354  017777
12205  047356  037777
```

```
;       ALTERNATING 0-1
PAT13:
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525
        052525

;DATA PATTERN 14
;       ALTERNATING 1-0
PAT14:
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252
        125252

;DATA PATTERN 15
;       SHIFTING ZEROS AND ONES
PAT15:
        000001
        000003
        000007
        000017
        000037
        000077
        000177
        000377
        000777
        001777
        003777
        007777
        017777
        037777
```

```
 12206  047360  077777                          077777
 12207  047362  177777                          177777
 12208
 12209                                          ;DATA PATTERN 16
 12210                                          ;       COMPOSITE ROTATING
 12211  047364                                  PAT16:
 12212  047364  072307                          072307
 12213  047366  135143                          135143
 12214  047370  156461                          156461
 12215  047372  167230                          167230
 12216  047374  073514                          073514
 12217  047376  035646                          035646
 12218  047400  016723                          016723
 12219  047402  107351                          107351
 12220  047404  143564                          143564
 12221  047406  061672                          061672
 12222  047410  030735                          030735
 12223  047412  114356                          114356
 12224  047414  046167                          046167
 12225  047416  123073                          123073
 12226  047420  151453                          151453
 12227  047422  164616                          164616
 12228
```

```
12229                                   .SBTTL  FIELDS AND VARIABLES FOR OPERATION CHECKING
12230           024000                  CS1ERBIT=24000                  ;CS1 ERROR BITS SPAR & CTO
12231           177400                  CS2ERBIT=177400                 ;CS2 ERROR BITS
12232                                                                   ;DLT,WCE,UPE,NED,NEM
12233                                                                   ;PGE,MOS,UFE
12234           000070                  DSERBIT=70                      ;DS ERROR BITS
12235                                                                   ;SPDLSS,DROT,ACLO
12236
12237   047424  000000                  EXPWC:  .WORD   0               ;EXPECTED WORD COUNT (GIVEN)
12238   047426  000000                  EXPUBA: .WORD   0               ;EXPECTED UPPER BA (COMPUTED)
12239   047430  000000                  EXPBA:  .WORD   0               ;EXPECTED BUS ADDRESS (COMPUTED)
12240   047432  000000                  EXPCYL: .WORD   0               ;EXPECTED CYLINDER (COMPUTED)
12241   047434  000000                  EXPSEC: .WORD   0               ;EXPECTED SECTOR (COMPUTED)
12242   047436  000000                  EXPTRK: .WORD   0               ;EXPECTED TRACK (COMPUTED)
12243
12244   047440  000000                  REALWC: .WORD   0               ;WORD COUNT AT END OF OPERATION
12245   047442  000000                  REALUB: .WORD   0               ;REAL UPPER BA
12246   047444  000000                  REALBA: .WORD   0               ;BUS ADDRESS
12247   047446  000000                  REALCY: .WORD   0               ;CYLINDER
12248   047450  000000                  REALTRK:.WORD   0               ;TRACK
12249   047452  000000                  REALSEC:.WORD   0               ;SECTOR
12250
12251   047454  000000                  GRP1ER: .WORD   0               ;GROUP 1 ERROR FIELDS
12252           000001                          SPARERR=BIT0            ;CONTROLLER DETECTED DRIVE BUS PARITY ERR
12253           000002                          SKIERR= BIT1            ;SEEK INCOMPLETE
12254           000004                          NXFERR= BIT2            ;NON-EXECUTABLE DRIVE FUNCTION
12255           000010                          DRPARERR=BIT3           ;DRIVE DETECTED DRIVE BUS PARITY ERROR
12256           000020                          FMTERR= BIT4            ;FORMAT ERROR
12257           000040                          DTYERR= BIT5            ;DRIVE TYPE ERROR
12258           000100                          ACLOERR=BIT6            ;AC LOW ERROR
12259           000200                          SPDERR= BIT7            ;SPEED LOSS ERROR
12260           000400                          DROTERR=BIT8            ;DRIVE OFF TRACK ERROR
12261           001000                          COERR=  BIT9            ;CYLINDER OVER FLOW ERROR
12262           002000                          IDAERR= BIT10           ;ILLEGAL DISK ADDRESS ERROR
12263           004000                          WLERR=  BIT11           ;WRITE LOCK ERROR
12264           010000                          DTERR=  BIT12           ;DRIVE TIMING ERROR
12265           020000                          NCERWE= BIT13           ;NO CERR WITH ERROR SET ERROR
12266           040000                          UNSERR= BIT14           ;DRIVE UNSAFE ERROR
12267           100000                          CERNER= BIT15           ;CERR BUT NO ERROR SET ERROR
12268
12269   047456  000000                  GRF2ER: .WORD   0               ;GROUP 2 ERROR FIELD
12270           000001                          ECHERR= BIT0            ;ECC HARD ERROR
12271           000002                          DCKERR= BIT1            ;DATA CHECK ERROR
12272           000004                          WCKERR= BIT2            ;WRITE CHECK ERROR
12273           000010                          DLTERR= BIT3            ;DATA LATE ERROR
12274           000020                          OPIERR= BIT4            ;OPERATION INCOMPLETE ERROR
12275           000040                          HVRCERR=BIT5            ;HEADER VRC ERROR
12276           000100                          BSERR=  BIT6            ;BAD SECTOR ERROR
12277
12278   047460  000000                  GRP3ER: .WORD   0               ;GROUP 3 ERROR FIELD
12279           000001                          NEDERR= BIT0            ;NON-EXISTANT DRIVE ERROR
12280           000002                          CTOERR= BIT1            ;CONTROLLER TIME OUT ERROR
12281           000004                          UFERR=  BIT2            ;UNIT FIELD ERROR
12282           000010                          MDSERR= BIT3            ;MULTIPLE DRIVE SELECT ERROR
12283           000020                          PGERR=  BIT4            ;PROGRAMMING ERROR
12284           000040                          NEMERR= BIT5            ;NON-EXISTANT MEMORY ERROR
```

```
 12285            000100                      UPERR=  BIT6            ;UNIBUS PARITY ERROR
 12286            000200                      ILFERR= BIT7            ;ILLEGAL FUNCTION ERROR.
 12287
 12288  047462    000000              GRP4ER: .WORD   0               ;GROUP 4 ERROR FIELD
 12289            000001                      WCERR=  BIT0            ;WORD COUNT ERROR FLAG
 12290            000002                      UBAERR= BIT1            ;UPPER BA ERROR
 12291            000004                      BAERR=  BIT2            ;BUS ADDRESS ERROR FLAG
 12292            000010                      CYLERR= BIT3            ;CYL ADDRESS ERROR FLAG
 12293            000020                      TRKERR= BIT4            ;TRACK ADDRESS ERROR FLAG
 12294            000040                      SECERR= BIT5            ;SECTOR ADDRESS ERROR FLAG
 12295
 12296  047464    053774              GRP4MS: .WORD   EM10
 12297  047466    054047                      .WORD   EM11A
 12298  047470    054021                      .WORD   EM11
 12299  047472    054123                      .WORD   EM12
 12300  047474    054156                      .WORD   EM13
 12301  047476    054206                      .WORD   EM14
 12302
 12303  047500    054240              GRP3MS: .WORD   EM15
 12304  047502    054263                      .WORD   EM16
 12305  047504    054306                      .WORD   EM17
 12306  047506    054327                      .WORD   EM18
 12307  047510    054354                      .WORD   EM19
 12308  047512    054376                      .WORD   EM20
 12309  047514    054422                      .WORD   EM21
 12310  047516    054446                      .WORD   EM22
 12311
 12312  047520    054467              GRP2MS: .WORD   EM23
 12313  047522    054500                      .WORD   EM24
 12314  047524    054513                      .WORD   EM25
 12315  047526    054527                      .WORD   EM26
 12316  047530    054541                      .WORD   EM27
 12317  047532    054566                      .WORD   EM28
 12318  047534    054601                      .WORD   EM29
 12319
 12320  047536    054622              GRP1MS: .WORD   EM30
 12321  047540    054675                      .WORD   EM31
 12322  047542    054715                      .WORD   EM32
 12323  047544    054753                      .WORD   EM33
 12324  047546    055021                      .WORD   EM34
 12325  047550    055036                      .WORD   EM35
 12326  047552    055057                      .WORD   EM36
 12327  047554    055066                      .WORD   EM37
 12328  047556    055111                      .WORD   EM38
 12329  047560    055131                      .WORD   EM39
 12330  047562    055153                      .WORD   EM40
 12331  047564    055205                      .WORD   EM41
 12332  047566    055226                      .WORD   EM42
 12333  047570    055251                      .WORD   EM43
 12334  047572    055313                      .WORD   EM44
 12335  047574    055330                      .WORD   EM45
 12336
 12337  047576    000000              GPSUMF: .WORD   0               ;GROUP ERROR SUMMARY FLAGS
 12338            000001                      GRP1ST= BIT0            ;GROUP 1 ERROR SET
 12339            000002                      GRP2ST= BIT1            ;GROUP 2 ERROR SET
 12340            000004                      GRP3ST= BIT2            ;GROUP 3 ERROR SET
```

```
12341                   000010                  GP1NR=  BIT3            ;GROUP 1 ERROR NOT RECEIVED
12342                   000020                  GP2NR=  BIT4            ;GROUP 2 ERROR NOT RECEIVED
12343                   000040                  GP3NR=  BIT5            ;GROUP 3 ERROR NOT RECEIVED
12344                   040000                  DRSTER= BIT14           ;ERROR IN GETTING DRIVE STATUS FLAG.
12345                   100000                  REPNR=  BIT15           ;REPORTING NOT RECEIVED SWITCH
12346
12347                                           .SBTTL  TABLE OF OPERATION MESSAGE ADDRESS
12348                                           ;*      THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE
12349                                           ;*      USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.
12350
12351   047600   047640                  CMNDLB: .WORD   OPER00          ;ADDRESS OF SELECT MESSAGE
12352   047602   047655                          .WORD   OPER02          ;               PACK ACK
12353   047604   047666                          .WORD   OPER04          ;               DRIVE CLEAR
12354   047606   047702                          .WORD   OPER06          ;               UNLOAD
12355   047610   047711                          .WORD   OPER10          ;               START SPINDLE
12356   047612   047727                          .WORD   OPER12          ;               RECALIBRATE
12357   047614   047743                          .WORD   OPER14          ;               OFFSET
12358   047616   047752                          .WORD   OPER16          ;               SEEK
12359   047620   047757                          .WORD   OPER20          ;               READ DATA
12360   047622   047771                          .WORD   OPER22          ;               WRITE DATA
12361   047624   050004                          .WORD   OPER24          ;               READ HEADER
12362   047626   050021                          .WORD   OPER26          ;               WRITE HEADER
12363   047630   050037                          .WORD   OPER30          ;               WRITE CHECK
12364   047632   050053                          .WORD   OPER32          ;ILLEGAL OPERATION 33
12365   047634   050077                          .WORD   OPER34          ;                  35
12366   047636   050123                          .WORD   OPER36          ;                  37
12367
12368                                           .SBTTL  OPERATION MESSAGES
12369   047640   051104   053111   020105  OPER00: .ASCIZ  /DRIVE SELECT/
12370   047646   042523   042514   052103
12371   047654      000
12372   047655      120   041501   020113  OPER02: .ASCIZ  /PACK ACK/
12373   047662   041501   000113
12374   047666   051104   053111   020105  OPER04: .ASCIZ  /DRIVE CLEAR/
12375   047674   046103   040505   000122
12376   047702   047125   047514   042101  OPER06: .ASCIZ  /UNLOAD/
12377   047710      000
12378   047711      123   040524   052122  OPER10: .ASCIZ  /START SPINDLE/
12379   047716   051440   044520   042116
12380   047724   042514      000
12381   047727      122   041505   046101  OPER12: .ASCIZ  /RECALIBRATE/
12382   047734   041111   040522   042524
12383   047742      000
12384   047743      117   043106   042523  OPER14: .ASCIZ  /OFFSET/
12385   047750   000124
12386   047752   042523   045505      000  OPER16: .ASCIZ  /SEEK/
12387   047757      122   040505   020104  OPER20: .ASCIZ  /READ DATA/
12388   047764   040504   040524      000
12389   047771      127   044522   042524  OPER22: .ASCIZ  /WRITE DATA/
12390   047776   042040   052101   000101
12391   050004   042522   042101   044040  OPER24: .ASCIZ  /READ HEADERS/
12392   050012   040505   042504   051522
12393   050020      000
12394   050021      127   044522   042524  OPER26: .ASCIZ  /WRITE HEADERS/
12395   050026   044040   040505   042504
12396   050034   051522      000
```

```
12397  050037     127  044522  042524  OPER30: .ASCIZ  /WRITE CHECK/
12398  050044  041440  042510  045503
12399  050052     000
12400  050053     111  046114  043505  OPER32: .ASCIZ  /ILLEGAL FUNCTION 33/
12401  050060  046101  043040  047125
12402  050066  052103  047511  020116
12403  050074  031463     000
12404  050077     111  046114  043505  OPER34: .ASCIZ  /ILLEGAL FUNCTION 35/
12405  050104  046101  043040  047125
12406  050112  052103  047511  020116
12407  050120  032463     000
12408  050123     111  046114  043505  OPER36: .ASCIZ  /ILLEGAL FUNCTION 37/
12409  050130  046101  043040  047125
12410  050136  052103  047511  020116
12411  050144  033463     000
12412  050147     127  044522  042524  OPER37: .ASCIZ  /WRITE DATA ABORTED WITH BSE/
12413  050154  042040  052101  020101
12414  050162  041101  051117  042524
12415  050170  020104  044527  044124
12416  050176  041040  042523     000
12417  050203                          OPER40:
12418  050203     127  044522  042524  OPER41: .ASCIZ  /WRITE CHECK ABORTED WITH WCE/
12419  050210  041440  042510  045503
12420  050216  040440  047502  052122
12421  050224  042105  053440  052111
12422  050232  020110  041527  000105
12423  050240  051127  052111  020105  OPER42: .ASCIZ  /WRITE DATA ABORTED WITH NON-EXISTANT MEMORY/
12424  050246  040504  040524  040440
12425  050254  047502  052122  042105
12426  050262  053440  052111  020110
12427  050270  047516  026516  054105
12428  050276  051511  040524  052116
12429  050304  046440  046505  051117
12430  050312  000131
12431  050314  042522  042101  042040  OPER43: .ASCIZ  /READ DATA ABORTED WITH NON-EXISTANT MEMORY/
12432  050322  052101  020101  041101
12433  050330  051117  042524  020104
12434  050336  044527  044124  047040
12435  050344  047111  042455  044530
12436  050352  052123  047101  020124
12437  050360  042515  047515  054522
12438  050366     000
12439  050367     127  044522  042524  OPER44: .ASCIZ  /WRITE DATA ABORTED WITH UNIBUS PARITY ERROR/
12440  050374  042040  052101  020101
12441  050402  041101  051117  042524
12442  050410  020104  044527  044124
12443  050416  052440  044516  052502
12444  050424  020123  040520  044522
12445  050432  054524  042440  051122
12446  050440  051117     000
12447
```

```
12448                                           .SBTTL  ASCII MESSAGES
12449
12450  050443     040  000040                  SPACE2: .ASCIZ  /  /
12451  050446  005015  045522  030466  OPR001: .ASCIZ  <15><12>/RK611 BUS ADDRESS ( /
12452  050454  020061  052502  020123
12453  050462  042101  051104  051505
12454  050470  020123  020050     000
12455  050475     040  020051  020075  OPR002: .ASCIZ  / ) = /
12456  050502     000
12457  050503     122  033113  030461  OPR003: .ASCIZ  /RK611 VECTOR ADDRESS ( /
12458  050510  053040  041505  047524
12459  050516  020122  042101  051104
12460  050524  051505  020123  020050
12461  050532     000
12462  050533     122  033113  030461  OPR004: .ASCIZ  /RK611 PRIORITY ( /
12463  050540  050040  044522  051117
12464  050546  052111  020131  020050
12465  050554     000
12466  050555     111  051516  043125  OPR005: .ASCIZ  /INSUFFICIENT MEMORY. PROGRAM ABORTING./<15><12>
12467  050562  044506  044503  047105
12468  050570  020124  042515  047515
12469  050576  054522  020056  051120
12470  050604  043517  040522  020115
12471  050612  041101  051117  044524
12472  050620  043516  006456  000012
12473  050626  005015  047524  041040  OPR006: .ASCII  <15><12>/TO BYPASS TESTING DRIVE 0, PLACE IT OFF-LINE/
12474  050634  050131  051501  020123
12475  050642  042524  052123  047111
12476  050650  020107  051104  053111
12477  050656  020105  026060  050040
12478  050664  040514  042503  044440
12479  050672  020124  043117  026506
12480  050700  044514  042516
12481  050704  005015  047524  052040          .ASCIZ  <15><12>/TO TEST DRIVE 0, REPLACE PROGRAM PACK WITH SCRATCH PACK/
12482  050712  051505  020124  051104
12483  050720  053111  020105  026060
12484  050726  051040  050105  040514
12485  050734  042503  050040  047522
12486  050742  051107  046501  050040
12487  050750  041501  020113  044527
12488  050756  044124  051440  051103
12489  050764  052101  044103,  050040
12490  050772  041501  000113
12491  050776  005015  051104  053111  OPR007: .ASCII  <15><12>/DRIVE 0 WILL NOT BE TESTED. TO TEST DRIVE 0,/<15><12>
12492  051004  020105  020060  044527
12493  051012  046114  047040  052117
12494  051020  041040  020105  042524
12495  051026  052123  042105  020056
12496  051034  047524  052040  051505
12497  051042  020124  051104  053111
12498  051050  020105  026060  005015
12499  051056  042522  052123  051101          .ASCIZ  /RESTART AT 214 AND MOUNT SCRATCH PACK AS DIRECTED./
12500  051064  020124  052101  031040
12501  051072  032061  040440  042116
12502  051100  046440  052517  052116
12503  051106  051440  051107  052101
```

```
12504  051114  044103  050040  041501
12505  051122  020113  051501  042040
12506  051130  051111  041505  042524
12507  051136  027104     000
12508  051141     015  047012  020117  OPR008: .ASCIZ  <15><12>/NO DRIVES AVAILABLE FOR TESTING. PROGRAM ABORTED/
12509  051146  051104  053111  051505
12510  051154  040440  040526  046111
12511  051162  041101  042514  043040
12512  051170  051117  052040  051505
12513  051176  044524  043516  020056
12514  051204  051120  043517  040523
12515  051212  020115  041101  051117
12516  051220  042524  000104
12517  051224  005015  044124  020105  OPR009: .ASCIZ  <15><12>/THE FOLLOWING DRIVES WILL BE TESTED/<15><12>
12518  051232  047506  046114  053517
12519  051240  047111  020107  051104
12520  051246  053111  051505  053440
12521  051254  046111  020114  042502
12522  051262  052040  051505  042524
12523  051270  006504  000012
12524  051274  005015  047516  050040  OPR010: .ASCII  <15><12>/NO PARITY OPTION FOR MEMORY AREA IN USE/<15><12>
12525  051302  051101  052111  020131
12526  051310  050117  044524  047117
12527  051316  043040  051117  046440
12528  051324  046505  051117  020131
12529  051332  051101  040505  044440
12530  051340  020116  051525  006505
12531  051346     012
12532  051347     106  051117  052040          .ASCII  /FOR TEST - UNIBUS PARITY ERROR AND/<15><12>
12533  051354  051505  020124  020055
12534  051362  047125  041111  051525
12535  051370  050040  051101  052111
12536  051376  020131  051105  047522
12537  051404  020122  047101  006504
12538  051412     012
12539  051413     062  024064  024470          .ASCIZ  /24(8) SECTOR FORMAT DATA XFER TEST BYPASSED/<15><12>
12540  051420  051440  041505  047524
12541  051426  020122  047506  046522
12542  051434  052101  042040  052101
12543  051442  020101  043130  051105
12544  051450  052040  051505  020124
12545  051456  054502  040520  051523
12546  051464  042105  005015     000
12547  051471     015  046412  046505  OPR011: .ASCII  <15><12>/MEMORY SIZE NOT LARGE ENOUGH FOR BUS ADDRESS BITS 16 & 17 TESTS
12548  051476  051117  020131  044523
12549  051504  042532  047040  052117
12550  051512  046040  051101  042507
12551  051520  042440  047516  043525
12552  051526  020110  047506  020122
12553  051534  052502  020123  042101
12554  051542  051104  051505  020123
12555  051550  044502  051524  030440
12556  051556  020062  020046  033461
12557  051564  052040  051505  051524
12558  051572  005015
12559  051574  046101  020114  040504          .ASCIZ  /ALL DATA XFER TESTS WITH ADDR >/
```

```
12560  051602  040524  054040  042536
12561  051610  020122  042524  052123
12562  051616  020123  044527  044124
12563  051624  040440  042104  020122
12564  051632  000076
12565  051634  031063  000113          OPR012: .ASCIZ  /32K/
12566  051640  032066  000113          OPR013: .ASCIZ  /64K/
12567  051644  033071  000113          OPR014: .ASCIZ  /96K/
12568  051650  041040  050131  051501  OPR015: .ASCIZ  / BYPASSED/<15><12>
12569  051656  042523  006504  000012
12570  051664  005015  020012  020040  OPR016: .ASCII  <15><12><12>/       *** CAUTION ***/<15><12><12>
12571  051672  020040  020040  025052
12572  051700  020052  040503  052125
12573  051706  047511  020116  025052
12574  051714  006452  005012
12575  051720  044124  051511  050040          .ASCII  /THIS PROGRAM SHOULD BE HALTED BY TYPING ↑C./<15><12>
12576  051726  047522  051107  046501
12577  051734  051440  047510  046125
12578  051742  020104  042502  044040
12579  051750  046101  042524  020104
12580  051756  054502  052040  050131
12581  051764  047111  020107  041536
12582  051772  006456     012
12583  051775     111  020106  040510          .ASCII  /IF HALTED USING THE HALT KEY, THE STATE OF THE DRIVE/<15><12>
12584  052002  052114  042105  052440
12585  052010  044523  043516  052040
12586  052016  042510  044040  046101
12587  052024  020124  042513  026131
12588  052032  052040  042510  051440
12589  052040  040524  042524  047440
12590  052046  020106  044124  020105
12591  052054  051104  053111  006505
12592  052062     012
12593  052063     117  020122  040503          .ASCII  /OR CARTRIDGE CANNOT BE PREDICTED./<15><12><12>
12594  052070  052122  044522  043504
12595  052076  020105  040503  047116
12596  052104  052117  041040  020105
12597  052112  051120  042105  041511
12598  052120  042524  027104  005015
12599  052126     012
12600  052127     101  046114  042040          .ASCII  /ALL DRIVES TO BE TESTED MUST BE ON-LINE,/<15><12>
12601  052134  044522  042526  020123
12602  052142  047524  041040  020105
12603  052150  042524  052123  042105
12604  052156  046440  051525  020124
12605  052164  042502  047440  026516
12606  052172  044514  042516  006454
12607  052200     012
12608  052201     122  040505  054504          .ASCII  /READY, AND WRITE LOCK RESET./<15><12>
12609  052206  020054  047101  020104
12610  052214  051127  052111  020105
12611  052222  047514  045503  051040
12612  052230  051505  052105  006456
12613  052236     012
12614  052237     101  054516  042040          .ASCII  /ANY DRIVE NOT TO BE TESTED MUST BE OFF-LINE./<15><12><12>
12615  052244  044522  042526  047040
```

```
12616  052252  052117  052040  020117
12617  052260  042502  052040  051505
12618  052266  042524  020104  052515
12619  052274  052123  041040  020105
12620  052302  043117  026506  044514
12621  052310  042516  006456  005012
12622  052316  047516  042524  020072          .ASCII  /NOTE: 2ND AND SUBSEQUENT PASS RUN TIME IS/<15><12>
12623  052324  047062  020104  047101
12624  052332  020104  052523  051502
12625  052340  050505  042525  052116
12626  052346  050040  051501  020123
12627  052354  052522  020116  044524
12628  052362  042515  044440  006523
12629  052370     012
12630  052371     040  020040  020040          .ASCIZ  /      APPROX 2 MIN 30 SEC FOR EACH DRIVE./<15><12>
12631  052376  040440  050120  047522
12632  052404  020130  020062  044515
12633  052412  020116  030063  051440
12634  052420  041505  043040  051117
12635  052426  042440  041501  020110
12636  052434  051104  053111  027105
12637  052442  005015     000
12638  052445     015  043012  051111  OPRO17: .ASCII  <15><12>/FIRST 256 SECTORS NOT BSE ERROR FREE./
12639  052452  052123  031040  033065
12640  052460  051440  041505  047524
12641  052466  051522  047040  052117
12642  052474  041040  042523  042440
12643  052502  051122  051117  043040
12644  052510  042522  027105
12645  052514  040515  044530  052515          .ASCIZ  /MAXIMUM DATA TRANSFER TEST BYPASSED/<15><12>
12646  052522  020115  040504  040524
12647  052530  052040  040522  051516
12648  052536  042506  020122  042524
12649  052544  052123  041040  050131
12650  052552  051501  042523  006504
12651  052560  000012
12652  052562  020040  020040  006440  OPRO18: .ASCIZ  /     /<15><12>/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/<15><12>
12653  052570  047412  046116  020131
12654  052576  020061  051104  053111
12655  052604  027105  047440  042526
12656  052612  046122  050101  042520
12657  052620  020104  050117  051105
12658  052626  052101  047511  051516
12659  052634  041040  050131  051501
12660  052642  042523  006504  000012
12661  052650  005015  041523  050117  OPRO19: .ASCII  <15><12>@SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)@
12662  052656  035105  041440  030510
12663  052664  024040  051124  043511
12664  052672  026051  042440  031465
12665  052700  034055  023073  044103
12666  052706  026062  042440  034464
12667  052714  031055  024040  041501
12668  052722  041440  052517  046120
12669  052730  026105  027040  053062
12670  052736  041457  024515
12671  052742  005015  042101  052512          .ASCIZ  <15><12>/ADJUST R72 FOR CONSTANT LEVEL ON CH2/<15><12>
```

```
12672  052750  052123  051040  031067
12673  052756  043040  051117  041440
12674  052764  047117  052123  047101
12675  052772  020124  042514  042526
12676  053000  020114  047117  041440
12677  053006  031110  005015      000
12678  053013      015  050012  047522  OPR020: .ASCIZ  <15><12>/PROGRAM HALT PENDING - CARTRIDGE FORMAT BEING CORRECTED/<15><12
12679  053020  051107  046501  044040
12680  053026  046101  020124  042520
12681  053034  042116  047111  020107
12682  053042  020055  040503  052122
12683  053050  044522  043504  020105
12684  053056  047506  046522  052101
12685  053064  041040  044505  043516
12686  053072  041440  051117  042522
12687  053100  052103  042105  005015
12688  053106      000
12689  053107      015  025012  025052  OPR021: .ASCIZ  <15><12>/*****  PROGRAM HALTED  *****/<15><12>
12690  053114  025052  020040  051120
12691  053122  043517  040522  020115
12692  053130  040510  052114  042105
12693  053136  020040  025052  025052
12694  053144  006452  000012
12695  053150  040503  052122  044522  OPR022: .ASCIZ  /CARTRIDGE FORMAT CORRECTION FAILED/<15><12>
12696  053156  043504  020105  047506
12697  053164  046522  052101  041440
12698  053172  051117  042522  052103
12699  053200  047511  020116  040506
12700  053206  046111  042105  005015
12701  053214      000
12702  053215      015  050012  047522  ABORT:  .ASCIZ  <15><12>/PROGRAM ABORTING BECAUSE ERROR THRESHOLD EXCEEDF      <12>
12703  053222  051107  046501  040440
12704  053230  047502  052122  047111
12705  053236  020107  042502  040503
12706  053244  051525  020105  051105
12707  053252  047522  020122  044124
12708  053260  042522  044123  046117
12709  053266  020104  054105  042503
12710  053274  042105  042105  005015
12711  053302      000
12712
12713                                  .SBTTL ERROR MESSAGES
12714  053303      106  052101  046101  EM1:    .ASCIZ  /FATAL-NON EXISTANT MEMORY AT RK611 BASE ADDRESS/
12715  053310  047055  047117  042440
12716  053316  044530  052123  047101
12717  053324  020124  042515  047515
12718  053332  054522  040440  020124
12719  053340  045522  030466  020061
12720  053346  040502  042523  040440
12721  053354  042104  042522  051523
12722  053362      000
12723  053363      106  052101  046101  EM2:    .ASCIZ  /FATAL-WRITE READY AND IE DID NOT CAUSE INTERRUPT/
12724  053370  053455  044522  042524
12725  053376  051040  040505  054504
12726  053404  040440  042116  044440
12727  053412  020105  044504  020104
```

```
12728  053420  047516  020124  040503
12729  053426  051525  020105  047111
12730  053434  042524  051122  050125
12731  053442  000124
12732  053444  040506  040524  026514  EM3:    .ASCIZ  /FATAL-PARITY ERROR TRAP. PC AT ERROR = /
12733  053452  040520  044522  054524
12734  053460  042440  051122  051117
12735  053466  052040  040522  027120
12736  053474  050040  020103  052101
12737  053502  042440  051122  051117
12738  053510  036440  000040
12739  053514  054105  042520  052103  EM4:    .ASCIZ  /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
12740  053522  042105  044440  052116
12741  053530  051105  052522  052120
12742  053536  042040  042111  047040
12743  053544  052117  047440  041503
12744  053552  051125  047440  020122
12745  053560  040527  020123  040514
12746  053566  042524  020056  047503
12747  053574  046515  047101  020104
12748  053602  040527  035123     000
12749  053607     123  041125  054523  EM5:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT RESET ERROR/
12750  053614  052123  046505  041440
12751  053622  042514  051101  042040
12752  053630  042111  047040  052117
12753  053636  051040  051505  052105
12754  053644  042440  051122  051117
12755  053652     000
12756  053653     123  041125  054523  EM6:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
12757  053660  052123  046505  041440
12758  053666  042514  051101  042040
12759  053674  042111  047040  052117
12760  053702  051040  051505  052105
12761  053710  042040  053105  041511
12762  053716  020105  047111  042524
12763  053724  051122  050125  000124
12764  053732  052523  051502  051531  EM7:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT SET READY/
12765  053740  042524  020115  046103
12766  053746  040505  020122  044504
12767  053754  020104  047516  020124
12768  053762  042523  020124  042522
12769  053770  042101  000131
12770  053774  047527  042122  041440  EM10:   .ASCIZ  /WORD COUNT INCORRECT/
12771  054002  052517  052116  044440
12772  054010  041516  051117  042522
12773  054016  052103     000
12774  054021     102  051525  040440  EM11:   .ASCIZ  /BUS ADDRESS INCORRECT/
12775  054026  042104  042522  051523
12776  054034  044440  041516  051117
12777  054042  042522  052103     000
12778  054047     125  050120  051105  EM11A:  .ASCIZ  /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
12779  054054  041040  051525  040440
12780  054062  042104  042522  051523
12781  054070  041040  052111  020123
12782  054076  047111  047503  051122
12783  054104  041505  020124  041050
```

```
12784  054112  030501  026066  030440
12785  054120  024467     000
12786  054123     103  046131  047111  EM12:   .ASCIZ  /CYLINDER ADDRESS INCORRECT/
12787  054130  042504  020122  042101
12788  054136  051104  051505  020123
12789  054144  047111  047503  051122
12790  054152  041505  000124
12791  054156  051124  041501  020113  EM13:   .ASCIZ  /TRACK ADDRESS INCORRECT/
12792  054164  042101  051104  051505
12793  054172  020123  047111  047503
12794  054200  051122  041505  000124
12795  0542    042523  052103  051117  EM14:   .ASCIZ  /SECTOR ADDRESS INCORRECT./
12796  0542    040440  042104  042522
12797  05422   051523  044440  041516
12798  054230  051117  042522  052103
12799  054236  000056
12800  054240  047516  026516  054105  EM15:   .ASCIZ  /NON-EXISTANT DRIVE/
12801  054246  051511  040524  052116
12802  054254  042040  044522  042526
12803  054262     000
12804  054263     103  047117  051124  EM16:   .ASCIZ  /CONTROLLER TIMEOUT/
12805  054270  046117  042514  020122
12806  054276  044524  042515  052517
12807  054304  000124
12808  054306  047125  052111  043040  EM17:   .ASCIZ  /UNIT FIELD ERROR/
12809  054314  042511  042114  042440
12810  054322  051122  051117     000
12811  054327     115  046125  050111  EM18:   .ASCIZ  /MULIPLE DRIVE SELECT/
12812  054334  042514  042040  044522
12813  054342  042526  051440  046105
12814  054350  041505  000124
12815  054354  051120  043517  040522  EM19:   .ASCIZ  /PROGRAMMING ERROR/
12816  054362  046515  047111  020107
12817  054370  051105  047522  000122
12818  054376  047516  026516  054105  EM20:   .ASCIZ  /NON-EXISTANT MEMORY/
12819  054404  051511  040524  052116
12820  054412  046440  046505  051117
12821  054420  000131
12822  054422  047125  041111  051525  EM21:   .ASCIZ  /UNIBUS PARITY ERROR/
12823  054430  050040  051101  052111
12824  054436  020131  051105  047522
12825  054444  000122
12826  054446  046111  042514  040507  EM22:   .ASCIZ  /ILLEGAL FUNCTION/
12827  054454  020114  052506  041516
12828  054462  044524  047117     000
12829  054467     105  041503  044040  EM23:   .ASCIZ  /ECC HARD/
12830  054474  051101  000104
12831  054500  040504  040524  041440  EM24:   .ASCIZ  /DATA CHECK/
12832  054506  042510  045503     000
12833  054513     127  044522  042524  EM25:   .ASCIZ  /WRITE CHECK/
12834  054520  041440  042510  045503
12835  054526     000
12836  054527     104  052101  020101  EM26:   .ASCIZ  /DATA LATE/
12837  054534  040514  042524     000
12838  054541     117  042520  040522  EM27:   .ASCIZ  /OPERATION INCOMPLETE/
12839  054546  044524  047117  044440
```

```
12840  054554  041516  046517  046120
12841  054562  052105  000105
12842  054566  042510  042101  051105  EM28:   .ASCIZ  /HEADER VRC/
12843  054574  053040  041522     000
12844  054601     102  042101  051440  EM29:   .ASCIZ  /BAD SECTOR ERROR/
12845  054606  041505  047524  020122
12846  054614  051105  047522  000122
12847  054622  047503  052116  047522  EM30:   .ASCIZ  /CONTROLLER DETECTED DRIVE BUS PARITY ERROR/
12848  054630  046114  051105  042040
12849  054636  052105  041505  042524
12850  054644  020104  051104  053111
12851  054652  020105  052502  020123
12852  054660  040520  044522  054524
12853  054666  042440  051122  051117
12854  054674     000
12855  054675     123  042505  020113  EM31:   .ASCIZ  /SEEK INCOMPLETE/
12856  054702  047111  047503  050115
12857  054710  042514  042524     000
12858  054715     116  047117  042455  EM32:   .ASCIZ  /NON-EXECUTABLE DRIVE FUNCTION/
12859  054722  042530  052503  040524
12860  054730  046102  020105  051104
12861  054736  053111  020105  052506
12862  054744  041516  044524  047117
12863  054752     000
12864  054753     104  044522  042526  EM33:   .ASCIZ  /DRIVE DETECTED DRIVE BUS PARITY ERROR/
12865  054760  042040  052105  041505
12866  054766  042524  020104  051104
12867  054774  053111  020105  052502
12868  055002  020123  040520  044522
12869  055010  054524  042440  051122
12870  055016  051117     000
12871  055021     106  051117  040515  EM34:   .ASCIZ  /FORMAT ERROR/
12872  055026  020124  051105  047522
12873  055034  000122
12874  055036  051104  053111  020105  EM35:   .ASCIZ  /DRIVE TYPE ERROR/
12875  055044  054524  042520  042440
12876  055052  051122  051117     000
12877  055057     101  020103  047514  EM36:   .ASCIZ  /AC LOW/
12878  055064  000127
12879  055066  050123  047111  046104  EM37:   .ASCIZ  /SPINDLE SPEED LOSS/
12880  055074  020105  050123  042505
12881  055102  020104  047514  051523
12882  055110     000
12883  055111     104  044522  042526  EM38:   .ASCIZ  /DRIVE OFF TRACK/
12884  055116  047440  043106  052040
12885  055124  040522  045503     000
12886  055131     103  046131  047111  EM39:   .ASCIZ  /CYLINDER OVERFLOW/
12887  055136  042504  020122  053117
12888  055144  051105  046106  053517
12889  055152     000
12890  055153     111  046114  043505  EM40:   .ASCIZ  /ILLEGAL DISK PACK ADDRESS/
12891  055160  046101  042040  051511
12892  055166  020113  040520  045503
12893  055174  040440  042104  042522
12894  055202  051523     000
12895  055205     127  044522  042524  EM41:   .ASCIZ  /WRITE LOCK ERROR/
```

```
12896  055212  046040  041517  020113
12897  055220  051105  047522  000122
12898  055226  051104  053111  020105  EM42:   .ASCIZ  /DRIVE TIMING ERROR/
12899  055234  044524  044515  043516
12900  055242  042440  051122  051117
12901  055250     000
12902  055251     116  020117  042503  EM43:   .ASCIZ  /NO CERR WITH SOME OTHER ERROR SET/
12903  055256  051122  053440  052111
12904  055264  020110  047523  042515
12905  055272  047440  044124  051105
12906  055300  042440  051122  051117
12907  055306  051440  052105     000
12908  055313     104  044522  042526  EM44:   .ASCIZ  /DRIVE UNSAFE/
12909  055320  052440  051516  043101
12910  055326  000105
12911  055330  042503  051122  051440  EM45:   .ASCIZ  /CERR SET BUT NO OTHER ERROR SET/
12912  055336  052105  041040  052125
12913  055344  047040  020117  052117
12914  055352  042510  020122  051105
12915  055360  047522  020122  042523
12916  055366  000124
12917
12918  055370  053126  042040  042111  EM46:   .ASCIZ  /VV DID NOT SET WITH PACK ACK/
12919  055376  047040  052117  051440
12920  055404  052105  053440  052111
12921  055412  020110  040520  045503
12922  055420  040440  045503     000
12923  055425     123  040524  052524  EM47:   .ASCIZ  /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
12924  055432  020123  040526  044514
12925  055440  020104  042523  020124
12926  055446  047117  051440  046105
12927  055454  041505  020124  047524
12928  055462  047040  047117  042455
12929  055470  044530  052123  047101
12930  055476  020124  051104  053111
12931  055504  000105
12932  055506  052123  052101  051525  EM48:   .ASCIZ  /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
12933  055514  053040  046101  042111
12934  055522  051040  051505  052105
12935  055530  047440  020116  042523
12936  055536  042514  052103  052040
12937  055544  020117  054105  051511
12938  055552  040524  052116  042040
12939  055560  044522  042526     000
12940  055565     123  040524  052524  EM49:   .ASCIZ  /STATUS VALID NOT RESET ON DRIVE RELEASE/
12941  055572  020123  040526  044514
12942  055600  020104  047516  020124
12943  055606  042522  042523  020124
12944  055614  047117  042040  044522
12945  055622  042526  051040  046105
12946  055630  040505  042523     000
12947  055635     105  050130  041505  EM50:   .ASCIZ  /EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS: /
12948  055642  042524  020104  047062
12949  055650  020104  047111  042524
12950  055656  051122  050125  020124
12951  055664  044504  020104  047516
```

```
 12952  055672  020124  041517  052503
 12953  055700  020122  051117  053440
 12954  055706  051501  046040  052101
 12955  055714  027105  041440  046517
 12956  055722  040515  042116  053440
 12957  055730  051501  000072
 12958  055734  040503  047116  052117  EM51:   .ASCII  /CANNOT READ BAD SECTOR FILE/<15><12>
 12959  055742  051040  040505  020104
 12960  055750  040502  020104  042523
 12961  055756  052103  051117  043040
 12962  055764  046111  006505     012
 12963  055771     124  051505  044524          .ASCIZ  /TESTING ABORTED ON THIS DRIVE/
 12964  055776  043516  040440  047502
 12965  056004  052122  042105  047440
 12966  056012  020116  044124  051511
 12967  056020  042040  044522  042526
 12968  056026     000
 12969  056027     101  044514  047107  EM52:   .ASCII  /ALIGNMENT PACK ON THIS DRIVE./<15><12>
 12970  056034  042515  052116  050040
 12971  056042  041501  020113  047117
 12972  056050  052040  044510  020123
 12973  056056  051104  053111  027105
 12974  056064  005015
 12975  056066  042524  052123  047111          .ASCIZ  /TESTING ABORTED ON THIS DRIVE./
 12976  056074  020107  041101  051117
 12977  056102  042524  020104  047117
 12978  056110  052040  044510  020123
 12979  056116  051104  053111  027105
 12980  056124     000
 12981  056125     124  020117  040515  EM53:   .ASCIZ  /TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
 12982  056132  054516  041040  042101
 12983  056140  051440  041505  047524
 12984  056146  051522  020056  042524
 12985  056154  052123  047111  020107
 12986  056162  041101  051117  042524
 12987  056170  020104  047117  052040
 12988  056176  044510  020123  051104
 12989  056204  053111  000105
 12990  056210  042510  042101  051105  EM54:   .ASCIZ  /HEADER WORD MISCOMPARE/
 12991  056216  053440  051117  020104
 12992  056224  044515  041523  046517
 12993  056232  040520  042522     000
 12994  056237     104  052101  020101  EM55:   .ASCIZ  /DATA WORD MISCOMPARE/
 12995  056244  047527  042122  046440
 12996  056252  051511  047503  050115
 12997  056260  051101  000105
 12998
 12999  056264  040503  047116  052117  EM56:   .ASCIZ  /CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/
 13000  056272  043040  047111  020104
 13001  056300  042510  042101  051105
 13002  056306  030040  044440  020116
 13003  056314  042522  042101  040440
 13004  056322  042116  051440  051117
 13005  056330  020124  042510  042101
 13006  056336  051105  000123
 13007  056342  052123  052101  051525  EMSVAL: .ASCIZ  /STATUS VALID/
```

```
 13008  056350  053040  046101  042111
 13009  056356     000
 13010  056357     123  046105  041505  EMNZPR: .ASCIZ  /SELECT WITH NON-ZERO STATUS PAIR/
 13011  056364  020124  044527  044124
 13012  056372  047040  047117  055055
 13013  056400  051105  020117  052123
 13014  056406  052101  051525  050040
 13015  056414  044501  000122
 13016  056420  051127  052111  047111  EMWCS2: .ASCIZ  /WRITING COMMAND STATUS REGISTER 2/
 13017  056426  020107  047503  046515
 13018  056434  047101  020104  052123
 13019  056442  052101  051525  051040
 13020  056450  043505  051511  042524
 13021  056456  020122  000062
 13022          055036                  EMDTPE= EM35            ;DRIVE TYPE ERROR
 13023  056462  051104  053111  020105  EMDI:   .ASCIZ  /DRIVE INTERRUPT/
 13024  056470  047111  042524  051122
 13025  056476  050125  000124
 13026          054753                  EMDPAR= EM33                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
 13027  056502  051104  053111  020105  EMDSC:  .ASCIZ  /DRIVE STATUS CHANGE/
 13028  056510  052123  052101  051525
 13029  056516  041440  040510  043516
 13030  056524  000105
 13031  056526  051104  053111  020105  EMDA:   .ASCIZ  /DRIVE ATTENTION/
 13032  056534  052101  042524  052116
 13033  056542  047511  000116
 13034  056546  047503  052116  047522  EMCCLR: .ASCIZ  /CONTROLLER CLEAR/
 13035  056554  046114  051105  041440
 13036  056562  042514  051101     000
 13037          047640                  EMSELD= OPER00          ;DRIVE SELECT
 13038          047666                  EMDCLR= OPER04                  ;DRIVE CLEAR
 13039          047727                  EMRCAL= OPER12                  ;RECALIBRATE
 13040  056567     123  041505  047117  EM2INT: .ASCIZ  /SECOND INTERRUPT/
 13041  056574  020104  047111  042524
 13042  056602  051122  050125  000124
 13043  056610  042523  045505  052040  EMSKSF: .ASCIZ  /SEEK TO SELF/
 13044  056616  020117  042523  043114
 13045  056624     000
 13046          047752                  EMSK=   OPER16                  ;SEEK
 13047  056625     125  042516  050130  EMUXIT: .ASCIZ  /UNEXPECTED INTERRUPT/
 13048  056632  041505  042524  020104
 13049  056640  047111  042524  051122
 13050  056646  050125  000124
 13051  066652  047125  041111  051525  EMUR:   .ASCIZ  /UNIBUS RESET/
 13052  056660  051040  051505  052105
 13053  056666     000
 13054  056667     122  051505  052105  EMRSET: .ASCIZ  /RESET/
 13055  056674     000
 13056          054527                  EMDLT=  EM26                    ;DATA LATE
 13057  056675     122  040505  044504  EMRDB:  .ASCIZ  /READING DATA BUFFER/
 13058  056702  043516  042040  052101
 13059  056710  020101  052502  043106
 13060  056716  051105     000
 13061  056721     103  047117  051124  EMCERR: .ASCIZ  /CONTROLLER ERROR/
 13062  056726  046117  042514  020122
 13063  056734  051105  047522  000122
```

```
13064  056742  052523  051502  051531  EMSCLR: .ASCIZ  /SUBSYSTEM CLEAR/
13065  056750  042524  020115  046103
13066  056756  040505  000122
13067  056762  052515  052114  050111  EMMI:   .ASCIZ  /MULTIPLE INTERRUPTS/
13068  056770  042514  044440  052116
13069  056776  051105  052522  052120
13070  057004  000123
13071  057006  040502  020104  042523  DRVABT: .ASCII  /BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(8))/<15><12>
13072  057014  052103  051117  020123
13073  057022  047117  050040  041501
13074  057030  020113  047111  040440
13075  057036  042522  051501  052440
13076  057044  042523  020104  054502
13077  057052  052040  051505  020124
13078  057060  041450  046131  031440
13079  057066  031061  034050  024451
13080  057074  005015
13081  057076  042524  052123  047111          .ASCIZ  /TESTING ABORTED ON THIS DRIVE/
13082  057104  020107  041101  051117
13083  057112  042524  020104  047117
13084  057120  052040  044510  020123
13085  057126  051104  053111  000105
13086
13087                                  .SBTTL  DATA HEADERS FOR ERROR REPORTS
13088  057134  051524  020124  052516  DH001:  .ASCIZ  /TST NUM ERR PC  DRIVE/
13089  057142  020115  051105  020122
13090  057150  041520  020040  051104
13091  057156  053111  000105
13092  057162  051524  020124  052516  DH002A: .ASCIZ  /TST NUM ERR PC  DRIVE/
13093  057170  020115  051105  020122
13094  057176  041520  020040  051104
13095  057204  053111  000105
13096  057210  045522  051503  020061  DH002B: .ASCIZ  /RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA/
13097  057216  020040  045522  051503
13098  057224  020062  020040  045522
13099  057232  051504  020040  020040
13100  057240  045522  051105  020040
13101  057246  020040  045522  051501
13102  057254  043117  020040  045522
13103  057262  041504  046131  020040
13104  057270  045522  040504     000
13105  057275     122  041113  020101  DH002C: .ASCIZ  /RKBA    RKWC/
13106  057302  020040  051040  053513
13107  057310  000103
13108  057312  030102  020060  020040  DH002D: .ASCIZ  /A00     B00     A01     B01     A02     B02     A03     B03/
13109  057320  020040  030102  020060
13110  057326  020040  020040  030101
13111  057334  020061  020040  020040
13112  057342  030102  020061  020040
13113  057350  020040  030101  020062
13114  057356  020040  020040  030102
13115  057364  020062  020040  020040
13116  057372  030101  020063  020040
13117  057400  020040  030102  000063
13118  057406  045522  051503  020061  DH003B: .ASCIZ  /RKCS1   RKCS2   RKDS    RKER    RKASOF  RkMR1/
13119  057414  020040  045522  051503
```

```
13120  057422  020062  020040  045522
13121  057430  051504  020040  020040
13122  057436  045522  051105  020040
13123  057444  020040  045522  051501
13124  057452  043117  020040  045522
13125  057460  051115  000061
13126  057464  044124  020105  041101  DH005:  .ASCIZ  /THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:/
13127  057472  053117  020105  051101
13128  057500  020105  054105  042520
13129  057506  052103  042105  042440
13130  057514  051122  051117  020123
13131  057522  044124  052101  042040
13132  057530  042111  047040  052117
13133  057536  051440  052105  044440
13134  057544  020116  050117  051105
13135  057552  052101  047511  035116
13136  057560      000
13137  057561      124  042510  040440  DH006:  .ASCIZ  /THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:/
13138  057566  047502  042526  040440
13139  057574  042522  052440  042516
13140  057602  050130  041505  042524
13141  057610  020104  051105  047522
13142  057616  051522  051440  052105
13143  057624  044440  020116  050117
13144  057632  051105  052101  047511
13145  057640  035116      000
13146  057643      124  042510  040440  DH007:  .ASCIZ  /THE ABOVE ARE ERRORS SET IN OPERATION:/
13147  057650  047502  042526  040440
13148  057656  042522  042440  051122
13149  057664  051117  020123  042523
13150  057672  020124  047111  047440
13151  057700  042520  040522  044524
13152  057706  047117  000072
13153  057712  047101  020131  044506  DH005A: .ASCIZ  /ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID/
13154  057720  046105  020104  044527
13155  057726  044124  040440  046114
13156  057734  047440  042516  020123
13157  057742  052515  052123  041040
13158  057750  020105  047503  051516
13159  057756  042111  051105  042105
13160  057764  044440  053116  046101
13161  057772  042111      000
13162  057775      105  051122  051117  DH010:  .ASCIZ  /ERROR AT COMPLETION OF OPERATION/
13163  060002  040440  020124  047503
13164  060010  050115  042514  044524
13165  060016  047117  047440  020106
13166  060024  050117  051105  052101
13167  060032  047511  000116
13168  060036  054105  052120  020104  DH010A: .ASCIZ  /EXPTD   IS/
13169  060044  020040  051511      000
13170  060051      122  040505  020104  DH010B: .ASCIZ  /READ DATA WITH FORCED CYLINDER OVERFLOW/
13171  060056  040504  040524  053440
13172  060064  052111  020110  047506
13173  060072  041522  042105  041440
13174  060100  046131  047111  042504
13175  060106  020122  053117  051105
```

```
13176  060114  046106  053517     000
13177  060121     116  052117  051440  DH011:  .ASCIZ  /NOT SET AS A RESULT OF/
13178  060126  052105  040440  020123
13179  060134  020101  042522  052523
13180  060142  052114  047440  000106
13181  060150  047516  020124  042522  DH012:  .ASCIZ  /NOT RESET AS A RESULT OF/
13182  060156  042523  020124  051501
13183  060164  040440  051040  051505
13184  060172  046125  020124  043117
13185  060200     000
13186  060201     123  052105  040440  DH013:  .ASCIZ  /SET AS A RESULT OF/
13187  060206  020123  020101  042522
13188  060214  052523  052114  047440
13189  060222  000106
13190  060224  042522  042523  020124  DH014:  .ASCIZ  /RESET AS A RESULT OF/
13191  060232  051501  040440  051040
13192  060240  051505  046125  020124
13193  060246  043117     000
13194  060251     107  047517  020104  DH015:  .ASCIZ  /GOOD    BAD     WORD NUM/
13195  060256  020040  041040  042101
13196  060264  020040  020040  053440
13197  060272  051117  020104  052516
13198  060300  000115
13199  060302  052523  051502  051531  DH016:  .ASCIZ  /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/
13200  060310  042524  020115  046103
13201  060316  040505  020122  047524
13202  060324  051040  051505  052105
13203  060332  051440  042505  020113
13204  060340  044514  044515  020124
13205  060346  046101  047514  044527
13206  060354  043516  044040  040505
13207  060362  051504  052040  020117
13208  060370  042522  047514  042101
13209  060376     000
13210  060377     123  046105  041505  DH017:  .ASCIZ  /SELECT AFTER RECAL/
13211  060404  020124  043101  042524
13212  060412  020122  042522  040503
13213  060420  000114
13214  060422  042522  040503  020114  DH018:  .ASCIZ  /RECAL COMPLETE ATTN AFTER SELECT/
13215  060430  047503  050115  042514
13216  060436  042524  040440  052124
13217  060444  020116  043101  042524
13218  060452  020122  042523  042514
13219  060460  052103     000
13220
13221                                  .SBTTL  DATA TABLES FOR ERROR REPORTS
13222          060464                  .EVEN
13223  060464  001302  001116  001626  DT003:  .WORD   $TESTN,$ERRPC,DRVNUM,T.CS1,T.CS2,T.DS,T.ER,T.ASOF,T.MR1
13224  060472  001540  001550  001552
13225  060500  001554  001556  001566
13226  060506  001302  001116  001626  DT001:  .WORD   $TESTN,$ERRPC,DRVNUM
13227  060514                          DT002:
13228  060514                          DT004:
13229  060514                          DT005:
13230  060514                          DT006:
13231  060514                          DT007:
```

```
13232  060514  001302  001116  001626  DT010:  .WORD   $TESTN,$ERRPC,DRVNUM
13233  060522  001540  001550  001552          .WORD   T.CS1,T.CS2,T.DS,T.ER,T.ASOF
13234  060530  001554  001556
13235  060534  001560  001546                  .WORD   T.DCYL,T.DA
13236  060540  001544  001542                  .WORD   T.BA,T.WC
13237  060544  001202  001204  001206          .WORD   $REG10,$REG11,$REG12,$REG13,$REG14,$REG15,$REG16,$REG17
13238  060552  001210  001212  001214
13239  060560  001216  001220
13240  060564  001302  001116  001626  DT015:  .WORD   $TESTN,$ERRPC,DRVNUM
13241  060572  001202  001204  001206  DT015A: .WORD   $REG10,$REG11,$REG12
13242                                  .SBTTL  DATA FORMATS FOR ERROR REPORTS
13243  060600  000001                  DF001:  .WORD   1
13244  060602     003     000                  .BYTE   3,0
13245
13246  060604  000004                  DF002:  .WORD   4
13247  060606     000     000                  .BYTE   0,0
13248  060610  057162                          .WORD   DH002A
13249  060612     003     000                  .BYTE   3,0
13250  060614  057210                          .WORD   DH002B
13251  060616     007     000                  .BYTE   7,0
13252  060620  057275                          .WORD   DH002C
13253  060622     002     000                  .BYTE   2,0
13254
13255
13256  060624  000002                  DF003:  .WORD   2
13257  060626     003     000                  .BYTE   3,0
13258  060630  057406                          .WORD   DH003B
13259  060632     006     000                  .BYTE   6,0
13260
13261  060634  000006                  DF004:  .WORD   6
13262  060636     000     000                  .BYTE   0,0
13263  060640  000000                  DF004A: .WORD   0
13264  060642     000     000                  .BYTE   0,0
13265  060644  057162                          .WORD   DH002A
13266  060646     003     000                  .BYTE   3,0
13267  060650  057210                          .WORD   DH002B
13268  060652     007     000                  .BYTE   7,0
13269  060654  057275                          .WORD   DH002C
13270  060656     002     000                  .BYTE   2,0
13271  060660  057312                          .WORD   DH002D
13272  060662     010     000                  .BYTE   10,0
13273
13274  060664  000007                  DF005:  .WORD   7
13275  060666     000     000                  .BYTE   0,0
13276  060670  000000                  DF005A: .WORD   0
13277  060672     000     000                  .BYTE   0,0
13278  060674  057162                          .WORD   DH002A
13279  060676     003     000                  .BYTE   3,0
13280  060700  057210                          .WORD   DH002B
13281  060702     007     000                  .BYTE   7,0
13282  060704  057275                          .WORD   DH002C
13283  060706     002     000                  .BYTE   2,0
13284  060710  057312                          .WORD   DH002D
13285  060712     010     000                  .BYTE   10,0
13286  060714  057712                          .WORD   DH005A
13287  060716     000     000                  .BYTE   0,0
```

```
13288
13289  060720  000005              DF006:  .WORD   5
13290  060722  000     000                 .BYTE   0,0
13291  060724  000000              DF006A: .WORD   0
13292  060726  000     000                 .BYTE   0,0
13293  060730  057162                      .WORD   DH002A
13294  060732  003     000                 .BYTE   3,0
13295  060734  057210                      .WORD   DH002B
13296  060736  007     000                 .BYTE   7,0
13297  060740  057275                      .WORD   DH002C
13298  060742  002     000                 .BYTE   2,0
13299
13300  060744  000004              DF007:  .WORD   4
13301  060746  000     000                 .BYTE   0,0
13302  060750  000000              DF007A: .WORD   0
13303  060752  000     000                 .BYTE   0,0
13304  060754  057162                      .WORD   DH002A
13305  060756  003     000                 .BYTE   3,0
13306  060760  057406                      .WORD   DH003B
13307  060762  006     000                 .BYTE   6,0
13308
13309  060764  000004              DF010:  .WORD   4
13310  060766  000     000                 .BYTE   0,0
13311  060770  000000              DF010A: .WORD   0
13312  060772  000     000                 .BYTE   0,0
13313  060774  057162                      .WORD   DH002A
13314  060776  003     000                 .BYTE   3,0
13315  061000  060036                      .WORD   DH010A
13316  061002  002     000                 .BYTE   2,0
13317
13318  061004  000004              DF011:  .WORD   4
13319  061006  000     000                 .BYTE   0,0
13320  061010  000000              DF011A: .WORD   0
13321  061012  000     000                 .BYTE   0,0
13322  061014  057162                      .WORD   DH002A
13323  061016  003     000                 .BYTE   3,0
13324  061020  057406                      .WORD   DH003B
13325  061022  000006  000000              .WORD   6,0
13326
13327  061026  000002              DF015:  .WORD   2
13328  061030  003     000                 .BYTE   3,0
13329  061032  060251                      .WORD   DH015
13330  061034  003     000                 .BYTE   3,0
13331
13332  061036  000001              DF016:  .WORD   1
13333  061040  003     000                 .BYTE   3,0
13334
13335  061042  000052              BS26:   .BLKW   52
13336  061166  000052              BS24:   .BLKW   52
13337           061376                      .=61376
13338  061376  000240                      NOP
13339  061400  005014              SPCHLR: CLR     (R4)            ;CLEAR MEM MAINT REG
13340  061402  005237  001662              INC     INTSET          ;COUNT THE INTERRUPT
13341  061406  000240                      NOP
13342  061410  000002              SPCPAR: RTI                     ;RETURN
13343  061412  000240                      NOP
```

```
 13344  061414  001000                  IBUFF:  .BLKW   1000
 13345  063414  001000                  OBUFF:  .BLKW   1000
 13346          000001                  .END
```

```
ABASE = 177440          1955#   2277    2318
ABORT   053215          9529   12702#
ABTFAI  043650          9324   11293#
ACDW1 = 000000          2277    2320
ACDW2 = 000000          2277
ACLO  = 000010          2050#  10387
ACLOER= 000100         10389   12258#
ACPUOP= 000000          2277    2292
ADDW0 = 000000          2277
ADDW1 = 000000          2277
ADDW10= 000000          2277
ADDW11= 000000          2277
ADDW12= 000000          2277
ADDW13= 000000          2277
ADDW14= 000000          2277
ADDW15= 000000          2277
ADDW2 = 000000          2277
ADDW3 = 000000          2277
ADDW4 = 000000          2277
ADDW5 = 000000          2277
ADDW6 = 000000          2277
ADDW7 = 000000          2277
ADDW8 = 000000          2277
ADDW9 = 000000          2277
ADEVCT= 000000          2277    2283
ADEVM = 000000          2277    2319
ADJCLK  043274          3207   11186#
AENV  = 000000          2277    2288
AENVM = 000000          2277    2289
AFATAL= 000000          2277    2280
AMADR1= 000000          2277    2305
AMADR2= 000000          2277    2309
AMADR3= 000000          2277    2312
AMADR4= 000000          2277    2315
AMAMS1= 000000          2277    2299
AMAMS2= 000000          2277    2307
AMAMS3= 000000          2277    2310
AMAMS4= 000000          2277    2313
AMSGAD= 000000          2277    2285
AMSGLG= 000000          2277    2286
AMSGTY= 000000          2277    2279
AMTYP1= 000000          2277    2300
AMTYP2= 000000          2277    2308
AMTYP3= 000000          2277    2311
AMTYP4= 000000          2277    2314
APASS = 000000          2277    2282
APRIOR= 000240          1954#   2277
APTCSU= 000040          9382#  11326
APTEMV= 000001          9338    9380#   9437   11319
APTSIZ= 000200          2780    9379#
APTSPO= 000100          9340    9381#  11321
ASWREG= 000000          2277    2290
ATESTN= 000000          2277    2281
AUNIT = 000000          2277    2284
AUSWR = 000000          2277    2291
AVECT1= 000210          1953#   2277    2316
```

```
AVECT2= 000000          2277    2317
BAERR = 000004         10190   12291#
BAI   = 000020          2013#   4722    7464    7479    7564    7635    7691   10158
BA16  = 000400          1998#   4956    5042    5382    5435
BA17  = 001000          1999#   5102    5193    5382    5435
BOSRCK  041640         10761#  10930
BITCNT  035730          8718    9884#  10532   10542   10550   10562   10572   10583
BIT0  = 000001          1903#   1995    2028    2047    2689    3023    3047    3636    3873    4325    4513    4515    4517
                        5530    6573    7022    8828    9008   10042   12252   12270   12279   12289   12338
BIT00 = 000001          1893#   1903
BIT01 = 000002          1892#   1902
BIT02 = 000004          1891#   1901
BIT03 = 000010          1890#   1900
BIT04 = 000020          1889#   1899
BIT05 = 000040          1888#   1898
BIT06 = 000100          1887#   1897
BIT07 = 000200          1886#   1896
BIT08 = 000400          1885#   1895    9205
BIT09 = 001000          1884#   1894    9223    9448
BIT1  = 000002          1902#   2029    2690    3636    3875    4571    8839    9019   12253   12271   12280   12290   12339
BIT10 = 002000          1883#   2000    2019    2038    2071    2085    2098    2112    2126    2696    9402   11054   12262
BIT11 = 004000          1882#   2001    2020    2039    2056    2072    2086    2099    2113    2127    2697    9230   11136
                       12263
BIT12 = 010000          1881#   2003    2021    2040    2073    2087    2100    2114    2128   10973   10981   10999   11023
                       11027   11051   11161   12264
BIT13 = 020000          1880#   2004    2022    2041    2057    2074    2088    2101    2115    2129    9409    9560   10973
                       10981   10999   11023   11051   11161   12265
BIT14 = 040000          1879#   2005    2023    2042    2058    2075    2089    2102    2116    2130    4253    4254    6752
                        6753    9191   10903   10948   10964   11016   12266   12344
BIT15 = 100000          1878#   2006    2007    2024    2043    2059    2076    2135    2698    4250    4251    4403    7132
                        7270   10903   10925   10946   12267   12345
BIT2  = 000004          1901#   2030    2049    2691    5528    6571    9608   12254   12272   12281   12291   12340
BIT3  = 000010          1900#   2012    2031    2050    2692   12255   12273   12282   12292   12341
BIT4  = 000020          1899#   2013    2032    2051    2065    2106    2120    3123   12256   12274   12283   12293   12342
BIT5  = 000040          1898#   2014    2033    2052    2066    2080    2093    2107    2121    3125   12257   12275   12284
                       12294   12343
BIT6  = 000100          1897#   1996    2015    2034    2053    2067    2081    2094    2108    2122    2693    5561    5616
                        6615    6643    9647    9985   12258   12276   12285
BIT7  = 000200          1896#   1997    2016    2035    2054    2068    2082    2095    2109    2123    2694    2805    3010
                        3109    8550    9573   10880   12259   12286
BIT8  = 000400          1895#   1998    2017    2036    2055    2069    2083    2096    2110    2124    2695    3091    3162
                        3414    8557    8584    8653    8812    8822    8845    8853    8992    9002    9025    9033   10886
                       12260
BIT9  = 001000          1894#   1999    2018    2037    2070    2084    2097    2111    2125    4455    4456    4510    4511
                       10886   10901   10918   12261
BPTVEC= 000014          1910#
BSE   = 000200          2035#   7416    7574   10381   10428
BSERPT= 000200          2694#   7418    7420    7577    7579
BSERR = 000100          6784    6860   10430   12276#
BSF24P  001636          2671#   4065   10928
BSF26P  001640          2673#   3969   10924
BSS24P  001642          2675#   4071#  10952
BSS26P  001644          2677#   4056#  10949
BS24    061166          2671    4080   13336#
BS24CT  001650          2680#   4076#
BS26    061042          2673    4079   13335#
```

```
BS26CT  001646          2679#  4063*
CCLR  = 100000          2007#  3448   3502   7995   8271  10031  10058
CDT   = 002000          2000#  3358
CERNER= 100000         10326  12267#
CERR  = 100000          2006#  3037   3078   4168  10046  10102  10106  10324  10399
CFMT  = 010000          2003#  6409   6437   6495   6510   6582   6672   6687   8352   9941  10241  10898  10926
                       10950  10957
CHKBA   037110         10155#  10349
CHKCTS  037322          6788  10217#  10350
CHKOP   037642         10307#  11965
CHKST   037650         10305   10310#
CHKWC   037056         10136#  10347
CHKWE   037610         10298#  11966
CKSWR = 104407          9185   9398   9447  11955#
CLEAR = 000105          1980#  3536   7611   7670   7744   8098   8202   8241   8337   8471   8906   8925
CLKCAL  035102          9684#  9779
CMNDLB  047600          9790  10353  10645  12351#
COE   = 001000          2037#
COERR = 001000          6384  12261#
CP1170= 002000          2696#  5512   5526   5342   5565   5572   6564   6596   6619   6626   9592   9635
CR    = 000015          1818#  11365  11375
CRLF  = 000200          1819#  2804   11336  11375
CSRPTR  001672          2702#  5514*  5528*  5530*  5535   5567*  5569*  5574*  5576*  6571*  6573*  6589   6621*
                        6623*  6628*  6630*  9591*  9623*
CS1ERR= 024000         10316  12230#
CS2ERR= 177400         10318  12231#
CTO   = 004000          2001#  10048  10438
CTOERR= 000002          7785  10440  12280#
CTRHLT  043404          2423   2952   3107  11214#  11561  11644
CYLERR= 000010         10267  12292#
DCK   = 100000          2043#  10381  10409
DCKERR= 000002          8011  10411  12271#
DDISP = 177570          1825#  2224   2768
DDT   = 000400          2055#
DESHLD  001700          2709#  10966  10990*  11042*  11166
DF001   060600          2371  13243#
DF002   060604          2386  13246#
DF003   060624          2397  13256#
DF004   060634          2478  13261#
DF004A  060640         10577  10588  13263#
DF005   060664          2492  13274#
DF005A  060670         10597  13276#
DF006   060720          2517  13289#
DF006A  060724         10555  10567  13291#
DF007   060744          2543  13300#
DF007A  060750         10529  13302#
DF010   060764          2566  13309#
DF010A  060770          5398*  5451*  5586*  6794*  6801*  7101*  7108*  7119*  7125*  7239*  7246*  7257*  7263*
                       10353*  13311#
DF011   061004          2580   2589   2598   2606  13318#
DF011A  061010          3193*  3276*  3292*  3327*  3338*  3374*  3425*  3430*  3436*  3442*  3458*  3464*  3481*
                        3487*  3493*  3499*  3521*  3527*  3553*  3559*  3687*  3702*  3709*  3726*  3755*  3767*
                        3796*  3816*  3823*  4164*  4171*  9630*  8696*  8766*  8783*  8963*  10009*  13320#
DF015   061026          2618  13327#
DF016   061036          2626  13332#
DH001   057134          2369  13088#
```

```
DH002A  057162          2395    2616   13092#  13248   13265   13278   13293   13304   13313   13322
DH002B  057210         13096#  13250   13267   13280   13295
DH002C  057275         13105#  13252   13269   13282   13297
DH002D  057312         13108#  13271   13284
DH003B  057406         13118#  13258   13306   13324
DH005   057464         10638   13126#
DH005A  057712         13153#  13286
DH006   057561         10641   10698   13137#
DH007   057643         10636   13146#
DH010   057775          2564   13162#
DH010A  060036         13168#  13315
DH010B  060051         13170#
DH011   060121          2578   13177#
DH012   060150          2587   13181#
DH013   060201          2596   13186#
DH014   060224          2604   13190#
DH015   060251         13194#  13329
DH016   060302          7900   13199#
DH017   060377          8775   13210#
DH018   060422          8783   13214#
DH2N    001372          2384#   7900#   8775#   9790#   9957#   9995#
DH4N    001412          2476#  10578   10589
DH5N    001422          2490#  10598
DH6N    001432          2515#  10556   10568
DH7N    001442          2541#  10530
DI      = 040000        2005#   3427    3461    3484    3524    3556    3752    3793    3820    8913    8932   10102
DISPLA  001142          2224#   2768#   2776#   9245#   9401#
DISPRE  000174          2165#   2776
DLT     = 100000        2024#   4161   10416
DLTERR= 000010         10418   12273#
DMD     = 000040        2066#   3589    3597    3636    3645    8072    8154    8162    8262    8385   11197
DRA     = 000001        2047#
DRDY    = 000200        2054#   3051    3056    3080    8523
DROT    = 000040        2052#  10395
DROTER= 000400          8290    8424   10397   12260#
DRPAR   = 000010        2031#
DRPARE= 000010          3409   12255#
DRSTER= 040000         10593   10599   10733   12344#
DRVABT  057006          4089   13071#
DRVBIT  001630          2667#   3132#   3197   3199#   3998   3999   4048   4049   8550   8554#   8555   8557   8563#
DRVDRP  001666          2700#   2881#   3023#   3047#   3084#   3134   3164   3999#   4049#
DRVMSK= 000007          2011#
DRVNUM  001626          2666#   3141#   3203#   3223   3263   3313   3330   3356   3400   3416   3578   3626   3676
                        3742    3764    3848   3865   3909   3926   6410   6438   6898   7410   8521   8553#   8564#
                        8741#   8753#   8774#   8894#   8915#   8935#   8944#   8959#   9823   9949  11189  13223  13226
                       13232   13240
DRVRPT= 004000          2697#   2882    3014   3101   3119
DSC     = 040000        2058#   3190    3433   3490   3706
DSERMI= 000070         10320   12234#
DSWR    = 177570        1824#   2223    2767
DTE     = 010000        2040#
DTERR   = 010000        8083   12264#
DTYE    = 000040        2033#
DTYERR= 000040         12257#
DT002   060506          2370   13226#
DT002   060514          2385   13227#
```

```
DT003   060464          2396   13223#
DT004   060514          2477   13228#
DT005   060514          2491   13229#
DT006   060514          2516   13230#
DT007   060514          2542   13231#
DT010   060514          2579    2588    2597    2605   13232#
DT015   060564          2565    2617   13240#
DT015A  060572          2625   13241#
DOTST = 000001          2689#   3024    3046    3058    3064
ECCW  = 020000          2074#
ECH   = 000100          2034#  10381   10405
ECHERR= 000001          8011   10407   12270#
EMCCLR  056546          3458    3464   13034#
EMCERR  056721          4170   13061#
EMDA    056526          3424    3457    3480    3520    3552    8629    8695    8962   13031#
EMDCLR= 047666          3553    3559   13038#
EMDI    056462          3429    3463    3486    3526    3558    3754    3795    3822   13023#
EMDLT = 054527          4163   10008   13056#
EMDPAR= 054753          3425    3430    3436    3442    3481    3487    3493    3499   13026#
EMDSC   056502          3192    3435    3492    3708   13027#
EMDTPE= 055036          3374   13022#
EMMI    056762          3766   13067#
EMNZPR  056357          3292    3521    3527   13010#
EMRCAL= 047727          3687   13039#
EMRDB   056675          4164    4171   10009   13057#
EMRSET  056667          3816    3823   13054#
EMSCLR  056742          3193    8696   13064#
EMSELD= 047640          3276    3327    3726   13037#
EMSK  = 047752          8963   13046#
EMSKSF  056610          3755    3767    3796   13043#
EMSVAL  056342          3275    3291    3326    3337    3373    3441    3498    3686    3701    3725    8765    8782   13007#
EMTVEC= 000030          1913#   2751*   2752*
EMUR    056652          8630   13051#
EMUXIT  056625          3815   13047#
EMWCS2  056420          3338   13016#
EM1     053303          2921    2950   12714#
EM1N    001360          2368#   2921*   2950*   3996*   4007*   4017*   4024*   4033*   4046*   4089*   7421*   7580*
EM10    053774          5415    5469    5601    7107    7124    7245    7262   12296   12770#
EM10N   001450          2563#   5397*   5408*   5415*   5450*   5462*   5469*   5587*   5601*   6791*   6800*   7098*   7107*
                        7116*   7124*   7236*   7245*   7254*   7262*  10360*
EM11    054021          5408    5462    5587    7098    7116    7236    7254   12298   12774#
EM11A   054047          5397    5450   12297   12778#
EM11N   001460          2577#   3275*   3326*   3373*   3424*   3429*   3435*   3441*   3480*   3486*   3492*   3498*   3686*
                        3708*   3725*   3754*   3795*   8765*   8962*
EM12    054123         12299   12786#
EM12N   001470          2586#   3192*   3291*   3337*   3552*   3558*   3701*   3822*   8629*   8695*
EM13    054156          4163*   6791   12300   12791#
EM13N   001500          2595#   3766*   3815*   4170*  10008*
EM14    054206          6800   12301   12795#
EM14N   001510          2603#   3457*   3463*   3526*   8782*
EM15    054240         12303   12800#
EM15N   001520          2615#  10879*  10882*
EM16    054263         12304   12804#
EM17    054306         12305   12808#
EM18    054327         12306   12811#
EM19    054354         12307   12815#
```

```
EM2     053363          12723#
EM2INT  056567           3702    3709   13040#
EM2N    001370           2383#   3695*   9791*   9966*   9994*
EM20    054376          12308   12818#
EM21    054422          12309   12822#
EM22    054446          12310   12826#
EM23    054467          12312   12829#
EM24    054500          12313   12831#
EM25    054513          12314   12833#
EM26    054527          12315   12836#  13056
EM27    054541          12316   12838#
EM28    054566          12317   12842#
EM29    054601          12318   12844#
EM3     053444           9562   12732#
EM3N    001400           2394#   3158*   3172*   3186*   3243*  10108*  10110*  10121*
EM30    054622          12320   12847#
EM31    054675          12321   12855#
EM32    054715          12322   12858#
EM33    054753          12323   12864#  13026
EM34    055021          12324   12871#
EM35    055036          12325   12874#  13022
EM36    055057          12326   12877#
EM37    055066          12327   12879#
EM38    055111          12328   12883#
EM39    055131          12329   12886#
EM4     053514           9791    9966   12739#
EM4N    001410           2475#   3520*
EM40    055153          12330   12890#
EM41    055205          12331   12895#
EM42    055226          12332   12898#
EM43    055251          12333   12902#
EM44    055313          12334   12908#
EM45    055330          12335   12911#
EM46    055370           3186   12918#
EM47    055425           3158   12923#
EM48    055506           3172   12932#
EM49    055565           3243   12940#
EM5     053607          10108   12749#
EM5N    001420           2489#
EM50    055635           3695   12947#
EM51    055734           3996    4007    4017    4024   12958#
EM52    056027           4033   12969#
EM53    056125           4046   12981#
EM54    056210          10879   12990#
EM55    056237          10882   12994#
EM56    056264           9994   12999#
EM6     053653          10110   12756#
EM6N    001430           2514#
EM7     053732          10121   12764#
EM7N    001440           2540#
ERRCNT  001632           2668#   9524*   9527
ERRLMT  001634           2670#   4193    4689    4762    4843    4912    4993    5062    5142    5213    5287    5358    5694
                         5782    5865    5950    6035    6094    6151    6209    6265    6321    6455    6530    6707    6832
                         6870    7507
ERRVEC= 000004           1906#   2765    2766*   2777*   9113    9114    9118*   9123*   9135*   9140*   9149*   9159*   9160*
                         9196    9197*   9199*   9202*   9582    9583    9584*   9585*   9619*   9642*   9657*   9658*
```

```
EXPBA   047430          10160*  10187*  12239*
EXPCYL  047432          10246*  10260*  10265   12240*
EXPSEC  047434           6802   10264*  12241*
EXPTRK  047436           6792   10263*  12242*
EXPUBA  047426          10186*  10194   12238*
EXPWC   047424          10136*  10138   10361   12237*
FNTE  = 000020           2032*
FNTERR= 000020          12256*
FPFMT = 000400           2695*   7350
GENCOM  041740           4120    4188    4196    4246    4321    4400    4452    4506    4567    4628    4657    4683    4692
                         4723    4757    4765    4808    4821    4837    4846    4883    4906    4915    4950    4969    4986
                         4996    5031    5055    5065    5108    5135    5145    5182    5206    5216    5245    5265    5281
                         5290    5319    5337    5352    5361    5516    5643    5654    5688    5697    5727    5750    5776
                         5785    5813    5829    5859    5868    5898    5910    5944    5953    5983    5995    6029    6038
                         6065    6069    6088    6097    6123    6127    6145    6154    6180    6184    6203    6212    6236
                         6240    6259    6268    6292    6296    6315    6324    6416    6450    6458    6486    6490    6524
                         6533    6560    6663    6667    6701    6710    6749    6762    6766    6826    6835    6842    6864
                         6873    6905    6977    7026    7162    7291    7364    7449    7501    7510    7957    7976    8485
                         8529   10873*  11263
GETDRS  036442          10030*  10594
GETRK   035652           9860*  11964
GNS   = ****** U         2164    2803    9067    9074   11947   11948   11949   11950   11951   11953   11955   11956   11957
                        11958   11959   11960   11961   11962   11963   11964   11965   11966   11967   11968   11969   11970
                        11971   11972   11973   11974   11975   11976   11977   11978   11979   11980
GO    = 000001           1995*
GPSUMF  047576          10315*  10528   10699*  12337*
GP1NR = 000010          10586   10591   10639   10693   10730   12341*
GP2NR = 000020          10565   10639   10693   10726   12342*
GP3NR = 000040          10545   10639   10693   12343*
GRP1ER  047454          10311*  10326*  10373   10479   10571   10681   12251*
GRP1MS  047536          10682   10713   12320*
GRP1ST= 000001          10575   10591   10647   10695   10730   12338*
GRP2ER  047456          10312*  10403   10549   10675   12269*
GRP2MS  047520          10676   10707   12312*
GRP2ST= 000002          10553   10647   10695   10726   12339*
GRP3ER  047460          10313*  10432   10531   10652   12278*
GRP3MS  047500          10653   10702   12303*
GRP3ST= 000004          10536   10647   10695   12340*
GRP4ER  047462           6787*   6789    6798   10140*  10190*  10196*  10267*  10270*  10273*  10314*  10351   10354   12288*
GRP4MS  047464          10360   12296*
GTSWR = 104406           2798   11953*
HT    = 000011           1816*  11334   11375
HVRC  = 000400           2036*  10381   10424
HVRCER= 000040          10426   12275*
IBUFF   061414           3977    4031    4037    4139    4671    4737    4828    4976    5123    5260    5333    5661    5676
                         5745    5764    5836    5850    5917    5932    6002    6017    6076    6133    6191    6247    6303
                         6347    6374    6512    6689    6814    6849    6931    7487    7922    8000    9944    9998   10961
                        11022   13344*
IDAE  = 002000           2038*
IDAERR= 002000           3879    3887    3935   12262*
IE    = 000100           1996*   2940    3809    7790    7893    8091    8463
ILF   = 000001           2028*  10381   10462
ILFERR= 000200          10464   12286*
INTHLR  034354           2887    3007    5614    6641    9552*
INTR  = 000300           1991*
INTSET  001662           2685*   2914*   2919    2939*   2944    3682*   3764    3800*   3855*   3916*   5537    6591    7409*
```

```
                    7444*  7599*  7659*  7737*  7789*  7826*  7895*  7990   8049*  8092*  8196*  8232*  8329*
                    8456*  8595*  8663*  8750*  8769*  8901*  8922*  9543*  9553*  9643*  9651   9656*  9689
                    9712   9780   9835*  9954*  9958   11238* 13340*
IOTVEC= 000020       1911*  2749*  2750*
IR    = 000100       2015*
IWAT1M  035224       9736*  11980
IWAT1S  035244       9742*  11977
IWAT11  035304       9754*  11973
IWAT12  035274       9751*  11974
IWAT14  035264       9748*  11975
IWAT15  035254       9745*  11976
IWAT16  035364       9772*  11967
IWAT2S  035234       9739*  11978
IWAT32  035354       9769*  11968
IWAT4S  035344       9766*  11969
IWAT64  035334       9763*  11970
IWAT8S  035214       9733*  11979
IWAT80  035324       9760*  11971
IWAT9S  035314       9757*  11972
KIPAR0= 172340       1944*   9099   9127
KIPAR1= 172342       1945*
KIPAR2= 172344       1946*
KIPAR3= 172346       1947*
KIPAR4= 172350       1948*
KIPAR5= 172352       1949*
KIPAR6= 172354       1950*  11026*
KIPAR7= 172356       1951*   9143
KIPDR0= 172300       1933*
KIPDR1= 172302       1934*
KIPDR2= 172304       1935*
KIPDR3= 172306       1936*
KIPDR4= 172310       1937*
KIPDR5= 172312       1938*
KIPDR6= 172314       1939*
KIPDR7= 172316       1940*
KWLADD  001710       2713*  5545*  5561*  5616*  6599*  6615*  6643*  9573*  9647*  9939*  9985*  11195*
KWLVEC  001712       2714*  9644
LCKHLR  034420       9572*  9645
LCLKPR= 100000       2698*  9654   9685   9782   9937   9983   11193
LCLKTK  001674       2704*  3804*  3806   9572*  9684*  9691   9693*  9718*  9785   9955*  9961
LF    = 000012       1817*  11369  11375
LOADRK  035570       9835*  11963
LRLOAD  035530       3974   4112   4140   4237   4265   4313   4346   4392   4412   4444   4465   4498   4526
                     4559   4585   4620   4649   4668   4714   4734   4801   4825   4876   4893   4955   4973
                     5024   5041   5101   5120   5175   5192   5237   5256   5312   5330   5381   5434   5504
                     5635   5658   5673   5719   5742   5761   5805   5833   5847   5890   5914   5929   5975
                     5999   6014   6073   6130   6188   6244   6300   6344   6371   6494   6509   6581   6671
                     6686   6741   6770   6811   6846   6928   6969   6988   7030   7045   7065   7166   7181
                     7203   7295   7312   7356   7390   7433   7456   7484   7557   7588   7610   7625   7649
                     7669   7684   7722   7743   7758   7770   7814   7833   7848   7871   7919   7961   7980
                     7997   8038   8053   8065   8097   8129   8147   8185   8201   8216   8240   8255   8276
                     8314   8336   8351   8366   8378   8409   8432   8470   8488   8532   9815*  11223  11252
L.ASOF  001612       2655*  8223*
L.BA    001604       2650*  6413*  6901*  8948*  10160  10175
L.CS1   001600       2648*  3029*  3071*  3142*  3175*  3224*  3264*  3314*  3357*  3358*  3401*  3448*  3467*
                     3502*  3508*  3536*  3579*  3627*  3677*  3713*  3743*  3785*  3847*  3866*  3908*  3927*
```

```
                    6409*  6437*  6899*  8589*  8658*  8742*  8754*  8896*  8906*  8917*  8925*  8938*  8949*
                    9818   9850   9941   9956   10094  10164  10241  10898  10926  10950  10957
L.CS2   001610      2654*  3027*  3070*  3140*  3223*  3233*  3263*  3313*  3356*  3400*  3578*  3588*  3626*
                    3637*  3676*  3742*  3784*  3848*  3865*  3909*  3926*  4722*  6410*  6438*  6898*  7464*
                    7479*  7564*  7635*  7691*  8155*  8588*  8657*  8740*  8752*  8895*  8916*  8936*  8945*
                   10040  10158
L.DA    001606      2651*  9951
L.DCYL  001614      2656*  3928*  6414*  6440*  6903*  8937*  8946*  9950   10246  10767  10890  11262*
L.DS    001606      2652*  6936*  10249
L.DT    001607      2653*  3867*  4204   6412*  6439*  6900*  10247  10770  10889  11273  11275*
L.MR1   001616      2657*  3265*  3279*  3402*  3446*  3510*  3534*  3589*  3636*  7840*  7879*  8072*  8154*
                    8262*  8283*  8385*  8417*  8440*  8468*
L.WC    001602      2649*  6411*  6902*  8947*  9838   10162  10213
MCLK  = 000400      2069*  9915   9916
MCLOCK  036004      3593   3641   8075   8159   8266   8388   8393   8398   9906*
MDS   = 001000      2018*  10446
MDSERR= 000010      10448  12282*
MEMBAS  001714      2715*  9596
MEMPAR  001670      2701*  9586*  9600*  9633
MEMPYB= 000004      2691*  5497   5500
MEMSZB= 000002      2690*  4788   4793   4936   4942   5087   5093
MERD  = 001000      2070*
MESMSK= 000017      2063*
MEMD  = 002000      2071*
MILCNT  001652      2681*  9711
MIND  = 000200      2068*  8391   8396
MMVEC = 000250      1922*
MMVECA  001716      2716*  5539*  5615*  6593*  6642*  9603   9632
MSP   = 000100      2067*
MYTIME  035150      3805   9710*  9784   9960
M.CADD= 017760      2138*
M.CDIF= 017760      2137*
M.DRV = 000007      2134*
M.HEAD= 007000      2141*
M.ID  = 000003      2136*
M.PAR = 100000      2135*
M.SECT= 000760      2140*
M.SER = 077770      2139*
NCERME= 020000      10401  12265*
NED   = 010000      2021*  3039   3095   10434
NEDERR= 000001      10436  12279*
NEM   = 004000      2020*  10454
NEMERR= 000040      5395   5448   10456  12284*
NEMORV  027764      4000   4050   8548*
NEXINT  034346      2911   9543*  9612
NXF   = 000004      2030*
NXFERR= 000004      12254*
OBUFF   063414      4115   4240   4249   4268   4316   4324   4349   4395   4403*  4415   4447   4455*  4456*
                    4468   4501   4509   4529   4562   4570   4588   4623   4652   4717   4743   4744   4879
                    5027   5178   5240   5276   5315   5347   5507   5525*  5529*  5568*  5575*  5588   5590
                    5595   5597   5638   5722   5808   5824   5893   5978   6413   6497   6572*  6584   6622*
                    6629*  6674   6744   6752*  6753*  6773   6901   6972   6991   7033   7048   7062*  7068
                    7095   7099   7114   7117   7169   7184   7199*  7206   7233   7237   7252   7255   7298
                    7310*  7315   7359   7393   7407   7459   7555*  7560   7628   7633*  7687   7851   7964
                    7983   8258   8381   8491   8535   8948   9606   10904  10941*  10942*  10962  11021  11255
                   13345*
```

```
OFFSET= 000115          1984#   8217
OFST  = 000004          2049#
OPER00  047640          8766   12351   12369#  13037
OPER02  047655         12352   12372#
OPER04  047666         12353   12374#  13038
OPER06  047702         12354   12376#
OPER10  047711         12355   12378#
OPER12  047727         12356   12381#  13039
OPER14  047743         12357   12384#
OPER16  047752         12358   12386#  13046
OPER20  047757         12359   12387#
OPER22  047771         12360   12389#
OPER24  050004          9967   12361   12391#
OPER26  050021         12362   12394#
OPER30  050037         12363   12397#
OPER32  050053         12364   12400#
OPER34  050077         12365   12404#
OPER36  050123         12366   12408#
OPER37  050147          6794    6801   12412#
OPER40  050203         12417#
OPER41  050203          7101    7108    7119    7125    7239    7246    7257    7263   12418#
OPER42  050240          5398   12423#
OPER43  050314          5451   12431#
OPER44  050367          5586   12439#
OPI   = 020000          2041#  10381   10420
OPIERR= 000020         10422   12274#
OPR001  050446          2834   12451#
OPR002  050475          2837    2848    2865   12455#
OPR003  050503          2844   12457#
OPR004  050533          2856   12462#
OPR005  050555          2810   12466#
OPR006  050626          3026   12473#
OPR007  050776          3022    3045   12491#
OPR008  051141          3106   12508#
OPR009  051224          3111   12517#
OPR010  051274          5499   12524#
OPR011  051471          4791    4939    5090   12547#
OPR012  051634          4792   12565#
OPR013  051640          4940   12566#
OPR014  051644          5091   12567#
OPR015  051650          4794    4941    5092   12568#
OPR016  051664          2829   12570#
OPR017  052445          7421    7580   12638#
OPR018  052562          8724   12652#
OPR019  052650         11186   12661#
OPR020  053013         11250   12678#
OPR021  053107         11278   12689#
OPR022  053150         11294   12695#
OPTFLG  001664          2687#  2827    2830*   2882*   3014    3024*   3046*   3058*   3064    3101    3119*   4788    4793*
                        4936   4942*   5087    5093*   5495    5497    5500*   5512    5526    5542    5565    5572    6557
                        6564   6596    6619    6626    7350    7418    7420*   7577    7579*   9587*   9592*   9593*   9601
                        9624*  9635    9654*   9685    9782    9937    9983   11193
OPTTST  034434          2883   3810    6659    8622    9581#
OR    = 000200          2016#
PACK  = 000103          1979#  3175
PARM    001730          2171   2726#
```

```
PARPRE= 000100          2693#   5495    6557    9587    9593    9601    9624
PAT   = 000020          2065#   3402    7840
PAT02   046624         11076   11988#
PAT03   046664         11081   12008#
PAT04   046724         11086   12028#
PAT05   046764         11091   12048#
PAT06   047024         11096   12068#
PAT10   047064         11101   12091#
PAT11   047124         11106   12111#
PAT12   047164         11111   12131#
PAT13   047224         11116   12151#
PAT14   047264         11121   12171#
PAT15   047324         11126   12191#
PAT16   047364         11131   11134   12211#
PCA   = 004000          2072#
PCD   = 010000          2073#
PERHLR  034364          5615    6642    9560#   9637
PGE   = 002000          2019#  10450
PGERR = 000020          7937   10452   12283#
PIP   = 020000          2057#
PIRQ  = 177772          1823#
PIRQVE= 000240          1917#
PROGEN  043604         11222   11249   11274   11278#  11295
PR0   = 000000          1840#   2891    3002    9659    9774
PR1   = 000040          1841#
PR2   = 000100          1842#
PR3   = 000140          1843#
PR4   = 000200          1844#
PR5   = 000240          1845#   1954
PR6   = 000300          1846#
PR7   = 000340          1847#   2735    2912    3008    5520    6566    9585    9605    9638    9646
PS    = 177776          1820#   1821
PSW   = 177776          1821#
PWRVEC= 000024          1912#   2755*   2756*  11877*  11878*  11887*  11893*  11905*  11906*
RDCHR = 104410         11720   11956#
RDDATA= 000121          1986#   3975    4669    4735    4826    4894    4974    5042    5121    5193    5258    5331    5435
                        5659    5674    5743    5762    5834    5848    5915    5930    6000    6015    6074    6131    6189
                        6245    6301    6345    6372    6510    6687    6812    6847    6929    7485    7920    7998
RDGATE= 100000          2076#
RDHEAD= 000125          1988#   4141    6437    7771    8066    8352
RDLIN = 104411         11794   11957#
RDOCT = 104412          2838    2849    2866   11958#
RDSTHD  036050          4181    6442    9936#
RDY   = 000200          1997#   2940   10119   11191
REALBA  047444         10191*  12246#
REALCY  047446         12247#
REALSE  047452          6803   12249#
REALTR  047450          6793   12248#
REALUB  047442         10197*  12245#
REALWC  047440         10141*  10274   10362   12244#
RECAL = 000113          1983#   3677    3847    3908    7759    7815    8054    8186    8315    8589    8658    8742    8896
                        8917   11224
REFMT   001676          2705#   2740*   4106*   4233*   4309*   4388*   4440*   4494*   4554*   4613*   4697*   6403*   6482*
                        6556*   6658*   6737*   6952*   8310*   8546*  11248   11260
REPNR = 100000         10699   12345#
RESREG= 104414          9523    9663    9798   10017   10067   10281   10363   10367   10721   10743   11960#
```

```
RESTRT  001740          2169    2729#
RESVEC= 000010          1907#
RKASOF= 000016          1966#   8598    8666
RKBA   = 000004         1961#   7407#
RKCS1  = 000000         1959#   2915#   2940#   3809#   7411#   7790#   7893#   7995#   8091#   8271#   8463#   8522#   8597#
                        8665#   9956#   9970    10031#  10042#  10046   10048   10058#  10101   11190#  11191
RKCS2  = 000010         1963#   2937#   3330#   7410#   8521#   8685#   9949#   9977    10040#  10050   10099#  11188#  11189#
RKDA   = 000006         1962#   9951#
RKDB   = 000024         1968#   4138    9948
RKDCYL= 000020          1967#   9950#
RKDS   = 000012         1964#   8523
RKECPS= 000030          1972#
RKECPT= 000032          1973#
RKER   = 000014         1965#   7416    7574
RKMR1  = 000026         1969#   3597#   3645#   8078#   8162#   8269#   8391#   8396#   8401#   9915#   9916#   10041#  11197#
                       11201#
RKMR2  = 000034         1970#   10056
RKMR3  = 000036         1971#   10057
RPRI   001622           2661#   2888#   5532    6575
RKSPAR= 000022          1974#   7929#
RKVEC   001620          2660#   2886#   3006    5538#   5614#   6592#   6641#
RKWC   = 000002         1960#   7408#   7992
RLS    = 000010         2012#   3233    3588
SAVREG= 104413          9470    9581    9733    9736    9739    9742    9745    9748    9751    9754    9757    9760    9763
                        9766    9769    9772    9936    10030   10217   10298   10307   11959#
SCLR   = 000040         2014#   8685    10099   11188
SCOP1  = 104415         3883    3891    3987    4134    4166    4173    4202    4286    4367    4819    6426    6449    6465
                        6806    6915    6945    7060    7083    7130    7196    7221    7268    7405    7428    7474    7499
                        7517    8786    8966    11961#
SCOP1$  035056          9672#   11961
SECERR= 000040          6798    10273   12294#
SEEK   = 000117         1985#   3743    3785    3866    3927    7434    7589    7650    7723    7834    8039    8938
SELDRV= 000101          1978#   3029    3071    3142    3224    3264    3314    3357    3401    3467    3508    3579    3627
                        3713    7872    8130    8148    8277    8367    8410    8433    8754
SETCLK  001720          2173    2723#
SKI    = 000002         2029#
SKIERR= 000002          3887    7861    12253#
SPACE2  050443          3126    9506    9518    12450#
SPAR   = 020000         2004#   10048   10383
SPARER= 000001          3409    10385   12252#
SPCHLR  061400          5538    6592    13339#
SPCPAR  061410          5539    6593    13342#
SPDERR= 000200          10393   12259#
SPOLSS= 000020          2051#   10391
SRCHLD  001702          2710#   10967   10991#  11043#  11167
SRTFLG  001624          2662#   2723#   2726#   2729#   2732#   2832    3020    3205
SRTINS= 000010          2692#   2827    2830
SRTSPL= 000111          1982#
SR0    = 177572         1926#   9124    9139#   9146#   10975#  10983#  11001#  11053#  11163#
SR1    = 177574         1927#
SR2    = 177576         1928#
SR3    = 172516         1929#   9136#
SSINIT  036656          10090#  11962
STACK  = 001100         1811#   2734    2747    2906    9533    9565    11279
START   001750          2162    2168    2732#
START1  001754          2724    2727    2730    2733#   11288
```

```
STKLMT= 177774           1822#
SVAL  = 100000           2059#  3156   3170   3241   3272   3288   3323   3334   3370   3439   3496   3684   3699
                         3723   8763   8779
SWR     001140           2223#  2745   2767*  2769   2775*  2782*  2796   9191   9205   9208   9223   9230   9402
                         9409   9444   9448   9525   9560   9672   10628  11564  11603  11658* 11885  11898*
SWREG   000176           2166#  2775   2796   11564  11603  11626
SW0   = 000001           1875#
SW00  = 000001           1865#  1875
SW01  = 000002           1864#  1874
SW02  = 000004           1863#  1873
SW03  = 000010           1862#  1872
SW04  = 000020           1861#  1871
SW05  = 000040           1860#  1870
SW06  = 000100           1859#  1869
SW07  = 000200           1858#  1868
SW08  = 000400           1857#  1867
SW09  = 001000           1856#  1866
SW1   = 000002           1874#
SW10  = 002000           1855#
SW11  = 004000           1854#
SW12  = 010000           1853#  9525
SW13  = 020000           1852#  10628
SW14  = 040000           1851#
SW15  = 100000           1850#
SW2   = 000004           1873#
SW3   = 000010           1872#
SW4   = 000020           1871#
SW5   = 000040           1870#
SW6   = 000100           1869#
SW7   = 000200           1868#
SW8   = 000400           1867#
SW9   = 001000           1866#  9672
S.ACLO= 000100           2094#
S.BRHM= 000100           2108#
S.BRKE= 040000           2130#
S.CART= 000400           2110#
S.DIB = 002000           2126#
S.DOOR= 000200           2109#
S.DRA = 000040           2080#
S.DROT= 020000           2101#
S.DRY = 000200           2082#
S.DSC = 040000           2089#
S.FLT = 000200           2095#
S.FORM= 001000           2084#
S.FWD = 002000           2112#
S.HDFL= 000200           2123#
S.HDHM= 000040           2107#  7887   8450
S.ICYL= 000040           2093#
S.ILF = 000400           2096#
S.LIMD= 020000           2129#
S.LOAD= 010000           2114#
S.MHD = 000400           2124#
S.NMOV= 010000           2128#
S.OFF = 002000           2085#
S.PAR = 001000           2097#
S.PIP = 020000           2088#
```

```
S.PLO = 004000          2127#
S.REV = 004000          2113#
S.RTZ = 020000          2115#
S.SECT= 000020          2120#
S.SKI = 002000          2098#
S.SPIN= 010000          2087#
S.SPLS= 010000          2100#
S.SPOK= 001000          2111#
S.TYPE= 000400          2083#
S.UNLD= 040000          2116#
S.UNS = 040000          2102#
S.VV  = 000100          2081#
S.WCLK= 000040          2121#
S.WGAT= 000100          2122#
S.WLE = 004000          2099#
S.WRL = 004000          2086#
S.XDOK= 000020          2106#
S.XERR= 001000          2125#
TBITVE= 000014          1908#
TCHKOP= 104421          3042    3098    3167    3181    3230    3239    3269    3285    3320    3452    3515    3542    3585
                        3633    3719    3770    3985    4131    4156    4182    4260    4293    4339    4373    4409    4462
                        4523    4582    4596    4635    4665    4680    4731    4754    4817    4835    4891    4904    4967
                        4984    5039    5053    5117    5132    5190    5204    5253    5273    5327    5344    5651    5670
                        5685    5735    5758    5773    5821    5845    5906    5926    5941    5992    6011    6026    6085
                        6142    6200    6256    6312    6356    6423    6443    6506    6521    6634    6683    6698    6759
                        6823    6912    6942    6985    7000    7042    7057    7178    7193    7307    7324    7371    7402
                        7426    7445    7471    7496    7585    7601    7607    7622    7647    7661    7666    7681    7704
                        7734    7740    7755    7830    7884    7973    8017    8108    8141    8167    8213    8229    8237
                        8252    8326    8333    8348    8363    8406    8447    8482    8500    8544    8612    8680    8747
                        8760    8911    8930    8956    11235   11242   11270   11965#
TCHKWE= 104422          3150    3364    3408    3473    3602    3650    3878    3886    3934    4280    4361    4424    4477
                        4538    5392    5445    5579    6383    6782    6858    7077    7215    7782    7860    7934    8009
                        8082    8289    8423    11966#
TGETRK= 104420          3036    3077    3332    3549    3683    3698    3750    3791    3819    4160    8624    8690    8778
                        9787    9965    10011   10111   10310   11964#
TIMCNT  001654          2682#   9687*   9716*   9719*   9778*   9953*
TKVEC = 000060          1915#   11541*  11542*
TLOADR= 104417          3031    3072    3144    3176    3226    3235    3266    3281    3316    3360    3404    3449    3469
                        3504    3511    3538    3581    3591    3629    3639    3679    3715    3745    3787    3850    3869
                        3911    3930    3982    4127    4152    4256    4273    4336    4354    4405    4420    4458    4473
                        4519    4534    4578    4592    4631    4661    4676    4727    4750    4813    4832    4887    4901
                        4963    4980    5035    5049    5113    5128    5186    5200    5249    5269    5323    5341    5389
                        5442    5541    5647    5666    5681    5731    5754    5769    5817    5841    5855    5902    5922
                        5937    5987    6007    6022    6081    6138    6196    6252    6308    6352    6379    6419    6502
                        6517    6595    6679    6694    6755    6778    6819    6854    6908    6938    6981    6996    7038
                        7053    7073    7174    7189    7211    7303    7320    7367    7398    7441    7466    7492    7565
                        7596    7618    7637    7656    7677    7694    7730    7751    7766    7778    7822    7841    7856
                        7880    7927    7969    7988    8005    8046    8061    8073    8105    8137    8157    8193    8209
                        8225    8248    8264    8285    8322    8344    8359    8374    8387    8419    8443    8478    8496
                        8540    8591    8659    8744    8756    8898    8907    8919    8926    8940    8951    11231   11266
                        11963#
TPVEC = 000064          1916#
TRAPVE= 000034          1914#   2753*   2754*
TRKERR= 000020          6789    10270   12293#
TRTVEC= 000014          1909#
TSSINI= 104416          2999    3067    3137    3220    3260    3310    3353    3397    3575    3608    3623    3673    3739
```

CROSS REFERENCE TABLE -- USER SYMBOLS

		3781	3844	3862	3905	3923	3971	4109	4125	4150	4179	4234	4263	4310
		4343	4389	4441	4495	4556	4614	4711	4798	4874	4948	5022	5099	5172
		5234	5309	5378	5431	5492	5632	5716	5802	5887	5972	6059	6118	6175
		6232	6288	6341	6368	6407	6435	6483	6579	6660	6738	6809	6895	6925
		6966	7024	7063	7160	7200	7288	7388	7454	7482	7552	7719	7787	7811
		7845	7866	7890	7916	7954	8035	8089	8126	8144	8182	8311	8429	8453
		8460	8518	8526	8578	8647	8737	8891	11216	11245	11962#			
TSTLUP	004570	3247#	8562											
TST1	003144	2904#	9090	9250										
TST10	005200	3395#	9257											
TST100	026036	7952#	9313											
TST101	026230	8032#	9314											
TST102	026454	8124#	9315											
TST103	026602	8180#	9316											
TST104	027124	8308#	9317											
TST105	027656	8515#	9318											
TST106	030060	8573#	9319											
TST107	030300	8642#	9320											
TST11	006124	3573#	9258											
TST110	030514	8715#	9321											
TST111	031212	8879#	9322											
TST12	006242	3621#	9259											
TST13	006352	3671#	9260											
TST14	006614	3737#	9261											
TST15	006746	3779#	9262											
TST16	007156	3840#	9263											
TST17	007346	3903#	9264											
TST2	003262	2935#	9251											
TST20	007470	3964#	9265											
TST21	010260	4104#	9266											
TST22	010656	4231#	9267											
TST23	011070	4307#	9268											
TST24	011306	4386#	9269											
TST25	011432	4438#	9270											
TST26	011564	4492#	9271											
TST27	011742	4552#	9272											
TST3	003350	2997#	9252											
TST30	012074	4611#	9273											
TST31	012370	4708#	9274											
TST32	012564	4783#	9275											
TST33	013020	4864#	9276											
TST34	013206	4932#	9277											
TST35	013440	5013#	9278											
TST36	013626	5083#	9279											
TST37	014050	5163#	9280											
TST4	004466	3218#	9253											
TST40	014236	5232#	9281											
TST41	014430	5307#	9282											
TST42	014624	5376#	9283											
TST43	015030	5429#	9284											
TST44	015234	5490#	9285											
TST45	016100	5630#	9286											
TST46	016304	5714#	9287											
TST47	016510	5800#	9288											
TST5	004570	3258#	9254											
TST50	016726	5885#	9289											

```
TST51   017132          5970#   9290
TST52   017336          6057#   9291
TST53   017462          6116#   9292
TST54   017606          6173#   9293
TST55   017732          6230#   9294
TST56   020056          6286#   9295
TST57   020202          6339#   9296
TST6    004746          3308#   9255
TST60   020246          6366#   9297
TST61   020320          6401#   9298
TST62   020576          6480#   9299
TST63   020760          6554#   9300
TST64   021372          6656#   9301
TST65   021560          6735#   9302
TST66   022264          6889#   9303
TST67   022474          6964#   9304
TST7    005074          3351#   9256
TST70   022600          7020#   9305
TST71   023222          7156#   9306
TST72   023644          7286#   9307
TST73   023754          7348#   9308
TST74   024606          7550#   9309
TST75   025264          7717#   9310
TST76   025500          7808#   9311
TST77   025756          7914#   9312
THAT1M= 104440         11980#
THAT1S= 104435         11977#
THAT11= 104431          3983    4128    4257    4337    4406    4459    4520    4579    4632    4662    6756    6909    7368
                       11973#
THAT12= 104432         11974#
THAT14= 104433         11975#
THAT15= 104434          6353    6380    6420    7399    7566    7638    7695    8457   11267   11976#
THAT16= 104423          3033    3074    3145    3178    3227    3236    3267    3282    3317    3361    3405    3470    3512
                        3539    3582    3599    3630    3647    3680    3716    3746    3788    3812    3852    3870    3913
                        3931    4153    7442    7597    7619    7657    7678    7731    7752    7767    7823    7842    7857
                        7881    7931    8047    8062    8106    8138    8194    8210    8226    8249    8272    8286    8323
                        8345    8375    8420    8444    8479    8592    8604    8660    8672    8745    8757    8899    8908
                        8920    8927    8941   11232   11967#
THAT2S= 104436          7779    8770   11978#
THAT32= 104424          4274    4355    4421    4593    4677    4751    4833    4902    4981    5050    5129    5201    5270
                        5342    5667    5682    5755    5770    5842    5856    6518    6779    6820    6855    6939    6997
                        7054    7074    7190    7212    7321    8080    8164    8234    8360   11968#
THAT48= 104425          4474    4535    5923    5938    6008    6023    7413    7493    8006   11969#
THAT64= 104426          6695    8270    8403    8497    8541   11970#
THAT8S= 104437          3691    3857    3918    7467    7791    7827    7896    8094    8198    8330    8465    8903    8923
                       11239   11979#
THAT80= 104427          7447    7604    7664    7738    8953   11971#
THAT96= 104430          4728    4814    4888    4964    5036    5114    5187    5250    5324    5390    5443    5563    5648
                        5732    5818    5903    5988    6082    6139    6197    6253    6309    6503    6617    6680    6982
                        7039    7175    7304    7970    8050   11972#
TYPOS = 104405          9071    9078   11951#
TYPE  = 104401          2790    2810    2829    2834    2837    2844    2848    2856    2865    3022    3026    3045    3106
                        3111    3124    3126    4791    4792    4794    4939    4940    4941    5090    5091    5092    5499
                        8724    9065    9072    9079    9404    9416    9419    9424    9435    9482    9483    9487    9488
                        9495    9506    9508    9518    9519    9521    9529    9562   10664   10665   10697   10698   11186
                       11250   11278   11294   11339   11433   11502   11558   11570   11624   11625   11628   11639   11649
```

```
                    11660   11679   11727   11733   11738   11742   11747   11748   11750   11753   11757   11823   11825
                    11907   11947#
TYPERR  034050       9434    9470#
TYPOC = 104402       2836    2847    2864    9418    9423    9503    9563   11627   11948#
TYPON = 104404      11950#
TYPOS = 104403      11949#
T.ASOF  001556       2637#   3422    3455    3478    3518    3550   8598*   8599    8626   8666*   8667    8692    8960
                    13223   13233
T.BA    001544       2632#   5406    5410    5460    5464    5582    5591    5595    5598    7095    7100    7114    7118
                     7233    7238    7252    7256   10188   10191   10276   13236
T.CS1   001540       2630#   3037    3078    3427    3461    3484    3524    3556    3752    3793    3820    4168    5399
                     5452    8913    8932    9788    9863   10101*  10102   10106   10119   10192   10316   10324   10341
                    10374   10642   13223   13233
T.CS2   001550       2634#   3039    3095    4161   10318   10375   13223   13233
T.DA    001546       2633#  10268   10271   10278   10280   13235
T.DB    001564       2640#
T.DCYL  001560       2638#  10265   10277   13235
T.DS    001552       2635#   3051    3056    3080    3082    3156    3170    3184    3190    3241    3272    3288    3323
                     3334    3370    3433    3439    3490    3496    3684    3699    3706    3723    8763    8779   10320
                    10376   13223   13233
T.ECPS  001570       2642#
T.ECPT  001572       2643#
T.ER    001554       2636#  10322   10377   13223   13233
T.MR1   001566       2641#  13223
T.MR2   001574       2644#   7887    8450
T.MR3   001576       2645#
T.SPAR  001562       2639#
T.WC    001542       2631#   5413    5417    5467    5471    5602    5605    5608    5611    7104    7110    7122    7127
                     7242    7248    7260    7265   10138   10141   10177   10220   10275   13236
UBAERR= 000002      10196   12290#
UFE   = 000400       2017#  10442
UFERR = 000004       3605    3653   10444   12281#
UNLOAD= 000107       1981#
UNS   = 040000       2042#
UNSERR= 040000       8424   12266#
UPE   = 020000       2022#  10458
UPERR = 000100       5582   10460   12285#
VV    = 000100       2053#   3184
WATSRT  035372       9735    9738    9741    9744    9747    9750    9753    9756    9759    9762    9765    9768    9771
                     9774#
WCE   = 040000       2023#  10413
WCERR = 000001      10140   12289#
WCKERR= 000004       7079    7217   10415   12272#
WLE   = 004000       2039#
WLERR = 004000      12263#
WRDATA= 000123       1987#   4266    4347    4413    4466    4527    4586    4650    4715    4802    4877    4956    5025
                     5102    5176    5238    5313    5382    5505    5636    5720    5806    5891    5976    6495    6582
                     6672    6771    6970    7031    7167    7296    7391    7411    7457    7558    7626    7849    7962
                     7981    8949
WRDCNT  001706       2712#  10969   10997*  11045*  11169
WRDNUM  001704       2711#  10968   10998*  11044*  11168
WRHEAD= 000127       1989#   4113    4238    4314    4393    4445    4499    4560    4621    6409    6742    6899    7357
                     8379    8489    8533   11253
WRL   = 004000       2056#   3082
WRTCHK= 000131       1990#   6989    7046    7066    7182    7204    7313    7685
WRTGAT= 040000       2075#
```

```
$APTHD  000224          2183    2189#
$ASTAT= ****** U        9360    9375
$ATYC   033304          9331    9333#
$ATY1   033260          9329#
$ATY3   033266          9330#  11324
$ATY4   033276          9332#   9440
$AUTOB  001134          2220#   2800*  11617   11774
$BASE   001352          2318#   2835    2843*   2889    2913
$BDADR  001122          2215#
$BDDAT  001126          2217#
$BELL   001266          2269#   9404    9460   11570   11767
$CDW1   001356          2320#
$CHARC  044140         11341*  11351*  11358   11367*  11372#
$CKSWR  045114         11603#  11955
$CMTAG  001100          2203#   2742    2743    2751    2757    2758    2759
$CM1  = 000020          2235#   2236#   2237#   2238#   2239#   2240#   2241#   2242#   2243#   2244#   2245#   2246#   2247#
                        2248#   2249#   2250#   2251#
$CM2  = 000040          2235#   2236#   2237#   2238#   2239#   2240#   2241#   2242#   2243#   2244#   2245#   2246#   2247#
                        2248#   2249#   2250#   2251#
$CM3  = 000020          2233#   2235
$CM4  = 000020          2251#   2252#   2253#   2254#   2255#   2256#   2257#   2258#   2259#   2260#   2261#   2262#   2263#
                        2264#   2265#   2266#   2267#
$CNTLC  046012         11558   11639   11767#
$CNTLG  046024         11624   11769#
$CNTLU  046017         11649   11742   11768#
$CORE   032406          9122    9149#
$CPUOP  001324          2292#
$CRLF   001273          2271#   9079    9424    9435    9460    9482    9487    9495    9508    9521   10664   10697   11340
                       11375   11660   11747   11767   11828
$CROUT  032436          9149    9156#
$DBLK   044360         11399   11433   11441#
$DEVCT  001306          2283#
$DEVM   001354          2319#   3009    3998*   4048*   8555    8577    8646    8717    8728    8883
$DOAGN  032170          9061    9082    9088#
$DTBL   044350         11402   11437#
$ENDAD  032160          2788    9084#   9455
$ENDCT  032024          2757    9063#
$ENULL  032174          9091#
$ENV    001316          2288#   2794    9338    9362    9437   11319
$ENVM   001317          2289#   2780    2805    3010    9340   11321   11326
$EOP    031770          2812    8725    9053#   9323    9535    9566   11285
$EOPCT  032016          2757*   2811*   9060#   9064    9534*   9564*  11284*
$ERFLG  001103          2206#   2946    3989    4028*   9175    9219    9221    9227*   9249    9399*   9460    9674   11280*
$ERMAX  001115          2212#   2760*   9221    9244*   9249
$ERROR  033526          2751    9190    9397#
$ERRPC  001116          2213#   9406*   9407*   9408    9414*   9415*   9417    9421*   9422    9460   13223   13226   13232
                       13240
$ERRTB  001360          2337#   9479
$ERTTL  001112          2210#   9076    9080*   9405*   9460
$ESCAP  001264          2268#   2759*   2890*   3030*   3060*   3133*   5540*   5585*   6594*   9243*   9451    9453    9460
                        9794*   9795*  10014*  10015*  10114*  10115*  10737   10741*  10742*  11281*
$ETABL  001316          2287#
$ETEND  001360          2195    2321#
$FATAL  001300          2280#   9366*
$FFLG   033524          9329*   9332*   9360    9369*   9377#
$FILLC  001156          2231#  11344   11375
```

```
$FILLS  001155          2230#  11375
$GDADR  001120          2214#
$GDDAT  001124          2216#
$GET42  032150          9081#
$GTSWR  045204         11625#  11953
$HD   = 000000          1796
$HIBTS  000224          2190#
$HIOCT  046212         11816*  11827#
$ICNT   001104          2207#   9234*   9235    9237*   9248
$ILLUP  046450         11877   11893   11910#
$INTAG  001135          2221#  11622*  11641   11661   11774
$ITEMB  001114          2211#   9408*   9439    9460    9473
$KTNEX  032400          9123    9148#
$KTOUT  032370          9140    9145#
$KT11   032236          2817    9121#   9125*   9148*
$LF     001274          2272#   9460   11375   11757   11767   11828
$LFLG   033523          9370*   9376#
$LPADR  001106          2208#   2761*   9216*   9225*   9241*   9246    9248
$LPERR  001110          2209#   2762*   3860*   3921*   3970*   4122*   4136*   4176*   4262*   4342*   4660*   4796*   4871*
                        4945*   5019*   5096*   5169*   6404*   6432*   6807*   6892*   6922*   7061*   7197*   7385*   7478*
                        8733*   8888*   9225    9242*   9248    9450    9676
$LSTAD  032502          2815    9161*   9168#
$LSTBK  032504          2808    2813    9162*   9169#
$MADR1  001330          2305#   2826*
$MADR2  001334          2309#
$MADR3  001340          2312#
$MADR4  001344          2315#
$MAIL   001276          2191    2195    2278#   2779    2794    9240    9437   11319
$MAMS1  001326          2299#   2825*   4785    4866    4934    5015    5085    5165
$MAMS2  001332          2307#
$MAMS3  001336          2310#
$MAMS4  001342          2313#
$MBADR  000226          2191#
$MFLG   033522          9330*   9336    9371*   9375#
$MNEM   046042         11628   11772#
$MSGAD  001312          2285#   9346*   9349
$MSGLG  001314          2286#   9351*
$MSGTY  001276          2279#   9344    9352*   9364    9368*
$MSWR   046031         11625   11770#
$MTYP1  001327          2300#   2824*
$MTYP2  001333          2308#
$MTYP3  001337          2311#
$MTYP4  001343          2314#
$MXCNT  033030          9238    9248#
$NULL   001154          2229#  11346   11375
$NWTST= 000001          2897#   2899    2928#   2930    2958#   2960    3211#   3213    3248#   3250    3300#   3302    3344#
                        3346    3377#   3379    3563#   3565    3610#   3612    3658#   3660    3729#   3731    3772#   3774
                        3830#   3832    3896#   3898    3942#   3944    4095#   4097    4215#   4217    4297#   4299    4377#
                        4379    4430#   4432    4482#   4484    4543#   4545    4602#   4604    4699#   4701    4773#   4775
                        4854#   4856    4922#   4924    5003#   5005    5073#   5075    5153#   5155    5224#   5226    5297#
                        5299    5369#   5371    5422#   5424    5476#   5478    5622#   5624    5705#   5707    5792#   5794
                        5876#   5878    5961#   5963    6046#   6048    6105#   6107    6162#   6164    6219#   6221    6275#
                        6277    6331#   6333    6358#   6360    6392#   6394    6472#   6474    6541#   6543    6647#   6649
                        6721#   6723    6881#   6883    6956#   6958    7002#   7004    7138#   7140    7276#   7278    7329#
                        7331    7534#   7536    7708#   7710    7799#   7801    7906#   7908    7941#   7943    8022#   8024
                        8113#   8115    8169#   8171    8294#   8296    8503#   8505    8566#   8568    8634#   8636    8700#
```

```
                    8702    8864#   8866
$OCNT   044612     11474*  11503*  11516#
$OMODE  044614     11469*  11473*  11478   11481*  11492*  11518#
$OVER   033014      9192    9217    9226    9236    9245#
$PASS   001304      2282#   2779*   2880*   3012    8721    9057*   9058*   9069    9091    9232    9249
$PASTM  000232      2193#
$POWER  046456     11908   11913#
$PWRON  046310      2755   11877#  11905
$PWRMG  046444     11908#
$PWRUP  046362     11887   11893#
$QUES   001272      2270#   9460   11375   11679   11750   11767   11825   11828
$ROCHR  045456     11692#  11956
$RODEC= ****** U   11959
$ROLIN  045546     11715#  11957
$ROOCT  046054     11789#  11958
$ROSZ = 000010     11708#
$REGAD  001160      2233#
$REGO   001162      2235#
$REG1   001164      2236#
$REG10  001202      2243#   5404*   5409*   5416*   5456*   5463*   5470*   5590*   5597*   5604*   5610*   6792*   6802*
                    7099*   7109*   7117*   7126*   7237*   7247*   7255*   7264*  10032   10065   10361*  10992*  13237
                   13241
$REG11  001204      2244#   5403*   5410*   5417*   5457*   5464*   5471*   5591*   5598*   5605*   5611*   6793*   6803*
                    7100*   7110*   7118*   7127*   7238*   7248*   7256*   7265*  10362*  10993*  13237   13241
$REG12  001206      2245#  10037   10994*  13237   13241
$REG13  001210      2246#  13237
$REG14  001212      2247#  13237
$REG15  001214      2248#  13237
$REG16  001216      2249#  13237
$REG17  001220      2250#  13237
$REG2   001166      2237#
$REG3   001170      2238#
$REG4   001172      2239#
$REG5   001174      2240#
$REG6   001176      2241#
$REG7   001200      2242#
$RESRE  046252     11861#  11960
$RTNAD  032172      9090#
$R2A  = ****** U   11961
$SAVRE  046214     11845#  11959
$SAVR6  046454     11886*  11894   11895*  11896*  11912#
$SCOPE  032506      2749    9184#
$SETUP= 000137      2721#   2748    2749    2751    2753    2755    2757    2758    2759    2761    2788    2791    9055
                    9185    9398    9447    9455   11562   11567   11568   11598   11774
$SIZE   032200      2807    9109#
$SIZEX  032442      9147    9157#
$STUP = 177777      2721#
$SVLAD  032760      9200    9239#
$SWR  = 167400      1784#   1796    1800    1801    1802    1803    1804    1805    1806    1807    2267    2268    2269
                    2758    2759    2761    2762    2905    2936    2998    3219    3259    3309    3352    3396    3574
                    3622    3672    3738    3780    3841    3904    3965    4105    4232    4308    4387    4439    4493
                    4553    4612    4709    4784    4865    4933    5014    5084    5164    5233    5308    5377    5430
                    5491    5631    5715    5801    5886    5971    6058    6117    6174    6231    6287    6340    6367
                    6402    6481    6555    6657    6736    6890    6965    7021    7157    7287    7349    7551    7718
                    7809    7915    7953    8033    8125    8181    8309    8516    8574    8643    8716    8880    9050
                    9056    9083    9089    9091    9176    9177    9178    9179    9180    9186    9203    9205    9206
```

```
                    9219    9220    9221    9228    9229    9230    9242    9245    9248    9389    9390    9391    9392
                    9393    9402    9409    9444    9448    9460   11909
$SWREG  001320      2290#   2782
$SWRMK= 000000      1807    1808    9180    9181    9209
$SWOBT  033032      9215    9249#   9794   10014   10114   10741
$TERM = 000102     11981#
$TESTN  001302      2281#   9240*   9471*   9472*   9792   10012   10112   10739   11214*  13223   13226   13232   13240
$TIMES  001262      2267#   2758*   2905*   2936*   2998*   3219*   3259*   3309*   3352*   3396*   3574*   3622*   3672*
                    3738*   3780*   3841*   3904*   3965*   4105*   4232*   4308*   4387*   4439*   4493*   4553*   4612*
                    4709*   4784*   4865*   4933*   5014*   5084*   5164*   5233*   5308*   5377*   5430*   5491*   5631*
                    5715*   5801*   5886*   5971*   6058*   6117*   6174*   6231*   6287*   6340*   6367*   6402*   6481*
                    6555*   6657*   6736*   6890*   6965*   7021*   7157*   7287*   7349*   7551*   7718*   7809*   7915*
                    7953*   8033*   8125*   8181*   8309*   8516*   8549*   8574*   8643*   8716*   8880*   9056*   9228*
                    9235    9238*   9248
$TKB    001146      2226#  11522   11543   11554   11579   11607   11634
$TKCNT  044616     11523#  11538*  11568   11585*  11699   11701*
$TKINT  044626      2739    3802    8617   11538#  11559   11620
$TKQEN= 044625     11527#  11593   11704
$TKQIN  044620     11524#  11539*  11540   11591*  11592*  11593   11595*
$TKQOU  044622     11525#  11540*  11702   11703*  11704   11706*
$TKQSR  044624     11526#  11539   11595   11706
$TKS    001144      2225#   5546*   5562*   5617*   6600*   6616*   6644*  11522   11544*  11575*  11577   11583*  11605
                   11621*  11631   11643*  11663*
$TKSRV  044676     11541   11554#
$TMPO   001222      2251#   2839*   2841    2843    2850*   2852    2855    2867*   2869    2871    2873*   2874*   2875*
                    2876*   2877*   2879   10100*  10104*
$TMP1   001224      2252#   8727*   8803*   8828    8839    8861*   8882*   8983*   9008    9019    9041*  10526   10907*
                   10909*  10911   10925*  10941   10942   10946   10948*
$TMP10  001242      2259#  10300*  10480   10582   10712
$TMP11  001244      2260#  10301*  10561   10706
$TMP12  001246      2261#  10302*  10541   10701
$TMP13  001250      2262#  10304*  10308*  10338   10477   10495   10539   10559   10580   10631   10686
$TMP14  001252      2263#  10654*  10673   10677*  10679   10683*  10704   10708*  10710   10714*
$TMP15  001254      2264#  10655*  10660*
$TMP16  001256      2265#   9884*   9889*
$TMP17  001260      2266#   9886*   9894    9896*
$TMP2   001226      2253#  10527
$TMP3   001230      2254#  10039*  10053*  10068
$TMP4   001232      2255#   3028*   3049    3053*
$TMP5   001234      2256#  10764*  10774*  10781
$TMP6   001236      2257#  10761*  10782
$TMP7   001240      2258#   3131*   3200*   3203   10762*  10783
$TN   = 000112      1785#   1796    2897    2905#   2928    2936#   2958    2998#   3211    3219#   3248    3259#   3300
                    3309#   3344    3352#   3377    3396#   3563    3574#   3610    3622#   3658    3672#   3729    3738#
                    3772    3780#   3[illegible]    3841#   3896    3904#   3942    3965#   4095    4105#   4215    4232#   4297
                    4308#   4377    4387#   4430    4439#   4482    4493#   4543    4553#   4602    4612#   4699    4709#
                    4773    4784#   4854    4865#   4922    4933#   5003    5014#   5073    5084#   5153    5164#   5224
                    5233#   5297    5308#   5369    5377#   5422    5430#   5476    5491#   5622    5631#   5705    5715#
                    5792    5801#   5[illegible]    5886#   5961    5971#   6046    6058#   6105    6117#   6162    6174#   6219
                    6231#   6275    6287#   6331    6340#   6358    6367#   6392    6402#   6472    6481#   6541    6555#
                    6647    6657#   6721    6736#   6881    6890#   6956    6965#   7002    7021#   7138    7157#   7276
                    7287#   7329    7349#   7534    7551#   7708    7718#   7799    7809#   7906    7915#   7941    7953#
                    8022    8033#   8113    8125#   8169    8181#   8294    8309#   8503    8516#   8566    8574#   8634
                    8643#   8700    8716#   8864    8880#   9210    9250   11214
$TPB    001152      2228#  11364*  11375
$TPFLG  001157      2232#  11313   11375
```

```
$TPS     001150          2227#  11362   11375
$TRAP    046466          2753   11924#
$TRAP2   046510         11935#  11946
$TRP   = 000041         11939#  11948#  11949#  11950#  11951#  11952#  11953   11954#  11955   11956#  11957#  11958#  11959#
                        11960#  11961#  11962#  11963#  11964#  11965#  11966#  11967#  11968#  11969#  11970#  11971#  11972#
                        11973#  11974#  11975#  11976#  11977#  11978#  11979#  11980#  11981#
$TRPAD   046522         11929   11946#  11981
$TSTM    000230          2192#
$TSTNM   001102          2205#   8561*   9055*   9175    9212*   9239*   9240    9245    9249    9401    9460    9471    11219
$TTYIN   046002         11717   11718   11730   11748   11762   11766#
$TYPBN=  ****** U       11952
$TYPDS   044144         11387#  11951
$TYPE    043662          9357   11313#  11939   11947
$TYPEC   044074         11343   11350   11357   11362#  11363   11665
$TYPEX   044142         11368   11370   11373#
$TYPOC   044414         11472#  11948
$TYPON   044430         11471   11474#  11950
$TYPOS   044370         11467#  '1949
$UNIT    001310          2284#
$UNITM   000234          2194#
$USWR    001322          2291#
$VECT1   001346          2316#   2845    2854*   2855*   2858    2878*   2879*   2884    2888
$VECT2   001350          2317#
$XTSTR   032532          9194#
$$GET4=  000000          9083#
$$SW08=  000112          9249#   9250    9251#   9252#   9253#   9254#   9255#   9256#   9257#   9258#   9259#   9260#   9261#
                         9262#   9263#   9264#   9265#   9266#   9267#   9268#   9269#   9270#   9271#   9272#   9273#   9274#
                         9275#   9276#   9277#   9278#   9279#   9280#   9281#   9282#   9283#   9284#   9285#   9286#   9287#
                         9288#   9289#   9290#   9291#   9292#   9293#   9294#   9295#   9296#   9297#   9298#   9299#   9300#
                         9301#   9302#   9303#   9304#   9305#   9306#   9307#   9308#   9309#   9310#   9311#   9312#   9313#
                         9314#   9315#   9316#   9317#   9318#   9319#   9320#   9321#   9322#   9323#
$OFILL   044613         11468*  11472*  11482   11517#
$40CAT   000000          2159#   9186    9411
.        = 065414        2144#   2160    2164#   2170#   2172#   2179    2180#   2182#   2184#   2202#   2273    2683#   2746
                         2761    2762    2804#   9068#   9091    9092#   9248    9249    9378#   9460   11375   11441#  11522
                        11526#  11527   11528#  11766#  11767   11774#  11828   11889   11911   11981   13222#  13335#  13336#
                        13337#  13344#  13345#
.$ASTA=  ****** U        9330    9333
.$X    = 000224          2179#   2184
```

```
COMMEN  1918#
ENDCOM  1918#
ERROR   1812#  2922  2951  3000  3034  3043  3068  3075  3099  3138  3146  3154  3159  3168  3173
               3179  3182  3187  3194  3221  3228  3231  3237  3240  3244  3261  3268  3270  3277  3283
               3286  3293  3311  3318  3321  3328  3339  3354  3362  3368  3375  3398  3406  3412  3426
               3431  3437  3444  3453  3459  3465  3471  3477  3482  3488  3494  3500  3513  3516  3522
               3528  3540  3543  3554  3560  3576  3583  3586  3600  3606  3609  3624  3631  3634  3648
               3654  3674  3681  3688  3696  3703  3710  3717  3721  3727  3740  3747  3756  3768  3771
               3782  3789  3797  3817  3824  3845  3853  3858  3863  3871  3882  3890  3906  3914  3919
               3924  3932  3938  3972  3984  3986  3997  4008  4018  4025  4034  4047  4090  4110  4126
               4129  4132  4151  4154  4157  4165  4172  4180  4183  4184  4185  4191  4200  4235  4258
               4261  4264  4275  4284  4294  4311  4338  4340  4344  4356  4365  4374  4390  4407  4410
               4422  4428  4442  4460  4463  4475  4481  4496  4521  4524  4536  4542  4557  4580  4583
               4594  4597  4615  4633  4636  4663  4666  4678  4681  4687  4695  4712  4729  4732  4752
               4755  4761  4768  4799  4815  4818  4834  4836  4842  4849  4875  4889  4892  4903  4905
               4911  4918  4949  4965  4968  4982  4985  4991  4999  5023  5037  5040  5051  5054  5060
               5068  5100  5115  5118  5130  5133  5140  5148  5173  5188  5191  5202  5205  5211  5219
               5235  5251  5254  5271  5274  5285  5293  5310  5325  5328  5343  5345  5356  5364  5379
               5391  5396  5405  5411  5418  5432  5444  5449  5458  5465  5472  5493  5577  5583  5592
               5599  5606  5612  5633  5649  5652  5668  5671  5683  5686  5692  5700  5717  5733  5736
               5756  5759  5771  5774  5780  5788  5803  5819  5822  5843  5846  5857  5863  5871  5888
               5904  5907  5924  5927  5939  5942  5948  5956  5973  5989  5993  6009  6012  6024  6027
               6033  6041  6060  6083  6086  6092  6100  6119  6140  6143  6149  6157  6176  6198  6201
               6207  6215  6233  6254  6257  6263  6271  6289  6310  6313  6319  6327  6342  6354  6357
               6369  6381  6387  6408  6421  6424  6436  6444  6445  6446  6453  6461  6484  6504  6507
               6519  6522  6528  6536  6580  6631  6635  6661  6681  6684  6696  6699  6705  6713  6739
               6757  6760  6780  6786  6795  6804  681C  6821  6824  6830  6838  6856  6862  6868  6876
               6896  6910  6913  6926  6940  6943  6967  6983  6986  6998  7001  7025  7040  7043  7055
               7058  7064  7075  7081  7102  7111  7120  7128  7161  7176  7179  7191  7194  7201  7213
               7219  7240  7249  7258  7266  7289  7305  7308  7322  7325  7369  7372  7389  7400  7403
               7414  7422  7427  7443  7446  7448  7455  7468  7472  7483  7494  7497  7505  7513  7553
               7572  7581  7586  7598  7602  7605  7608  7620  7623  7644  7648  7658  7662  7665  7667
               7679  7682  7701  7705  7720  7732  7735  7739  7741  7753  7756  7768  7780  7786  7788
               7792  7812  7824  7828  7831  7843  7846  7858  7864  7867  7882  7885  7891  7902  7917
               7932  7938  7955  7971  7974  8007  8013  8018  8036  8048  8051  8063  8081  8086  8090
               8095  8107  8109  8127  8139  8142  8145  8165  8168  8183  8195  8199  8211  8214  8227
               8230  8235  8238  8250  8253  8287  8293  8312  8324  8327  8331  8334  8346  8349  8361
               8364  8376  8404  8407  8421  8427  8430  8445  8448  8454  8461  8466  8480  8483  8498
               8501  8519  8527  8542  8545  8579  8593  8609  8613  8631  8648  8661  8677  8681  8697
               8738  8746  8748  8758  8761  8767  8776  8784  8892  8900  8904  8909  8912  8921  8924
               8928  8931  8942  8954  8957  8964  11217 11233 11236 11240 11243 11246 11268 11271
ESCAPE  1918#
GETPRI  1918#  9117
GETSWR  1918#  2791#
INITSS  2717#  4124  4149  4178  4711  4797  4873  4947  5021  5098  5171  5309  5378  5431  5492
               5632  5716  5802  5887  5972  6059  6118  6175  6232  6288  6341  6368  6406  6434  6483
               6579  6660  6738  6808  6894  6924  6966  7024  7063  7160  7200  7288  7387  7453  7481
               7552  7719  7787  7811  7845  7866  7890  7916  7954  8035  8088  8126  8144  8182  8311
               8429  8453  8460  8518  8526  8578  8647  8736  8890  11216 11245
LOADLS  2719#  4714  4734  4825  4876  4973  5024  5120  5175  5312  5330  5503  5635  5658  5673
               5719  5742  5761  5805  5833  5847  5890  5914  5929  5975  5999  6014  6073  6130  6188
               6244  6300  6344  6371  6494  6509  6581  6671  6686  6741  6770  6811  6846  6927  6969
               6988  7030  7045  7065  7166  7181  7203  7295  7312  7390  7433  7456  7557  7588  7610
               7625  7649  7669  7684  7722  7743  7758  7770  7814  7833  7848  7870  7919  7961  7980
               7997  8038  8053  8065  8097  8129  8147  8185  8201  8216  8240  8276  8314  8336  8351
               8366  8378  8409  8432  8470  8488  8532  11223 11252
```

```
MSG     2897#  2899   2928#  2930   2957#  2960   3211#  3213   3248#  3250   3300#  3302   3344#  3346   3377#
        3379   3563#  3565   3610#  3612   3658#  3660   3729#  3731   3772#  3774   3830#  3832   3896#  3898
        3942#  3944   4095#  4097   4215#  4217   4297#  4299   4377#  4379   4430#  4432   4482#  4484   4543#
        4545   4602#  4604   4699#  4701   4773#  4775   4854#  4856   4922#  4924   5003#  5005   5073#  5075
        5153#  5155   5224#  5226   5297#  5299   5369#  5371   5422#  5424   5476#  5478   5622#  5624   5705#
        5707   5792#  5794   5876#  5878   5961#  5963   6046#  6048   6105#  6107   6162#  6164   6219#  6221
        6275#  6277   6331#  6333   6358#  6360   6392#  6394   6472#  6474   6541#  6543   6647#  6649   6721#
        6723   6881#  6883   6956#  6958   7002#  7004   7138#  7140   7276#  7278   7329#  7331   7534#  7536
        7708#  7710   7798#  7801   7906#  7908   7940#  7943   8021#  8024   8113#  8115   8169#  8171   8294#
        8296   8503#  8505   8566#  8568   8634#  8636   8700#  8702   8864#  8866
MULT    1918#
NEWTST  1918#  2897   2928   2958   3211   3248   3300   3344   3377   3563   3610   3658   3729   3772   3830
        3896   3942   4095   4215   4297   4377   4430   4482   4543   4602   4699   4773   4854   4922   5003
        5073   5153   5224   5297   5369   5422   5476   5622   5705   5792   5876   5961   6046   6105   6162
        6219   6275   6331   6358   6392   6472   6541   6647   6721   6881   6956   7002   7138   7276   7329
        7534   7708   7799   7906   7941   8022   8113   8169   8294   8503   8566   8634   8700   8864
OPCHK   2721#  4731   4754   4817   4835   4891   4904   4967   4984   5039   5053   5117   5132   5190   5204
        5357   5344   5651   5670   5665   5735   5758   5773   5821   5845   5906   5926   5941   5992   6011
        6066   6085   6142   6200   6256   6312   6356   6423   6506   6521   6634   6683   6698   6759   6823
        6912   6942   6985   7000   7042   7057   7178   7193   7307   7324   7371   7402   7425   7445   7470
        7496   7584   7601   7607   7622   7646   7661   7666   7681   7703   7734   7740   7755   7830   7884
        7973   8016   8108   8141   8167   8213   8229   8237   8252   8326   8333   8348   8363   8406   8447
        8482   8500   8544   8611   8679   8747   8760   8910   8929   8956   11235  11242  11270
POP     1918#  9372   9373   9826   9919   10116  10984  11003  11007  11171  11428  11817  11866  11898  11899
PUSH    1918#  9333   9335   9356   9815   9906   10090  10873  10884  11387  11791  11846  11879  11885
REPORT  1918#
RESDC   2718#  4762   4843   4912   4993   5062   5142   5213   5358   5694   5782   5865   5950   6035   6094
        6151   6209   6265   6321   6455   6530   6707   6832   6870   7507
RKLOAD  2720#  4727   4750   4813   4832   4887   4901   4963   4980   5035   5049   5113   5128   5186   5200
        5323   5341   5389   5442   5647   5666   5681   5731   5754   5769   5817   5841   5855   5902   5922
        5937   5987   6007   6022   6081   6138   6196   6252   6308   6352   6379   6502   6517   6679   6694
        6755   6778   6819   6854   6908   6938   6981   6996   7038   7053   7073   7174   7189   7211   7303
        7320   7367   7398   7441   7492   7596   7618   7656   7677   7730   7751   7766   7778   7822   7841
        7856   7880   7969   8005   8046   8061   8105   8137   8193   8209   8225   8248   8285   8322   8344
        8359   8374   8419   8442   8478   8496   8540   8551   8659   8744   8756   8898   8907   8919   8926
        8940   11231  11266
SCOPE   1813#  2904   2935   2997   3218   3258   3308   3351   3395   3573   3621   3671   3737   3779   3840
        3903   3964   4104   4231   4307   4386   4438   4492   4552   4611   4708   4783   4864   4932   5013
        5083   5163   5232   5307   5376   5429   5490   5630   5714   5800   5885   5970   6057   6116   6173
        6230   6286   6339   6366   6401   6480   6554   6656   6735   6889   6964   7020   7156   7286   7348
        7550   7717   7808   7914   7952   8032   8124   8180   8308   8515   8548   8573   8642   8715   8879
        9054
SETPRI  1918#  11695
SETTRA  11939#  11948  11949  11950  11951  11953  11955  11956  11957  11958  11959  11960  11961  11962  11963
        11964  11965  11966  11967  11968  11969  11970  11971  11972  11973  11974  11975  11976  11977  11978
        11979  11980
SETUP   1918#  2741
SKIP    1918#
SLASH   1918#
SPACE   1918#
STARS   1918#  2176   2178   2185   2198   2273   2276   2897   2903   2928   2934   2958   2996   3211   3217
        3248   3257   3300   3307   3344   3350   3377   3394   3563   3572   3610   3620   3658   3670   3729
        3736   3772   3778   3830   3839   3896   3902   3942   3963   4095   4103   4215   4230   4297   4306
        4377   4385   4430   4437   4482   4491   4543   4551   4602   4610   4699   4707   4773   4782   4854
        4863   4922   4931   5003   5012   5073   5082   5153   5162   5224   5231   5297   5306   5369   5375
        5422   5428   5476   5489   5622   5629   5705   5713   5792   5799   5876   5884   5961   5969   6046
```

```
            6056    6105    6115    6162    6172    6219    6229    6275    6285    6331    6338    6358    6365    6392    6400
            6472    6479    6541    6553    6647    6655    6721    6734    6881    6888    6956    6963    7002    7019    7138
            7155    7276    7285    7329    7347    7534    7549    7708    7716    7799    7807    7906    7913    7941    7951
            8022    8031    8113    8123    8169    8179    8294    8307    8503    8514    8566    8572    8634    8641    8700
            8714    8864    8878    9046    9095    9172    9328    9385    9460    9469    9576    9666   10746   10759   10787
           11175   11298   11377   11444   11521   11598   11613   11684   11708   11777   11830   11875   11891   11918
SWRSU       1918#   2763#
TRMTRP     11939#  11981
TYPBIN      1918#
TYPDEC      1918#   9069    9076
TYPNAM      1784#   1918#   2784
TYPNUM      1918#
TYPOCS      1918#
TYPOCT      1918#   2835    9417    9422   11626
TYPTXT      1918#   9065    9072
$$CMRE      2196#   2235    2236    2237    2238    2239    2240    2241    2242    2243    2244    2245    2246    2247    2248
            2249    2250
$$CMTM      2196#   2251    2252    2253    2254    2255    2256    2257    2258    2259    2260    2261    2262    2263    2264
            2265    2266
$$ESCA      1918#
$$NEWT      1918#   2897    2928    2958    3211    3248    3300    3344    3377    3563    3610    3658    3729    3772    3830
            3896    3942    4095    4215    4297    4377    4430    4482    4543    4602    4699    4773    4854    4922    5003
            5073    5153    5224    5297    5369    5422    5476    5622    5705    5792    5876    5961    6046    6105    6162
            6219    6275    6331    6358    6392    6472    6541    6647    6721    6881    6956    7002    7138    7276    7329
            7534    7708    7799    7906    7941    8022    8113    8169    8294    8503    8566    8634    8700    8864
$$SET      11939#  11948   11949   11950   11951   11953   11955   11956   11957   11958   11959   11960   11961   11962   11963
           11964   11965   11966   11967   11968   11969   11970   11971   11972   11973   11974   11975   11976   11977   11978
           11979   11980
$$SETM      2779#
$$SKIP      1918#
.EQUAT      1784#    1808
.HEADE      1784#    1786
.KT11       1784#    1918
.SETUP      1784#    2721
.SWRHI      1784#    1796
.SWRLO      1784#    1808#
.$ACT1      1784#
.$APTB      2274#
.$APTH      1784#    2174
.$APTY      1784#    9326
.$CMTA      1784#    2196
.$EOP       1784#    9044
.$ERRO      1784#    9383
.$ERRT      1784#
.$POWE      1784#   11873
.$ROOC      1784#   11775
.$READ      1784#   11519
.$SAVE      1784#   11828
.$SCOP      1784#    9170
.$SIZE      1784#    9093
.$TRAP      1784#   11916
.$TYPD      1784#   11375
.$TYPE      1784#   11296
.$TYPO      1784#   11442
.$4OCA      1784#    2142
```

. ABS. 065414 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DZR6KD.BIN,DSKZ:DZR6KD.SEQ/CRF/SOL/DOC=DZR6KD
RUN-TIME: 32 34 3 SECONDS
RUN-TIME RATIO: 387/71=5.4
CORE USED: 30K (60 PAGES)

DOCUMENT PAGES: 272

EOF1DZR6KDSEQ 00010000 770323 PDP10 411