RK611

FUNCT. CONTROLLER DIAG.

MD-11-DZR6K-C

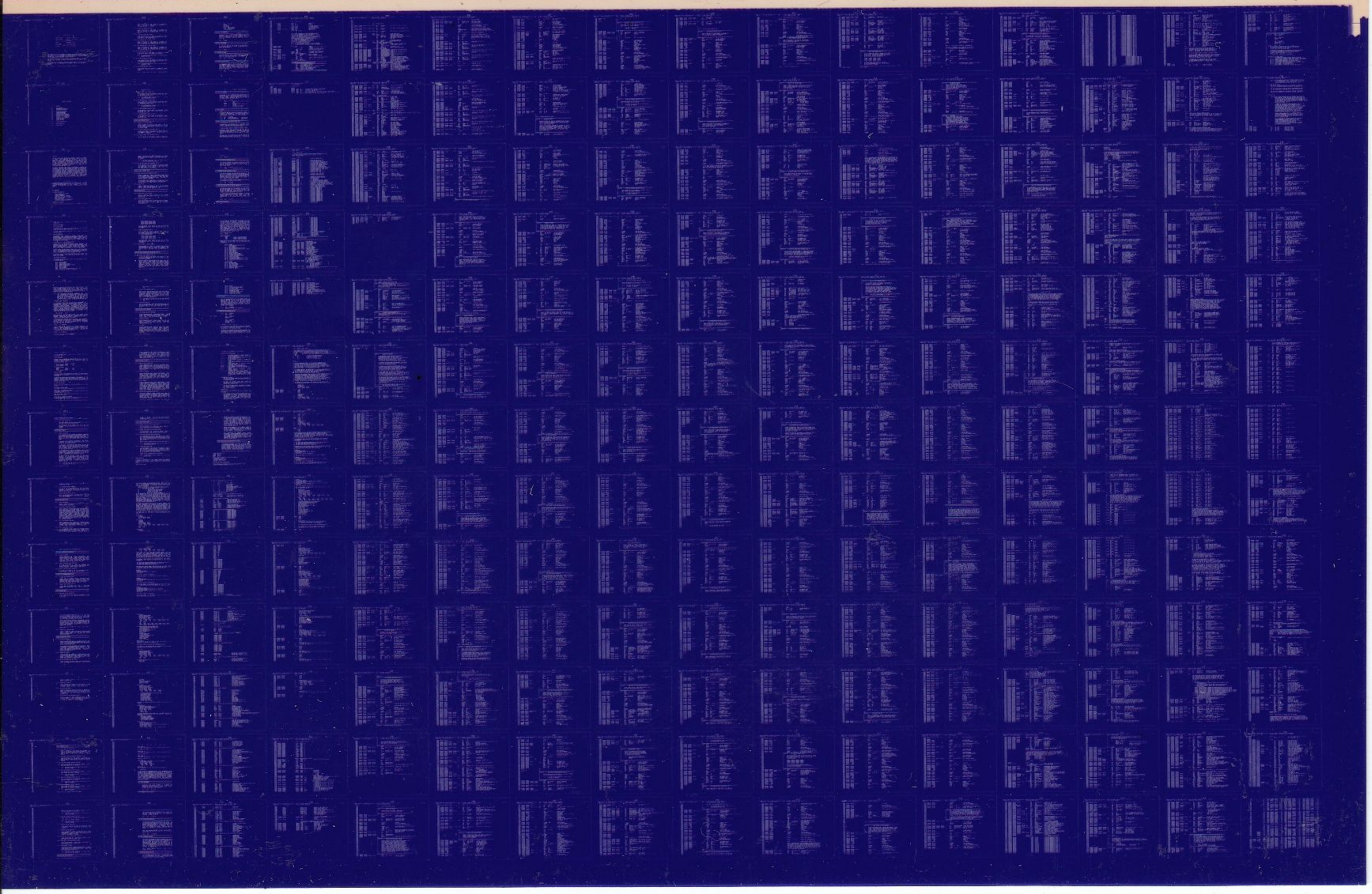
EP-DZR6K-C-DL-B

FICHE 1 OF 2

DEC 1976

digital

MADE IN USA



RK611

FUNCT. CONTROLLER DIAG.

MD-11-DZR6K-C

EP-DZR6K-C-DL-B

FICHE 2 OF 2

DEC 1976

digital

MADE IN USA

```
.REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6K-C-D
PRODUCT NAME:	RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DATE:	DECEMBER, 1976
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:

	TAG	LOCATION
KW11-L		
UNIBUS ADDRESS	KWLADD	1676
VECTOR ADDRESS	KWLVEC	1700
MM11		
UNIBUS ADDRESS		
BANK0	MMCSR1	1702
BANK 1	MMCSR2	1704
VECTOR ADDRESS	MMVEC	1706

.9 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM FOR ONE DRIVE IS APPROXIMATELY 50 SECONDS AND EACH SUBSEQUENT PASS IS APPROXIMATELY 6 MINUTES 50 SECONDS.

THE EXECUTION TIME FOR MULTIPLE DRIVES IS THE PRODUCT OF THE NUMBER OF DRIVES TIMES 52 SECONDS FOR THE FIRST PASS. FOR SUBSEQUENT PASSES THE RUN TIME IS THE PRODUCT OF 6 MINUTES 50 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.

                                THE DRIVE  MUST  BE  ON-LINE,  CYCLED  UP,  AND  WRITE
                                ENABLED.   IF  ANY ONE OF THESE CONDITIONS IS NOT TRUE
                                THAT DRIVE IS NOT TESTED AND  IT  IS  EXPECTED  TO  BE
                                OFF-LINE.    ADDRESSING   THAT   DRIVE   SHOULD  CAUSE
                                NON-EXISTANT DRIVE ERROR.  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.

                        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  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.

        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
```

```
540
541
542
543                     DATA TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY
544                     HEADER VRC ERROR SETS.
545
546             TEST 25 OPERATION INCOMPLETE
547
548                     FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR  21  HAS
549                     THE  WRONG  FORMAT.   ISSUE  A  WRITE  DATA  OF 400 TO
550                     CYLINER 0, TRACK 0, SECTOR 21.  MAKE SURE OPI SET.
551
552             TEST 26 OPI WITH HVRC ERROR
553
554                     FORMAT CYLINDER 312, TRACK 0 SUCH THAT  A  HEADER  VRC
555                     ERROR  IS  PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
556                     ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK
557                     0,  SECTOR  17.   THAT  BOTH  OPERATION INCOMPLETE AND
558                     HEADER VRC SET.
559
560             TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR
561
562                     FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR  20  HAS
563                     AN  HVRC  ERROR.   ISSUE  A WRITE DATA OF 400 WORDS TO
564                     CYLINDER 312, TRACK 0, AND SECTOR 21.  MAKE SURE  HVRC
565                     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 110
                        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  110  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:
```

000001 000020 000400 010000
000002 000040 001000 020000
000004 000100 002000 040000
000010 000200 004000 100000

MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUN IS CORRECT.

TEST 71 WRITE CHECK ERROR (PART 2)

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777 IN ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS ONE OF THE FOLLWOING CONFIGURATIONS:

177776 177757 177377 167777
177775 177737 176777 157777
177773 177677 175777 137777
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.

TEST 72 WRITE CHECK OF PARTIAL SECTOR

WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN CONFIGURATIONS. ISSUE A WRITE CHECK COMMAND OF 110 WORDS MAKING SURE THE 111TH WORD IS DIFFERENT THAN DATA ON DISK. MAKE SURE WRITE CHECK ERROR DOES NOT SET.

****
**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.

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  CONTPOLLER  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:

```
1068
1069
1070
1071                    THIS TABLE CONTAINS THE INFORMATION FOR EACH  ERROR  THAT  CAN
1072                    OCCUR.   THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER
1073                    FOUND IN LOCATION $ITEMB.  THIS NUMBER INDICATES WHICH ITEM IN
1074                    THE TABLE IS PERTINENT.
1075                        NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1076                        NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS
1077                               EXPLAINED AS FOLLOWS:
1078                        EM              ;;POINTS TO THE ERROR MESSAGE
1079                        DH              ;;POINTS TO THE DATA HEADER
1080                        DT              ;;POINTS TO THE DATA
1081                        DF              ;;POINTS TO THE DATA FORMAT
1082                    EM AND DH ARE ASCIZ DATA.  EM IS ALWAYS A MESSAGE BUT  DH  CAN
1083                    BE  A  MESSAGE  OR  A  SET OF COLUMN LABELS SPACED ACCROSS THE
1084                    PAGE, DT IS A STRING OF WORDS THAT POINT TO  THE  DATA  TO  BE
1085                    TYPED,  AND DF IS A STRING OF WORDS THAT TELL HOW THE DT WORDS
1086                    ARE TO BE TYPED.  IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A
1087                    PARTICULAR  FORMAT,  IT  IS  REPLACED WITH A ZERO.  THE NORMAL
1088                    USAGE OF THE ERROR TABLE IS TO HAVE A  TABLE  ENTRY  FOR  EACH
1089                    ERROR MESSAGE THAT CAN OCCUR.  IN THE INTEREST OF ECONOMICS OF
1090                    CORE MEMORY, THIS PROGRAM USES THE ERROR TABLE  IN  A  SIGHTLY
1091                    DIFFERENT MANNER AS DESCRIBED BELOW.
1092
1093                    THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT  FORMATS  THAT
1094                    ARE  USED  THROUGHOUT  THE  PROGRAM.   WHEN  AN ERROR IS TO BE
1095                    REPORTED, THE TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT  IS
1096                    CHOSEN  FROM  THE DEFINED SET.  THE TABLE ENTRY CHOSEN IS THEN
1097                    ALTERED BY CHANGING THE FIRST (AND POSSIBLY THE  SECOND)  WORD
1098                    TO  CONTAIN  THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE
1099                    MESSAGE PORTION OF THE REPORT.  THE DATA FIELDS FOR THAT ENTRY
1100                    ARE NEVER CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.
1101
1102                    THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW  WITH
1103                    THE  DEFINITION  OF  THE  ENTRY.  ALL DATA FIELDS ARE TYPED IN
1104                    OCTAL.
1105
1106                    ERROR ITEM 1
1107                        (MESSAGE)
1108                        TST NUM ERR PC  DRIVE
1109                        $TESTN  $ERRPC  DRVNUM
1110
1111                    ERROR ITEM 2
1112                        (MESSAGE)
1113                        (MESSAGE)
1114                        TST NUM ERR PC  DRIVE
1115                        $TESTN  $ERRPC  DRVNUM
1116                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
1117                          T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
1118                        RKBA    RKWC
1119                        T.BA    T.WC
1120
1121                    ERROR ITEM 3
1122                        (MESSAGE)
1123                        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.
```

```
1180
1181
1182
1183                                    ERROR ITEM 4
1184                                        (DESCRIPTION OF ERROR)
1185                                        ERROR IN OPERATION
1186                                        (DESCRIPTION OF OPERATION)
1187                                        TST NUM ERR PC  DRIVE
1188                                        $TESTN  $ERRPC  DRVNUM
1189                                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
1190                                          T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
1191                                        RKBA    RKWC
1192                                        T.BA    T.WA
1193                                         A00     B00     A01     B01     A02     B02     A03     B03
1194                                         $REG10  $REG11 $REG12 $REG13 $REG14 $REG15 $REG16 $REG17
1195
1196                                    THE ERRORS REPORTED BY THIS FORMAT ARE:
1197                                         CONTROLLER DETECTED DRIVE BUS ERROR
1198                                         DRIVE DETECTED DRIVE BUS ERROR
1199                                         SEEK INCOMPLETE
1200                                         NON-EXECUTABLE DRIVE FUNCTION
1201                                         DRIVE TIMING ERROR
1202                                         DRIVE UNSAFE
1203                                         AC LOW
1204                                         SPINDLE SPEED LOSS
1205                                         DRIVE OFF TRACK
1206                                         ILLEGAL DRIVE ADDRESS ERROR
1207                                         CYLINDER OVERFLOW
1208                                         DRIVE TYPE ERROR
1209                                         FORMAT ERROR
1210                                         WRITE LOCK ERROR
1211
1212                                    ERROR ITEM 5
1213                                    THIS ENTRY IS THE SAME AS  ITEM  4  WITH  THE  ADDITION  OF  A
1214                                    MESSAGE THAT FOLLOWS.  THIS MESSAGE IS:
1215
1216                                    "ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"
1217
1218                                    THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR A00
1219                                    THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.
1220
1221                                    ERROR ITEM 6
1222
1223                                        (DESCRIPTION OF ERROR)
1224                                        ERROR IN OPERATION
1225                                        (DESCRIPTION OF OPERATION)
1226                                        TST NUM ERR PC  DRIVE
1227                                        $TESTN  $ERRPC  DRVNUM
1228                                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA
1229                                          T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
1230                                        RKBA    RKWC
1231                                        T.BA    T.WC
1232
1233                                    THE ERRORS REPORTED BY THIS FORMAT ARE:
1234                                        DATA CHECK
1235                                        WRITE CHECK
```

```
1236
1237
1238
1239                                        ECC HARD
1240                                        DATA LATE
1241                                        OPERATION INCOMPLETE
1242                                        HEADER VRC ERROR
1243                                        BAD SECTOR ERROR
1244
1245                                    ERROR ITEM 7
1246                                        (DESCRIPTION OF ERROR)
1247                                        ERROR IN OPERATION
1248                                        (DESCRIPTION OF OPERATION)
1249                                        TST NUM ERR PC  DRIVE
1250                                        $TESTN  $ERRPC  DRVNUM
1251                                          RKCS1   RKCS2   RKDS    RKER    RKASOF
1252                                          T.CS1   T.CS2   T.DS    T.ER    T.ASOF
1253
1254                                          THE ERRORS REPORTED BY THIS FORMAT ARE:
1255                                          NON-EXISTANT DRIVE
1256                                          NON-EXISTANT MEMORY
1257                                          CONTROLLER TIME OUT
1258                                          UNIT FIELD ERROR
1259                                          MULTIPLE DRIVE SELECT
1260                                          PROGRAMMING ERROR
1261                                          UNIBUS PARITY ERROR
1262                                          ILLEGAL FUNCTION CODE
1263
1264                                    DESCRIPTON OF OPERATION CAN BE ANY COMMAND,  EITHER  LEGAL  OR
1265                                    ILLEGAL.
1266
1267                                    ERROR ITEM 10
1268                                        (DESCRIPTION OF ERROR)
1269                                        ERROR AT COMPLETION OF OPERATION
1270                                        (DESCRIPITON OF OPERATION)
1271                                        TST NUM ERR PC  DRIVE
1272                                        $TESTN  ERRPC   DRVNUM
1273                                          EXPT    IS
1274                                          $REG10  $REG11
1275
1276                                    THE ERRORS REPORTED BY THIS FORMAT ARE  SOFTWARE  DETECTED  BY
1277                                    COMPARING  EXPECTED RESULTS WITH ACTUAL RESULTS.  THE SPECIFIC
1278                                    ERRORS ARE:
1279                                          WORD COUNT INCORRECT
1280                                          BUS ADDRESS INCORRECT
1281                                          CYLINDER ADDRESS INCORRECT
1282                                          TRACK ADDRESS INCORRECT
1283                                          SECTOR ADDRESS INCORRECT
1284
1285                                    ERROR ITEM 11
1286                                        (ERROR INDICATOR OR STATUS BIT)
1287                                        NOT SET AS A RESULT OF
1288                                        (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION
1289                                        TST NUM ERR PC  DRIVE
1290                                        $TESTN  $ERRPC  DRVNUM
1291                                          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:

                                "RESET 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
```

```
;*****************************************************************

;       THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND
;       THE LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF
;       PRESENT; CLEARED OTHERWISE, AND THE APPROPRIATE
;       INTERRUPT VECTORS ARE SET UP.

;       CALL:   JSR     PC,OPTTST

;*****************************************************************
;LINE CLOCK SIMULATION ROUTINE
;*****************************************************************

;       THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK.  TO
;       DO THIS THE VALUE STORED IN MILCNT IS USED AS THE BASE
;       AND REPRESENTS THE NUMBER OF TIMES A DECREMENT AND
;       BRANCH LOOP CAN BE DONE IN 1 MILLISECOND.  THE TIMCNT
;       VALUE IS DECREMENTED AND IF IT REACHED 0 THE LINE
;       CLOCK TICK COUNTER IS BUMPED.  THEN THE TIMCNT IS
;       RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
;       TICK).

;       THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1
;       MILLISECOND AND BUMPS THE LINE CLOCK TICK COUNTER FOR
;       EACH 16 CALLS.

;       CALL:   JSR     PC,MYTIME

;*****************************************************************
;WAIT FOR INTERRUPT ROUTINE
;*****************************************************************

;       THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS.
;       THE CALL SPECIFIES HOW MANY TICKS OF THE LINE CLOCK
;       ARE TO ELAPSE WHILE WAITING FOR INTERRUPT.  IF
;       INTERRUPT DOES NOT OCCUR IN THAT PERIOD OF TIME, AN
;       ERROR MESSAGE IS PREPARED (BUT NOT PRINTED IN THE
;       ROUTINE) AND THEN RETURNS TO THE LOCATION FOLLOWING THE
;       CALL.  IF INTERRUPT OCCURS THE RETURN IS BUMPED BY 2.
;       NORMALLY AN ERROR CALL WILL BE IN THE LOCATION AFTER
;       THE CALL TO INTERRUPT WAIT.

;       CALL:   TWAT16 THROUGH TWAT159, TWAT1S, TWAT2S,
;       TWAT8S, AND TWAT1M

;*****************************************************************
;"L" REGISTER LOADING ROUTINE
;*****************************************************************

;       THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
;       THE "L" REGISTERS L.CS1-L.DCYL.  L.MR1 IS NOT LOADED
;       IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED FOR AN
;       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 TRANSITIONS
```

```
                        .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
                .CHKOP                  ;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
        BECAJSE FAILURE TO INITIALIZE CORRECTLY  IS  FATAL  TO
```

```
                        THE TEST.

                        CALL:   TSSINIT

;*******************************************************************
;WORD COUNT AT END OF OPERATION CHECK
;*******************************************************************

                        THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE
                        FOR  THE  WORD  COUNT  AGAINST THE SUPPLIED VALUE.  IF
                        UNEQUAL, THE ERROR FLAG (WCERR)  IS  SET  IN  GROUP  4
                        ERROR FLAGS (GRP4ER)

                        CALL:   JSR     R4,CHKWC
                                .WORD           ;EXPECTED WC VALUE

;*******************************************************************
;BUS  ADDRESS AT END OF OPERATION CHECK
;*******************************************************************

                        THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT  THE
                        END  OF  A  TRANSFER BY USING THE INITIAL BUS ADDRESS,
                        ADDING IN THE INITIAL WORD COUNT, AND SUBTRACTING  ANY
                        RESIDUAL  WORD  COUNT.   IF  THIS COMPUTED BA DOES NOT
                        EQUAL THE CONTENTS OF RKBA AN ERROR  FLAG  (BAERR)  IS
                        SET IN GROUP 4 ERROR FIELD (GRP4ER)

                        IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED
                        BUS ADDRESS IS THE STARTING BUS ADDRESS.

                        CALL:   JSR     R4,CHKBA

;*******************************************************************
;CYLINDER,TRACK,SECTOR TEST AT END OF OPERATION
;*******************************************************************

                        THIS ROUTINE CHECKS THAT THE CONTENTS  OF  THE  RKDCYL
                        AND RKDA ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE
                        END  OF  THE  OPERATION.   THE  CONTENTS  OF  THE LOAD
                        REGISTER  STORAGE  ARE  COUNTED ON TO HAVE THE INITIAL
                        VALUES TO MAKE THE NECESSARY CALCULATION.

                        ALL THREE VALUES ARE GENERATED AND STORED IN  EXPECTED
                        VALUES  STORAGE  EXPCYL,  EXPTRK,  EXPSEC.   ALL 3 ARE
                        CHECKED  AND  IF  ONE   OR   MORE   ARE   WRONG,   THE
                        CORRESPONDING BIT IN THE ERROR FLAGS FIELD (GRP4ER) IS
                        SET.

                        CALL:   JSR     R4,CHKCTS

;*******************************************************************
;OPERATION CHECK ROUTINE
;*******************************************************************
```

```
THIS IS WHERE ALL HARDWARE ERROR INDICATORS  AND  SOME
SOFTWARE  ERRORS  ARE  CHECKED.  THE GENERAL PROCEDURE
FLOW IS AS FOLLOWS:  THE ROUTINE IS CALLED WITH A TRAP
(TCHKOP).   THE  LOCATION FOLLOWING THE TRAP CALL WILL
HAVE AN ERROR TRAP WHICH THE ROUTINE WILL BYPASS IF NO
ERROR  IS  FOUND.   IF AN ERROR IS DETECTED, THE ERROR
TRAP CALL IS MODIFIED BY THIS ROUTINE  SUCH  THAT  THE
ERROR  TABLE  ITEM  WILL  BE  THE  PROPER ITEM FOR THE
FORMAT REQUIRED BY THIS ERROR.  THE ERROR TRAP WILL BE
MADE EITHER ERROR 4,5,6, 7, OR 10.  REFER TO THE ERROR
ITEM TABLE FOR A DESCRIPTION OF THE FORMAT  AND  WHICH
ERRORS ARE DISPLAYED IN WHAT FORMAT.

FOR NO EXPECTED ERRORS:
CALL:    TCHKOP

FOR EXPECTED ERRORS:
CALL:    TCHKWE
         .WORD           ;GROUP 1 EXPECTED ERRORS
         .WORD           ;GROUP 2 EXPECTED ERRORS
         .WORD           ;GROUP 3 EXPECTED ERRORS

WHERE EACH BIT IN THE THREE WORDS FOLLOWING  THE  CALL
REPRESENT  A  SPECIFIC ERROR.  THE BIT ASSIGNMENTS ARE
GIVEN BELOW:

        GROUP 1 ERRORS:
        BIT 0 - CONTROLLER DETECTED DRIVE BUS
                PARITY ERROR
        BIT 1 - SEEK INCOMPLETE
        BIT 2 - NON-EXECUTABLE DRIVE FUNCTION
        BIT 3 - DRIVE DETECTED DRIVE BUS PARITY ERROR
        BIT 4 - FORMAT ERROR
        BIT 5 - DRIVE TYPE ERROR
        BIT 6 - AC LOW ERROR
        BIT 7 - SPEED LOSS ERROR
        BIT 8 - DRIVE OFF TRACK ERROR
        BIT 9 - CYLINDER OVERFLOW ERROR
        BIT 10 - ILLEGAL DISK ADDRESS ERROR
        BIT 11 - WRITE LOCK ERROR
        BIT 12 - DRIVE TIMING ERROR
        BIT 13 - NO CERR WITH OTHER ERROR SET ERROR
        BIT 14 - DRIVE UNSAFE ERROR
        BIT 15 - CERR BUT NO OTHER ERROR SET ERROR

        GROUP 2 ERRORS:
        BIT 0 - ECC HARD ERROR
        BIT 1 - DATA CHECK ERROR
        BIT 2 - WRITE CHECK ERROR
        BIT 3 - DATA LATE ERROR
        BIT 4 - OPERATION INCOMPLETE ERROR
        BIT 5 - HEADER VRC ERROR
        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.
```

THE THIRD CALL IS IDENTICAL TO THE SECOND.

DEFINITION OF CONTROL WORD:

BIT 15 - DO COMPARE OPERATION OF IBUFF (SOURCE) TO OBUFF (DESTINATION). EXPECTED VALUES ARE IN OBUFF (DESTINATION.
BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
BIT 10 - CLEAR IBUFF TO PATTERN SELECTED.
BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES. OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16. REFER TO THE PROGRAM LISTING FOR PAT02 THROUGH PAT16.

EXPLANATION OF CALLS:

THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.

THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED FOR DATA GENERATION OR COMPARE AND FOR IBUFF INITALIZATION.

THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.

DESCRIPTION:

THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:

A. BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8) OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK ARE LISTED THEREIN (BIT 9).

```
1740
1741
1742
1743                                    B.  COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT 15).
1744                                        THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA.
1745                                        A HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT
1746                                        7) WHICH WILL CAUSE THE COMPARE TO BE LIMITED TO
1747                                        74(8) OR 102(8) WORDS OF HEADERS.  THE LENGTH
1748                                        DEPENDS ON THE FORMAT BIT THAT WAS LAST SPECIFIED
1749                                        IN L.CS1.  THE HEADERS MAY BE BUILT BEFORE THE
1750                                        COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8
1751                                        OR 9).  DATA CAN ALSO BE GENERATED BEFORE THE
1752                                        COMPARE (NON-ZERO BITS 6-0).
1753
1754                                    C.  RESUME COMPARE OPERATION.  IF A COMPARE OPERATION
1755                                        DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO
1756                                        CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE
1757                                        CAN BE RESUMED.  THIS IS DONE BY CALLING GENCOM
1758                                        WITH BIT 14 SET IN THE CONTROL WORD.
1759
1760                                    D.  DATA GENERATION OR COMPARE USING MEMORY
1761                                        MANAGEMENT.  MEMORY MANAGEMENT CAN BE INVOKED FOR
1762                                        EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH.  IN
1763                                        THIS MANNER, DATA GENERATION CAN BE MADE TO PLACE
1764                                        DATA ANYWHERE IN AVAILABLE MEMORY.  LIKEWISE DATA
1765                                        COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO ANY
1766                                        AREA OF AVAILABLE MEMORY.
1767
1768                            ;*******************************************************
1769                            ;PHASE LOCKED LOOP CLOCK ADJUSTMENT ROUTINE
1770                            ;*******************************************************
1771
1772                                    THIS ROUTINE IS ENTERED VIA START LOCATION 220(8).
1773                                    THE PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE
1774                                    INTERNAL PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK
1775                                    ADJUST ROUTINE.  THE ROUTINE SELECTS THE FIRST
1776                                    AVAILABLE DRIVE AND SETS AND RESETS THE DIAGNOSTIC
1777                                    MODE BIT IN RKMR1.  INSTRUCTIONS ON WHERE TO SCOPE AND
1778                                    WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
1779
1780                                    THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
1781                            %
1782                            .NLIST  CND,MD,MC,TOC
1783                            .LIST   ME
1784                            .ENABL  ABS,AMA
1785  167400                    $SWR=   167400
1786  000001                    $TN=    1
1787                            .TITLE  RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
1788                            ;*COPYRIGHT (C) 1976
1789                            ;*DIGITAL EQUIPMENT CORP.
1790                            ;*MAYNARD, MASS. 01754
1791                            ;*
1792                            ;*PROGRAM BY MARV TEGROTENHUIS
1793                            ;*
1794                            ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
1795                            ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
```

```
                                ;*
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        12            ABORT 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                          CR_F=   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
```

```
1908    000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1909    000014          TBITVEC=14              ;;"T" BIT
1910    000014          TRTVEC= 14              ;;TRACE TRAP
1911    000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1912    000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1913    000024          PWRVEC= 24              ;;POWER FAIL
1914    000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1915    000034          TRAPVEC=34              ;;"TRAP" TRAP
1916    000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
1917    000064          TPVEC=  64              ;;TTY PRINTER VECTOR
1918    000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1919                    .SBTTL  MEMORY MANAGEMENT DEFINITIONS
1920
1921                    ;*KT11 VECTOR ADDRESS
1922
1923    000250          MMVEC=  250
1924
1925                    ;*KT11 STATUS REGISTER ADDRESSES
1926
1927    177572          SR0=    177572
1928    177574          SR1=    177574
1929    177576          SR2=    177576
1930    172516          SR3=    172516
1931
1932                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1933
1934    172300          KIPDR0= 172300
1935    172302          KIPDR1= 172302
1936    172304          KIPDR2= 172304
1937    172306          KIPDR3= 172306
1938    172310          KIPDR4= 172310
1939    172312          KIPDR5= 172312
1940    172314          KIPDR6= 172314
1941    172316          KIPDR7= 172316
1942
1943                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
1944
1945    172340          KIPAR0= 172340
1946    172342          KIPAR1= 172342
1947    172344          KIPAR2= 172344
1948    172346          KIPAR3= 172346
1949    172350          KIPAR4= 172350
1950    172352          KIPAR5= 172352
1951    172354          KIPAR6= 172354
1952    172356          KIPAR7= 172356
1953
1954    000210          AVECT1= 210                     ;DEFINE RK611 VECTOR INTERRUPT
1955    000240          APRIOR= PR5                     ;DEFINE RK611 PRIORITY
1956    177440          ABASE=  177440                  ;DEFINE RK611 BASE BUS ADDRESS
1957
1958                    .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
1959
1960    000000          RKCS1=  0                       ;CONTROL AND STATUS REGISTER 1
1961    000002          RKWC=   2                       ;WORD COUNT REGISTER
1962    000004          RKBA=   4                       ;BUS ADDRESS REGISTER
1963    000006          RKDA=   6                       ;DESIRED TRACK SECTOR REGISTER
```

```
1964            000010                  RKCS2=  10                      ;CONTROL AND STATUS REGISTER 2
1965            000012                  RKDS=   12                      ;DRIVE STATUS REGISTER
1966            000014                  RKER=   14                      ;ERROR REGISTER
1967            000016                  RKASOF= 16                      ;ATTENTION SUMMARY AND OFFSET REGISTER
1968            000020                  RKDCYL= 20                      ;DESIRED CYLINDER REGISTER
1969            000024                  RKDB=   24                      ;DATA BUFFER
1970            000026                  RKMR1=  26                      ;MAINTENANCE REGISTER 1
1971            000034                  RKMR2=  34                      ;MAINTENANCE REGISTER 2
1972            000036                  RKMR3=  36                      ;MAINTENANCE REGISTER 3
1973            000030                  RKECPS= 30                      ;ECC POSITION INFORMATION
1974            000032                  RKECPT= 32                      ;ECC PATTERN INFORMATION
1975            000022                  RKSPAR= 22                      ;SPARE REGISTER
1976
1977                                    .SBTTL  DRIVE COMMANDS
1978
1979            000101                  SELDRV= 101                     ;SELECT DRIVE
1980            000103                  PACK=   103                     ;PACK ACKNOWLEDGE
1981            000105                  CLEAR=  105                     ;DRIVE CLEAR
1982            000107                  UNLOAD= 107                     ;UNLOAD
1983            000111                  SRTSPL= 111                     ;START SPINDLE
1984            000113                  RECAL=  113                     ;RECALIBRATE
1985            000115                  OFFSET= 115                     ;OFFSET
1986            000117                  SEEK=   117                     ;SEEK
1987            000121                  RDDATA= 121                     ;READ DATA
1988            000123                  WRDATA= 123                     ;WRITE DATA
1989            000125                  RDHEAD= 125                     ;READ HEADER
1990            000127                  WRHEAD= 127                     ;WRITE HEADER AND DATA
1991            000131                  WRTCHK= 131                     ;WRITE CHECK
1992            000300                  INTR=   300                     ;GENERATE INTERRUPT TO CPU
1993
1994                                    .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
1995
1996            000001                  GO=     BIT0                    ;GO BIT
1997            000100                  IE=     BIT6                    ;INTERRUPT ENABLE
1998            000200                  RDY=    BIT7                    ;CONTROLLER READY
1999            000400                  BA16=   BIT8                    ;BUS ADDRESS BIT 16
2000            001000                  BA17=   BIT9                    ;BUS ADDRESS BIT 17
2001            002000                  CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06)
2002            004000                  CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
2003                                                                    ;  DRIVE RESPONSE
2004            010000                  CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
2005            020000                  SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
2006            040000                  DI=     BIT14                   ;DRIVE INTERRUPT
2007            100000                  CERR=   BIT15                   ;CONTROLLER ERROR
2008            100000                  CCLR=   BIT15                   ;CONTROLLER CLEAR
2009
2010                                    .SBTTL  CONTROL AND STATUS REGISTER 2 BITS
2011
2012            000007                  DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
2013            000010                  RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
2014            000020                  BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
2015            000040                  SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
2016            000100                  IR=     BIT6                    ;INPUT READY
2017            000200                  OR=     BIT7                    ;OUTPUT READY
2018            000400                  UFE=    BIT8                    ;UNIT FIELD ERROR
2019            001000                  MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
```

```
2020            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
2021            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
2022            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
2023            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
2024            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
2025            100000          DLT=    BIT15                   ;DATA LATE ERROR
2026
2027                            .SBTTL  ERROR REGISTER BIT DEFINITION
2028
2029            000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
2030            000002          SKI=    BIT1                    ;SEEK INCOMPLETE
2031            000004          NXF=    BIT2                    ;NON-EXECUTABLE DRIVE FUNCTION
2032            000010          DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
2033            000020          FMTE=   BIT4                    ;FORMAT ERROR
2034            000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
2035            000100          ECH=    BIT6                    ;ECC HARD
2036            000200          BSE=    BIT7                    ;BAD SECTOR ERROR
2037            000400          HVRC=   BIT8                    ;HEADER VRC ERRROR
2038            001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
2039            002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
2040            004000          WLE=    BIT11                   ;WRITE LOCK ERROR
2041            010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
2042            020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
2043            040000          UNS=    BIT14                   ;DRIVE UNSAFE
2044            100000          DCK=    BIT15                   ;DATA CHECK
2045
2046                            .SBTTL  STATUS REGISTER BIT DEFINITION
2047
2048            000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
2049                                                            ;  THIS BIT IS RESET)
2050            000004          OFST=   BIT2                    ;DRIVE OFFSET
2051            000010          ACLO=   BIT3                    ;AC LOW
2052            000020          SPDLSS= BIT4                    ;SPEED LOSS
2053            000040          DROT=   BIT5                    ;DRIVE OFF TRACK
2054            000100          VV=     BIT6                    ;VOLUME VALID
2055            000200          DRDY=   BIT7                    ;DRIVE READY
2056            000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
2057            004000          WRL=    BIT11                   ;WRITE LOCK
2058            020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
2059            040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
2060            100000          SVAL=   BIT15                   ;STATUS VALID
2061
2062                            .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION
2063
2064            000017          MESMSK= 17                      ;MESSAGE MASK
2065
2066            000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
2067            000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
2068            000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
2069            000200          MIND=   BIT7                    ;MAINTENANCE INDEX
2070            000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
2071            001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
2072            002000          MEWD=   BIT10                   ;MAINTENACNE ENCODED WRITE DATA
2073            004000          PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
2074            010000          PCD=    BIT12                   ;PRECOMPENSATION DELAY
2075            020000          ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
```

```
2076            040000                  WRTGAT= BIT14                   ;WRITE GATE
2077            100000                  RDGATE= BIT15                   ;READ GATE
2078
2079                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
2080
2081            000040                  S.DRA=  BIT5                    ;DRIVE AVAILIABLE
2082            000100                  S.VV=   BIT6                    ;VOLUME VALID
2083            000200                  S.DRY=  BIT7                    ;DRIVE READY
2084            000400                  S.TYPE= BIT8                    ;DRIVE TYPE
2085            001000                  S.FORM= BIT9                    ;DRIVE FORMAT
2086            002000                  S.OFF=  BIT10                   ;OFFSET
2087            004000                  S.WRL=  BIT11                   ;WRITE LOCK
2088            010000                  S.SPIN= BIT12                   ;SPINDLE ON
2089            020000                  S.PIP=  BIT13                   ;POSITIONING IN PROGRESS
2090            040000                  S.DSC=  BIT14                   ;DRIVE STATUS CHANGE
2091
2092                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
2093
2094            000040                  S.ICYL= BIT5                    ;ILLEGAL CYLINDER ADDRESS
2095            000100                  S.ACLO= BIT6                    ;AC LOW
2096            000200                  S.FLT=  BIT7                    ;DRIVE FAULT
2097            000400                  S.ILF=  BIT8                    ;ILLEGAL FUNCTION
2098            001000                  S.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
2099            002000                  S.SKI=  BIT10                   ;SEEK INCOMPLETE
2100            004000                  S.WLE=  BIT11                   ;WRITE LOCK ERROR
2101            010000                  S.SPLS= BIT12                   ;SPEED LOSS
2102            020000                  S.DROT= BIT13                   ;DRIVE OFF TRACK
2103            040000                  S.UNS=  BIT14                   ;DRIVE UNSAFE
2104
2105                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
2106
2107            000020                  S.XDOK= BIT4                    ;TRANSDUCER OK
2108            000040                  S.HDHM= BIT5                    ;HEADS HOME
2109            000100                  S.BRHM= BIT6                    ;BRUSHES HOME
2110            000200                  S.DOOR= BIT7                    ;DOOR INTERLOCKED
2111            000400                  S.CART= BIT8                    ;CARTRAGE INTERLOCK
2112            001000                  S.SPOK= BIT9                    ;SPEED OK
2113            002000                  S.FWD=  BIT10                   ;FORWARD
2114            004000                  S.REV=  BIT11                   ;REVERSE
2115            010000                  S.LOAD= BIT12                   ;HEADS LOADING
2116            020000                  S.RTZ=  BIT13                   ;RETURN TO ZERO
2117            040000                  S.UNLD= BIT14                   ;HEADS UNLOADING
2118
2119                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
2120
2121            000020                  S.SECT= BIT4                    ;SECTOR ERROR
2122            000040                  S.WCLK= BIT5                    ;WRITE CLOCK AND NO WRITE GATE
2123            000100                  S.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
2124            000200                  S.HDFL= BIT7                    ;HEAD FAULT
2125            000400                  S.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
2126            001000                  S.XERR= BIT9                    ;INDEX ERROR
2127            002000                  S.DIB=  BIT10                   ;DIBIT ERROR
2128            004000                  S.PLO=  BIT11                   ;PLO ERROR
2129            010000                  S.NMOV= BIT12                   ;SEEK AND NO MOTION
2130            020000                  S.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
2131            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
                    017770              M.SER=  17770                   ;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 "MOV .+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                $40CAT: HALT                            ;;HALT
000002  000737                        BR      .-100                   ;;BRANCH TO 177700 & TIME OUT (NOT ON
                                                                        ;;11/05)
000004  001740                        .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  001740                JMP     @#START ;;GO TO START OF PROGRAM
000204  000137  001730                JMP     RESTRT                  ;JUMP TO RESTART ROUTINE
                    000214              .=214
000214  000137  001720                JMP     PARM                    ;JUMP TO OPERATOR ASSIGNED PARMETERS
                    000220              .=220
000220  000137  001710                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
```

```
                                ;INTERFACE SPEC.

                                $APTHD:
000224                          $HIPTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000224  000000                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000226  001276                  $TSTM:  .WORD   20.     ;;RUN TIM OF LONGEST TEST
000230  000024                  $PASTM: .WORD   60.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000232  000074                  $UNITM: .WORD   480.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000234  000340                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE (WORDS)
000236  000001
```

```
                                .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
```

```
2253  001224  000000                $TMP1:   .WORD   0              ;;USER DEFINED
2254  001226  000000                $TMP2:   .WORD   0              ;;USER DEFINED
2255  001230  000000                $TMP3:   .WORD   0              ;;USER DEFINED
2256  001232  000000                $TMP4:   .WORD   0              ;;USER DEFINED
2257  001234  000000                $TMP5:   .WORD   0              ;;USER DEFINED
2258  001236  000000                $TMP6:   .WORD   0              ;;USER DEFINED
2259  001240  000000                $TMP7:   .WORD   0              ;;USER DEFINED
2260  001242  000000                $TMP10:  .WORD   0              ;;USER DEFINED
2261  001244  000000                $TMP11:  .WORD   0              ;;USER DEFINED
2262  001246  000000                $TMP12:  .WORD   0              ;;USER DEFINED
2263  001250  000000                $TMP13:  .WORD   0              ;;USER DEFINED
2264  001252  000000                $TMP14:  .WORD   0              ;;USER DEFINED
2265  001254  000000                $TMP15:  .WORD   0              ;;USER DEFINED
2266  001256  000000                $TMP16:  .WORD   0              ;;USER DEFINED
2267  001260  000000                $TMP17:  .WORD   0              ;;USER DEFINED
2268  001262  000000                $TIMES:  0                      ;;MAX. NUMBER OF ITERATIONS
2269  001264  000000                $ESCAPE: 0                      ;;ESCAPE ON ERROR ADDRESS
2270  001266  177607  000377        $BELL:   .ASCIZ  <207><377><377> ;;CODE FOR BELL
2271  001272     077                $QUES:   .ASCII  /?/            ;;QUESTION MARK
2272  001273     015                $CRLF:   .ASCII  <15>           ;;CARRIAGE RETURN
2273  001274  000012                $LF:     .ASCIZ  <12>           ;;LINE FEED
2274                                ;;*****************************************************************
2275                                .SBTTL  APT MAILBOX-ETABLE
2276
2277                                ;;*****************************************************************
2278                                .EVEN
2279  001276                        $MAIL:                          ;;APT MAILBOX
2280  001276  000000                $MSGTY: .WORD    AMSGTY         ;;MESSAGE TYPE CODE
2281  001300  000000                $FATAL: .WORD    AFATAL         ;;FATAL ERROR NUMBER
2282  001302  000000                $TESTN: .WORD    ATESTN         ;;TEST NUMBER
2283  001304  000000                $PASS:  .WORD    APASS          ;;PASS COUNT
2284  001306  000000                $DEVCT: .WORD    ADEVCT         ;;DEVICE COUNT
2285  001310  000000                $UNIT:  .WORD    AUNIT          ;;I/O UNIT NUMBER
2286  001312  000000                $MSGAD: .WORD    AMSGAD         ;;MESSAGE ADDRESS
2287  001314  000000                $MSGLG: .WORD    AMSGLG         ;;MESSAGE LENGTH
2288  001316                        $ETABLE:                        ;;APT ENVIRONMENT TABLE
2289  001316     000                $ENV:   .BYTE    AENV           ;;ENVIRONMENT BYTE
2290  001317     000                $ENVM:  .BYTE    AENVM          ;;ENVIRONMENT MODE BITS
2291  001320  000000                $SWREG: .WORD    ASWREG         ;;APT SWITCH REGISTER
2292  001322  000000                $USWR:  .WORD    AUSWR          ;;USER SWITCHES
2293  001324  000000                $CPUOP: .WORD    ACPUOP         ;;CPU TYPE,OPTIONS
2294                                ;*                              BITS 15-11=CPU TYPE
2295                                ;*                                   11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2296                                ;*                                   11/70=06,PDQ=07,Q=10
2297                                ;*                              BIT 10=REAL TIME CLOCK
2298                                ;*                              BIT  9=FLOATING POINT PROCESSOR
2299                                ;*                              BIT  8=MEMORY MANAGEMENT
2300  001326     000                $MAMS1: .BYTE    AMAMS1         ;;HIGH ADDRESS,M.S. BYTE
2301  001327     000                $MTYP1: .BYTE    AMTYP1         ;;MEM. TYPE,BLK#1
2302                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
2303                                ;*                                   900 NSEC CORE=001
2304                                ;*                                   300 NSEC BIPOLAR=002
2305                                ;*                                   500 NSEC MOS=003
2306  001330  000000                $MADR1: .WORD    AMADR1         ;;HIGH ADDRESS,BLK#1
2307                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2308  001332     000                $MAMS2: .BYTE    AMAMS2         ;;HIGH ADDRESS,M.S. BYTE
```

```
 2309  001333  000          $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
 2310  001334  000000       $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
 2311  001336  000          $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
 2312  001337  000          $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
 2313  001340  000000       $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
 2314  001342  000          $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
 2315  001343  000          $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
 2316  001344  000000       $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
 2317  001346  000210       $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
 2318  001350  000000       $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
 2319  001352  177440       $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
 2320  001354  000000       $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
 2321  001356  000000       $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
 2322  001360               $ETEND:
 2323                       .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          EM1N:   .WORD   0
001362  056027                  DH001
001364  057400                  DT001
001366  057472                  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    RK.WC
                                        ;*      T.BA    T.WC
                                        
001370  000000                          EM2N:   .WORD   0
001372  000000                          DH2N:   .WORD   0
001374  057406                                  DT002
001376  057476                                  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  056055                          DH002A
001404  057356                                  DT003
001406  057516                                  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.
                                        ;*
```

```
                        ;* 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
                        ;*      $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.WA
                        ;*        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
                        
001410  000000          EM4N:   .WORD   0
001412  000000          DH4N:   .WORD   0
001414  057406                  DTO04
001416  057526                  DFO04

                        ;*      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.

001420  000000          EM5N:   .WORD   0
001422  000000          DH5N:   .WORD   0
```

```
2492  001424  057406                        DT005
2493  001426  057556                        DF005
2494
2495                                  ;*      ERROR ITEM 6
2496                                  ;*      (DESCRIPTION OF ERROR)
2497                                  ;*      ERROR IN OPERATION
2498                                  ;*      (DESCRIPTION OF OPERATION)
2499                                  ;*      TST NUM ERR PC  DRIVE
2500                                  ;*      $TESTN  $ERRPC  DRVNUM
2501                                  ;*        RKCS1  RKCS2  RKDS  RKER  RKASOF  RKDCYL  RKDA
2502                                  ;*        T.CS1  T.CS2  T.DS  T.ER  T.ASOF  T.DCYL  T.DA
2503                                  ;*      RKBA  RKWC
2504                                  ;*      T.BA  T.WC
2505                                  ;*
2506                                  ;*      THE ERRORS REPORTED BY THIS FORMAT ARE:
2507                                  ;*      DATA CHECK
2508                                  ;*      WRITE CHECK
2509                                  ;*      ECC HARD
2510                                  ;*      DATA LATE
2511                                  ;*      OPERATION INCOMPLETE
2512                                  ;*      HEADER VRC ERROR
2513                                  ;*      BAD SECTOR ERROR
2514
2515  001430  000000                  EM6N:   .WORD   0
2516  001432  000000                  DH6N:   .WORD   0
2517  001434  057406                          DT006
2518  001436  057612                          DF006
2519
2520                                  ;*      ERROR ITEM 7
2521                                  ;*      (DESCRIPTION OF ERROR)
2522                                  ;*      ERROR IN OPERATION
2523                                  ;*      (DESCRIPTION OF OPERATION)
2524                                  ;*      TST NUM ERR PC  DRIVE
2525                                  ;*      $TESTN  $ERRPC  DRVNUM
2526                                  ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF
2527                                  ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF
2528                                  ;*
2529                                  ;*      THE ERRORS REPORTED BY THIS FORMAT ARE:
2530                                  ;*      NON-EXISTANT DRIVE
2531                                  ;*      NON-EXISTANT MEMORY
2532                                  ;*      CONTROLLER TIME OUT
2533                                  ;*      UNIT FIELD ERROR
2534                                  ;*      MULTIPLE DRIVE SELECT
2535                                  ;*      PROGRAMMING ERROR
2536                                  ;*      UNIBUS PARITY ERROR
2537                                  ;*      ILLEGAL FUNCTION CODE
2538                                  ;*
2539                                  ;*      DESCRIPTON OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR ILLEGAL
2540
2541  001440  000000                  EM7N:   .WORD   0
2542  001442  000000                  DH7N:   .WORD   0
2543  001444  057406                          DT007
2544  001446  057636                          DF007
2545
2546                                  ;*      ERROR ITEM 10
2547                                  ;*      (DESCRIPTION OF ERROR)
```

```
                                        ;*      ERROR AT COMPLETION OF OPERATION
                                        ;*      (DESCRIPITON OF OPERATION)
                                        ;*      TST NUM ERR PC  DRIVE
                                        ;*      $TESTN  $ERRPC  DRVNUM
                                        ;*        EXPT    1S
                                        ;*        $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                  EM1ON:  .WORD   0
001452  056670                          DH010
001454  057456                          DT015
001456  057656                          DF010

                                        ;*      ERROR ITEM 11
                                        ;*      (ERROR INDICATOR OR STATUS BIT)
                                        ;*      NOT SET AS A RESULT OF
                                        ;*      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION
                                        ;*      TST NUN 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  057014                          DH011
001464  057406                          DT010
001466  057676                          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  057043                          DH012
001474  057406                          DT010
001476  057676                          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  057074                          DH013
001504  057406                          DT010
001506  057676                          DF011

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

```
2604  001510  000000                  EM14N:  .WORD   0
2605  001512  057117                          DH014
2606  001514  057406                          DT010
2607  001516  057676                          DF011
2608
2609                                  ;*      ERROR ITEM 15
2610                                  ;*      (HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)
2611                                  ;*      TST NUM ERR PC  DRIVE
2612                                  ;*      $TESTN  $ERRPC  DRVNUM
2613                                  ;*        GOOD    BAD     WORD NUM
2614                                  ;*        $REG10  $REG11  $REG12
2615
2616  001520  000000                  EM15N:  .WORD 0
2617  001522  056055                          DH002A
2618  001524  057456                          DT015
2619  001526  057720                          DF015
2620
2621                                  ;*      ERROR ITEM 16
2622                                  ;*      ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15
2623
2624  001530  000000                          0
2625  001532  000000                          0
2626  001534  057464                          DT015A
2627  001536  057730                          DF016
2628
```

```
                                .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.CB:   .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  060060          BSF24P: .WORD   BS24            ;POINTER TO BAD SECTORS 24 SECTOR MODE
                                                        ;   (FACTORY)
001640  057734          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
001656  000000          OPTFLG: .WORD   0               ;OPTION FLAGS
```

```
                                        DOTST=  BIT0            ;DRIVE 0 TO BE TESTED FLAG
                                        MEMSZB= BIT1            ;MEMORY SIZE REPORT FLAG
                                        MEMPYB= BIT2            ;MEMORY PARITY REPORT FLAG
                                        SRTINS= BIT3            ;START UP INSTRUCTIONS REPORTED FLAG
                                        PARBK0= BIT7            ;PARITY OPTION BANK 0
                                        PARBK1= BIT6            ;PARITY OPTION BANK 0 INTERLEAVED MEM
                                        BSERPT= BIT7            ;BSE HAS BEEN REPORTED
                                        FPFMT=  BIT8            ;FIRST PASS FORMAT SWITCH
                                        LCLKPR= BIT15           ;LINE CLOCK PRESENT

001660  000000                  LCLKTK: .WORD   0               ;LINE CLOCK TICK COUNTER
001662  000000                  INTSET: .WORD   0               ;NON-ZERO IF RK06 INTERRUPT SINCE LAST
                                                                ;CLEAR
001664  000000                  REFMT:  .WORD   0               ;REFORMAT SWITCH FOR HALT

                                ;       THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR
                                ;       HEADER OR DATA COMPARE CONTINUATION PROCESS.
001666  000000                  DESHLD: .WORD   0               ;DESTINATION HOLD
001670  000000                  SRCHLD: .WORD   0               ;SOURCE HOLD
001672  000000                  WRDNUM: .WORD   0               ;WORD NUMBER IN ERROR HOLD
001674  000000                  WRDCNT: .WORD   0               ;WORDS LEFT IN COMPARE HOLD
001676  177546                  KWLADD: .WORD   177546          ;KW11-L ADDRESS
001700  000100                  KWLVEC: .WORD   100             ;KW11-L VECTOR
001702  172100                  MMCSR1: .WORD   172100          ;MM11 ADDRESS
001704  172102                  MMCSR2: .WORD   172102          ;MM11 ADDRESS
001706  000114                  MMVECA: .WORD   114             ;MM11 VECRTOR
```

```
                                        .SBTTL  PROGRAM SETUP
001710  012737  000002  001624  SETCLK: MOV     #2,SRTFLG       ;SET START FLAG FOR CLOCK ADJUST
001716  000412                          BR      START1
001720  012737  000001  001624  PARM:   MOV     #1,SRTFLG       ;SET START FLAG FOR PARMETER START
001726  000406                          BR      START1
001730  012737  177777  001624  RESTRT: MOV     #-1,SRTFLG      ;LOAD START FLAG FOR PARMETER START
001736  000402                          BR      START1
001740  005037  001624          START:  CLR     SRTFLG          ;CLEAR START FLAG
001744  000005                  START1: RESET                   ;RESET THE WHOLE SYSTEM
001746  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
001752  012746  000340                  MOV     #PR7,-(SP)      ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
001756  012746  001764                  MOV     #1$,-(SP)       ;LOAD START OF PROGRAM
001762  000002                          RTI                     ;LOAD PSW

001764  004737  043664          1$:     JSR     PC,$TKINT       ;INITIALIZE KEYBOARD
001770  005037  001664                  CLR     REFMT           ;CLEAR REFORMAT SWITCH
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001774  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002000  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002002  022706  001140                  CMP     #SWR,R6 ;;DONE?
002006  001374                          BNE     .-6             ;;LOOP BACK IF NO
002010  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
002014  012737  031760  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002022  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002030  012737  033000  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002036  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002044  012737  045524  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002052  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
002060  012737  045346  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002066  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002074  013737  031276  031270          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002102  005037  001262                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002106  005037  001264                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002112  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002120  012737  002120  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002126  012737  002126  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002134  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002140  012737  002174  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002146  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002154  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002162  022777  177777  176750          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002170  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
002172  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
```

```
 2768  002174  012716  002202          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
 2769  002200  000002                          RTI
 2770  002202  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
 2771  002210  012737  000174  001142          MOV     #DISPREG,DISPLAY
 2772  002216  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
 2773
 2774  002222  005037  001304                  CLR     $PASS           ;;CLEAR PASS COUNT
 2775  002226  132737  000200  001317          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
 2776  002234  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
 2777  002236  012737  001320  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
 2778  002244                          67$:
 2779                                  .SBTTL  TYPE PROGRAM NAME
 2780                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
 2781  002244  005227  177777                  INC     #-1             ;;FIRST TIME?
 2782  002250  001066                          BNE     68$             ;;BRANCH IF NO
 2783  002252  022737  031432  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
 2784  002260  001462                          BEQ     68$             ;;BRANCH IF YES
 2785  002262  104401  002330                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 2786                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
 2787  002266  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
 2788  002272  001012                          BNE     70$             ;;BRANCH IF YES
 2789  002274  123727  001316  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
 2790  002302  001406                          BEQ     70$             ;;BRANCH IF YES
 2791  002304  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
 2792  002312  001005                          BNE     71$             ;;BRANCH IF NO
 2793  002314  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
 2794  002316  000403                          BR      71$
 2795  002320  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
 2796  002326                          71$:
 2797  002326  000437                          BR      68$             ;;GET OVER THE ASCIZ
 2798                                  ;;69$:  .ASCIZ  <CRLF>*RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC  MAINDEC-11-DZR6K-C*<CRLF>
 2799  002426                          68$:
 2800  002426  132737  000200  001317          BITB    #BIT7,$ENVM     ;TEST IF DO NOT SIZE
 2801  002434  001043                          BNE     3$              ;YES - SKIP
 2802  002436  004737  031452                  JSR     PC,$SIZE
 2803  002442  023727  031756  000740          CMP     $LSTBK,#740     ;MAKE SURE MEMORY IS SUFFICIENT
 2804  002450  103007                          BHIS    2$              ;YES - SKIP
 2805  002452  104401  047613                  TYPE    ,OPR005         ;MESSAGE (NOT ENOUGH MEMORY)
 2806  002456  012737  000001  031270          MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
 2807  002464  000137  031242                  JMP     $EOP
 2808  002470  013700  031756          2$:     MOV     $LSTBK,R0       ;GET LAST BANK
 2809  002474  012701  000006                  MOV     #6,R1           ;SET SHIFT COUNT
 2810  002500  013703  031754                  MOV     $LSTAD,R3       ;GET LAST ADDRESS
 2811  002504  005004                          CLR     R4              ;CLEAR R4 FOR OVERFLOW
 2812  002506  005737  031510                  TST     $KT11           ;MEM MANAGE PRESENT?
 2813  002512  100005                          BPL     23$             ;NO - SKIP
 2814  002514  006100                  22$:    ROL     R0              ;SHIFT BANK LEFT
 2815  002516  006104                          ROL     R4              ;ADD IN CARRY
 2816  002520  005301                          DEC     R1              ;DECREMENT COUNT
 2817  002522  001374                          BNE     22$             ;LOOP IF NOT ZERO
 2818  002524  050003                          BIS     R0,R3           ;SET BANK BITS IN LAST ADDRESS
 2819  002526  112737  000001  001327  23$:    MOVB    #1,$MTYP1       ;FORCE MEMORY TYPE TO 1
 2820  002534  110437  001326                  MOVB    R4,$MAMS1       ;STORE UPPER MEMORY ADDRESS
 2821  002540  010337  001330                  MOV     R3,$MADR1       ;STORE LOWER ADDRESS
 2822  002544  032737  000010  001656  3$:     BIT     #SRTINS,OPTFLG  ;TEST IF ALREDY REPORTED
 2823  002552  001005                          BNE     24$             ;YES - SKIP
```

```
2824  002554  104401  050604                  TYPE    ,OPR016         ;TYPE STARTUP INSTRUCTIONS
2825  002560  052737  000010  001656          BIS     #SRTINS,OPTFLG  ;SET REPORTED FLAG
2826  002566                          24$:
2827  002566  022737  000001  001624          CMP     #1,SRTFLG       ;CHECK IF PARAMETER START
2828  002574  001122                          BNE     15$             ;NO, START TESTING
2829  002576  104401  047504          5$:     TYPE    ,OPR001         ;TYPE "RK611 BUS ADDRESS (  ) ="
2830  002602  013746  001352                  MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
2831  002606  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS,
2832  002610  104401  047533                  TYPE    ,OPR002
2833  002614  104412                          RDOCT                   ;GET VALUE
2834  002616  012637  001222                  MOV     (SP)+,$TMP0
2835  002622  001407                          BEQ     7$              ;CHECK IF <CR>
2836  002624  022737  160000  001222          CMP     #160000,$TMP0   ;CHECK IF IN I/O PAGE
2837  002632  101361                          BHI     5$
2838  002634  013737  001222  001352          MOV     $TMP0,$BASE     ;LOAD NEW BUS ADDRESS
2839  002642  104401  047541          7$:     TYPE    ,OPR003         ;TYPE "RK611 VECTOR ADDRESS (  ) ="
2840  002646  013746  001346                  MOV     $VECT1,-(SP)    ;GET $VECT1 FOR TYPOUT
2841  002652  042716  160000                  BIC     #160000,(SP)    ;CLEAR PRIORITY BITS
2842  002656  104402                          TYPOC
2843  002660  104401  047533                  TYPE    ,OPR002
2844  002664  104412                          RDOCT                   ;GET VALUE
2845  002666  012637  001222                  MOV     (SP)+,$TMP0
2846  002672  001412                          BEQ     10$             ;CHECK IF <CR>
2847  002674  022737  001000  001222          CMP     #1000,$TMP0
2848  002702  101757                          BLOS    7$              ;CHECK IF LEGAL
2849  002704  042737  017777  001346          BIC     #17777,$VECT1   ;CLEAR OLD VECTOR
2850  002712  053737  001222  001346          BIS     $TMP0,$VECT1    ;LOAD NEW VECTOR ADDRESS
2851  002720  104401  047571          10$:    TYPE    ,OPR004         ;TYPE "RK611 PRIORITY (  ) ="
2852  002724  005046                          CLR     -(SP)
2853  002726  113716  001347                  MOVB    $VECT1+1,(SP)
2854  002732  006216                          ASR     (SP)            ;SHIFT 5 BITS RIGHT
2855  002734  006216                          ASR     (SP)
2856  002736  006216                          ASR     (SP)
2857  002740  006216                          ASR     (SP)
2858  002742  006216                          ASR     (SP)
2859  002744  104402                          TYPOC
2860  002746  104401  047533                  TYPE    ,OPR002
2861  002752  104412                          RDOCT                   ;GET VALUE
2862  002754  012637  001222                  MOV     (SP)+,$TMP0
2863  002760  001430                          BEQ     15$             ;CHECK IF <CR>
2864  002762  022737  000007  001222          CMP     #7,$TMP0        ;CHECK IF LEGAL
2865  002770  103753                          BLO     10$
2866  002772  022737  000004  001222          CMP     #4,$TMP0
2867  003000  101347                          BHI     10$
2868  003002  006337  001222                  ASL     $TMP0           ;SHIFT 5 BITS LEFT
2869  003006  006337  001222                  ASL     $TMP0
2870  003012  006337  001222                  ASL     $TMP0
2871  003016  006337  001222                  ASL     $TMP0
2872  003022  006337  001222                  ASL     $TMP0
2873  003026  042737  160000  001347          BIC     #160000,$VECT1+1 ;CLEAR OLD PRIORITY
2874  003034  053737  001222  001347          BIS     $TMP0,$VECT1+1  ;LOAD RK611 PRIORITY
2875  003042  004737  033704          15$:    JSR     PC,OPTTST       ;SETUP PARITY CHECK & CLOCK
2876  003046  013700  001346                  MOV     $VECT1,R0       ;STORE VECTOR FOR USE
2877  003052  042700  160000                  BIC     #160000,R0      ;CLEAR PRIORITY BITS
2878  003056  010037  001620                  MOV     R0,RKVEC
2879  003062  012710  033626                  MOV     #INTHLR,(R0)    ;SETUP INTERRUPT ADDRESS
```

```
 2880  003066  113737  001347  001622          MOVB    $VECT1+1,RKPRI  ;STORE PRIORITY FOR JSE
 2881  003074  013702  001352                  MOV     $BASE,R2        ;SET BASE ADDRESS
 2882  003100  005037  001264                  CLR     $ESCAPE         ;CLEAR ESCAPE
 2883  003104  012746  000000                  MOV     #PRO,-(SP)      ;SET PRIORITY
 2884  003110  012746  003116                  MOV     #16$,-(SP)
 2885  003114  000002                          RTI
 2886  003116                          16$:
 2887
```

```
                                .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.
                                ;;*****************************************************************
003116  000004                  TST1:   SCOPE
003120  012737  000100  001262          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003126  012706  001100          MOV     #STACK,SP       ;CLEAN OFF STACK
003132  012701  000004          MOV     #4,R1           ;SET POINTER TO VECTOR
003136  012146                  MOV     (R1)+,-(SP)     ;STORE OLD VECTOR CONTENTS
003140  011146                  MOV     (R1),-(SP)
003142  012701  000004          MOV     #4,R1           ;RESET POINTER
003146  012721  033620          MOV     #NEXINT,(R1)+   ;SET VECTOR TO NEM TEST HANDLER
003152  012711  000340          MOV     #PR7,(R1)       ;SET PRIORITY
003156  013702  001352          MOV     $BASE,R2        ;SET POINTER TO RK611 BASE ADDRESS
003162  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT COUNTER
003166  012762  000000  000000  MOV     #0,RKCS1(R2)    ;WRITE CS1 TO SEE IN NEM WILL SET
003174  000240                  NOP
003176  000240                  NOP
003200  000240                  NOP
003202  005737  001662          TST     INTSET          ;TEST IF COUNTER IS 0
003206  001406                  BEQ     1$              ;YES - SKIP ERROR REPORT
003210  012737  052216  001360  MOV     #EM1,EMIN       ;MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
003216  104001                  ERROR   1
003220  000137  042442          JMP     CTRHLT          ;GO TO CONTROLLED HALT
003224  012701  000006  1$:     MOV     #6,R1           ;RESTORE VECTOR
003230  012611                  MOV     (SP)+,(R1)
003232  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.
                                ;;*****************************************************************
003234  000004                  TST2:   SCOPE
003236  012737  000100  001262          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003244  012762  005000  000010          MOV     #CLR,RKCS2(R2) ;CLEAR SUBSYSTEM, SPECIFICALLY TO
                                                                ;CLEAR ANY OLD INTERRUPTS
003252  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT COUNTER
003256  012762  000300  000000          MOV     #RDY!IE,RKCS1(R2)       ;WRITE CS1 TO FORCE INTERRUPT
003264  000240                  NOP
003266  000240                  NOP
003270  000240                  NOP
003272  005737  001662          TST     INTSET          ;TEST IF INTERRUPT OCCURRED
003276  001011                  BNE     3$              ;YES - SKIP ERROR REPORT
003300  105737  001103          TSTB    $ERFLG          ;TEST IF ERFLG ALREADY SET. IF SO THE
                                                        ;INTERRUPT WENT TO THE WRONG VECTOR
                                                        ;AND MESSAGE HAS BEEN REPORTED.
003304  001004                  BNE     2$              ;THEREFORE - EXIT
003306  012737  052216  001360  MOV     #EM1,EMIN       ;MESSAGE (NO INTERRUPT)
003314  104001                  ERROR   1
```

```
2944  003316  000137  042442          2$:     JMP     CTRHLT          ;GO TO CONTROLLED HALT
2945  003322                          3$:
2946
2947                                  .SBTTL  **STATUS VALID TESTS
2948
2949
2950                                  ;;*********************************************************************
2951                                  ;*TEST 3        SELECT ALL DRIVES
2952                                  ;*
2953                                  ;*      IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT,
2954                                  ;*      DETERMINE WHAT DRIVES ARE ON-LINE BY
2955                                  ;*      SELECTING ALL DRIVES.  IF NON-EXISTENT DRIVE REPORTED
2956                                  ;*      MAKE SURE STATUS VALID IS RESET.  IF DRIVE
2957                                  ;*      PRESENT MAKE SURE NO ERROR EXISTS, DRIVE
2958                                  ;*      IS CYCLED UP, AND STATJS VALID SET, AND DSC RESET.
2959                                  ;*
2960                                  ;*      IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES
2961                                  ;*      IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE
2962                                  ;*      CYCLED UP, AND STATUS VALID SET.
2963                                  ;*
2964                                  ;*      IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON
2965                                  ;*      THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM
2966                                  ;*      START (214) WAS USED. IF THE AUTOMATIC START (200)
2967                                  ;*      IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204)
2968                                  ;*      WILL RETAIN THE TEST STATUS OF DRIVE 0.
2969                                  ;*
2970                                  ;*      IF THE PARAM START IS USED, THE OPERATOR MUST
2971                                  ;*      EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED
2972                                  ;*      OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO
2973                                  ;*      BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME
2974                                  ;*      VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.
2975                                  ;*
2976                                  ;*      THE DRIVE MUST BE ON-LINE, CYCLED UP, AND WRITE ENABLED.
2977                                  ;*      IF ANY ONE OF THESE CONDITIONS IS NOT TRUE THAT DRIVE
2978                                  ;*      IS NOT TESTED AND IT IS EXPECTED TO BE OFF-LINE. ADDRESSING
2979                                  ;*      THAT DRIVE SHOULD CAUSE NON-EXISTANT DRIVE ERROR.
2980                                  ;*      AT COMPLETION OF THE TEST
2981                                  ;*      A MESSAGE WILL BE GIVEN TO IDENTIFY THE DRIVES TO BE
2982                                  ;*      USED IN TESTING.
2983                                  ;*
2984                                  ;*      NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE
2985                                  ;*            ANY OTHER TEST THAT FOLLOWS.
2986                                  ;*********************************************************************
2987  003322  000004                  TST3:   SCOPE
2988  003324  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
2989  003332  104416                          TSSINIT                 ;CALL SUBSYSTEM CLEAR AND TEST
2990  003334  104003                          ERROR   3
2991
2992  003336  012746  000000                  MOV     #PR0,-(SP)      ;SET PROCESSOR PRIORITY TO ALLOW
2993  003342  012746  003350                  MOV     #1$,-(SP)       ;RK611 INTERRUPTS
2994  003346  000002                          RTI
2995
2996  003350  013701  001620          1$:     MOV     RKVEC,R1        ;GET VECTOR
2997  003354  012721  033626                  MOV     #INTHLR,(R1)+ ;LOAD INTERRUPT VECTOR
2998  003360  012711  000340                  MOV     #PR7,(R1)
2999  003364  012703  001354                  MOV     #$DEVM,R3       ;GET ADDRESS OF DEVICE MAP
```

```
3000  003370  005737  001304          TST     $PASS           ;TEST IF FIRST PASS
3001  003374  001104                  BNE     7$              ;NO - SKIP TO DRIVE SELECT TEST
3002  003376  132737  000200  001317  BITB    #BIT7,$ENVM     ;TEST IF SHOULD SIZE
3003  003404  001402                  BEQ     92$             ;YES - SKIP TO DRIVE SIZING.
3004  003406  000137  003740          JMP     11$
3005  003412  005013          92$:    CLR     (R3)            ;CLEAR DEVICE MAP
3006  003414  123727  000041  000013  CMPB    @#41,#13        ;TEST IF RK06 IS LOAD DEVICE
3007  003422  001066                  BNE     77$             ;NO - SKIP
3008  003424  022737  000001  001624  CMP     #1,SRTFLG       ;WAS START AT PARAM?
3009  003432  001406                  BEQ     2$              ;YES - SKIP
3010  003434  104401  050034          TYPE    ,OPR007         ;NO TEST OF DRIVE 0
3011  003440  042737  000001  001656  BIC     #DOTST,OPTFLG   ;DR FLAG - NO TEST DRIVE 0
3012  003446  000457                  BR      7$
3013  003450  104401  047664  2$:     TYPE    .OPR006         ;MESSAGE - SWAP PACK ON DRIVE OFF LINE.
3014  003454  005037  001610          CLR     L.CS2           ;SET TO DRIVE 0
3015  003460  005037  001232          CLR     $TMP4           ;CLEAR FOR USE AS A SWITCH
3016  003464  012737  000101  001600  3$:     MOV     #SELDRV,L.CS1   ;LOAD FOR SELECT
3017  003472  104417                  TLOADRK                 ;LOAD RK & DO SELECT
3018
3019  003474  104423                  TWAT16                  ;WAIT 16MS FOR COMPLETION
3020  003476  104002                  ERROR   2               ;NOT DONE ON TIME
3021
3022  003500  104420                  TGETRK                  ;GET RK REGISTER
3023  003502  032737  100000  001540  BIT     #CERR,T.CS1     ;TEST IF CERR
3024  003510  001414                  BEQ     5$              ;NO - SKIP
3025  003512  032737  010000  001550  BIT     #NED,T.CS2      ;TEST IF NED
3026  003520  001002                  BNE     4$              ;YES - SKIP
3027
3028  003522  104421                  TCHKOP                  ;CHECK THE OPERATION AND REPORT THE ERROR
3029  003524  104004                  ERROR   4 ;OR5,6,7      ;AFTER THE ERROR IS REPORTED THE TEST
3030                                                          ;IS ABORTED
3031  003526  104401  050034  4$:     TYPE    ,OPR007         ;TYPE NO TEST OF DRIVE 0
3032  003532  042737  000001  001656  BIC     #DOTST,OPTFLG   ;DR FLAG - NO TEST OF DRIVE 0
3033  003540  000422                  BR      7$              ;SKIP OVER WAIT FOR PACK MOUNT
3034  003542  005737  001232  5$:     TST     $TMP4           ;TEST FLAG DRIVE READY HAS RESET
3035  003546  001010                  BNE     6$              ;YES - SKIP TO CHECK IF IT IS SET AGAIN
3036  003550  032737  000200  001552  BIT     #DRDY,T.DS      ;ELSE CHECK READY
3037  003556  001342                  BNE     3$              ;STILL SET - GET STATUS AGAIN
3038  003560  012737  177777  001232  MOV     #-1,$TMP4       ;ELSE SET FLAG TO INDICATE READY WENT LOW
3039  003566  000736                  BR      3$              ;GO GET STATUS AGAIN
3040
3041  003570  032737  000200  001552  6$:     BIT     #DRDY,T.DS      ;TEST IF READY SET AGAIN
3042  003576  001732                  BEQ     3$              ;NO - GO GET STATUS AGAIN
3043  003600  052737  000001  001656  77$:    BIS     #DOTST,OPTFLG   ;ELSE SET DRV 0 TEST FLAG.
3044
3045  003606  005000          7$:     CLR     R0              ;CLEAR FOR DRIVE NUMBER COUNTER
3046  003610  012701  000001          MOV     #1,R1           ;SET BIT 0 AS DRIVE SELECTOR
3047
3048  003614  032737  000001  001656  BIT     #DOTST,OPTFLG   ;TEST DRIVE 0?
3049  003622  001430                  BEQ     9$              ;NO - SKIP
3050
3051  003624  104416          8$:     TSSINIT                 ;INITIALZE SUBSYSTEM
3052  003626  104003                  ERROR   3               ;ERROR IF NOT SUCCESSFUL
3053
3054  003630  010037  001610          MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
3055  003634  012737  000101  001600  MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
```

```
3056  003642  104417                          TLOADRK                 ;LOAD RK REGS
3057
3058  003644  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
3059  003646  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3060
3061  003650  104420                          TGETRK                  ;GET RK REGS
3062  003652  032737  100000  001540          BIT     #CERR,T.CS1     ;ERROR?
3063  003660  001017                          BNE     10$             ;YES - SKIP
3064  003662  032737  000200  001552          BIT     #DRDY,T.DS      ;ELSE TEST IF DRIVE READY
3065  003670  001405                          BEQ     9$              ;NO - SKIP
3066  003672  032737  004000  001552          BIT     #WRL,T.DS       ;ELSE TEST IF WRITE LOCKED
3067  003700  001001                          BNE     9$              ;YES - SKIP
3068
3069  003702  050113                          BIS     R1,(R3) ;SET BIT - DRIVE PRESET IN MAP
3070
3071  003704  005200                  9$:     INC     R0              ;BUMPS TO NEXT DRIVE
3072  003706  006301                          ASL     R1              ;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
3073  003710  032701  000400                  BIT     #BIT8,R1        ;WAS LAST DRIVE DONE?
3074  003714  001743                          BEQ     8$              ;YES - SKIP
3075  003716  000410                          BR      11$             ;ELSE LOOP TO SELECT NEXT DRIVE
3076
3077  003720  032737  010000  001550  10$:     BIT     #NED,T.CS2      ;WAS CERR DUE TO NED?
3078  003726  001366                          BNE     9$              ;YES - BUMP TO NEXT DRIVE
3079
3080  003730  104421                          TCHKOP                  ;ELSE REPORT THE ERRORS
3081  003732  104004                          ERROR   4 ;OR5,6,7
3082  003734  000000                  101$:   .WORD   0
3083  003736  177777                  100$:   .WORD   -1              ;A SWITCH - IT NEVER GETS EXECUTED
3084  003740  005737  003736          11$:    TST     100$            ;TEST SWITCH
3085  003744  100036                          BPL     16$             ;IF PLUS - SKIP DRIVE TEST MESSAGE
3086  003746  005237  003736                  INC     100$            ;ELSE BUMP SWITCH TO PLUS
3087
3088  003752  005713                          TST     (R3)            ;ANY DRIVE AVAILABLE?
3089  003754  001004                          BNE     12$             ;BR IF NOT ZERO
3090  003756  104401  050124                  TYPE    .OPR008         ;ELSE REPORT NO DRIVES AVAILABLE
3091  003762  000137  042442                  JMP     CTRHLT          ;GO TO CONTROLLED HALT
3092
3093  003766  012701  000200          12$:    MOV     #BIT7,R1        ;SET DRIVE SELECTOR FOR DRIVE 7
3094  003772  012700  000007                  MOV     #7,R0           ;SET DRIVE NUMBER TO 7
3095  003776  104401  050207                  TYPE    .OPR009         ;TYPE PREFIX TO DRIVE TEST MESSAGE
3096
3097  004002  030113          13$:    BIT     R1,(R3)         ;TEST IF THIS DRIVE TO BE TESTED
3098  004004  001004                  BNE     15$             ;YES - SKIP
3099
3100  004006  005300          14$:    DEC     R0              ;ELSE DECREMENT DRIVE NUMBER
3101  004010  006201                  ASR     R1              ;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
3102  004012  001373                  BNE     13$             ;IF NOT SHIFTED OUT - LOOP
3103  004014  000412                  BR      16$             ;ELSE GO TO STATUS VALID TEST
3104
3105  004016  010037  003734  15$:    MOV     R0,101$         ;PUT DRIVE NUMBER IN TYPE LOCATION
3106  004022  052737  000060  003734  BIS     #BIT4!BIT5,101$ ;MAKE IT ASCIZ
3107  004030  104401                  TYPE                    ;TYPE IT
3108  004032  003734                  101$
3109  004034  104401  047501          TYPE    ,SPACE2         ;TYPE SOME SPACES
3110  004040  000762                  BR      14$             ;LOOP
3111
```

```
3112  004042  005000                  16$:  CLR     R0              ;CLEAR DRIVE NUMBER COUNTER
3113  004044  012701  000001              MOV     #1,R1           ;SET DRIVE SELECTOR TO DRIVE 0
3114  004050  012737  177777  001240      MOV     #-1,$TMP7       ;SET $TMP7 NEGATIVE
3115  004056  012737  177777  001630      MOV     #-1,DRVBIT      ;SET DRIVE SELECT BIT NEGATIVE
3116
3117  004064  104416                  17$:  TSSINIT               ;CLEAR SUBSYSTEM
3118  004066  104003                        ERROR   3             ;ERROR FOR BAD CLEAR
3119
3120  004070  010037  001610              MOV     R0,L.CS2        ;SET DRIVE SELECT
3121  004074  010037  001626              MOV     R0,DRVNUM       ;SET DRIVE NUMBER
3122  004100  012737  000101  001600      MOV     #SELDRV,L.CS1   ;SET FOR DRIVE SELECT
3123
3124  004106  104417                      TLOADRK               ;LOAD RK REGS
3125  004110  104423                      TWAIT16               ;WAIT FOR INTERRUPT
3126  004112  104002                      ERROR   2             ;ERROR TO SLOW/NOT COMPLETE
3127  004114  030113                      BIT     R1,(R3)       ;WAS THAT DRIVE AVAILABLE
3128  004116  001023                      BNE     19$           ;YES - SKIP
3129
3130  004120  104422                      TCHKWE                ;CHECK THAT ERROR OCCURRED
3131  004122  000000                      .WORD   0             ;NONE OF GROUP 1
3132  004124  000000                      .WORD   0             ;NONE OF GROUP 2
3133  004126  000001                      .WORD   1             ;NED IN GROUP 3
3134  004130  104004                      ERROR   4 ;OR5,6,7    ;ERROR IF NO ERROR OR WRONG ERROR
3135
3136  004132  032737  100000  001552      BIT     #SVAL,T.DS    ;DID STATUS VALID RESET?
3137  004140  001404                      BEQ     18$           ;YES - SKIP
3138  004142  012737  054340  001400      MOV     #EM47,EM3N    ;ELSE MESSAGE (SVAL NOT RESET WITH NED)
3139  004150  104003                      ERROR   3
3140  004152  005200                  18$:  INC     R0            ;BUMP TO NEXT DRIVE
3141  004154  006301                      ASL     R1            ;SHIFT DRIVE SELECT BIT
3142  004156  032701  000400              BIT     #BIT8,R1      ;ALL DRIVES CHECKED
3143  004162  001740                      BEQ     17$           ;NO - GET NEXT DRIVE
3144  004164  000457                      BR      21$           ;ELSE EXIT
3145  004166  104421                  19$:  TCHKOP                ;CHECK NO ERRORS SET
3146  004170  104004                      ERROR   4 ;OR5,6,7    ;REPORT ALL ERRORS
3147
3148  004172  032737  100000  001552      BIT     #SVAL,T.DS    ;CHECK SVAL SET
3149  004200  001004                      BNE     20$           ;YES - SKIP
3150  004202  012737  054421  001400      MOV     #EM48,EM3N    ;MESSAGE (NO SVAL FROM EXISTANT DR)
3151  004210  104003                      ERROR   3
3152
3153  004212  012737  000103  001600  20$:  MOV     #PACK,L.CS1   ;ELSE SET TO DO PACK ACK
3154  004220  104417                      TLOADRK               ;LOAD RK
3155
3156  004222  104423                      TWAIT16               ;WAIT FOR INTERRUPT
3157  004224  104002                      ERROR   2             ;TO SLOW/NOT COMPLETE ERROR
3158
3159  004226  104421                      TCHKOP                ;CHECK FOR ANY ERRORS
3160  004230  104004                      ERROR   4 ;OR5,6,7    ;YES - REPORT & ABORT TEST
3161
3162  004232  032737  000100  001552      BIT     #VV,T.DS      ;DID VV SET
3163  004240  001005                      BNE     22$           ;YES - SKIP
3164  004242  012737  054303  001400      MOV     #EM46,EM3N    ;MESSAGE (VV DID NOT SET/
3165  004250  104003                      ERROR   3
3166  004252  000737                      BR      18$
3167
```

```
3168  004254  032737  040000  001552  22$:    BIT     #DSC,T.DS       ;TEST IF DSC RESET
3169  004262  001410                          BEQ     23$             ;YES - SKIP
3170  004264  012737  055375  001470          MOV     #EMDSC,EM12N
3171  004272  012737  055635  057702          MOV     #EMSCLR,DF011A
3172  004300  104003                          ERROR   3               ;"DSC NOT RESET RESULT OF SUBSYS CLEAR"
3173  004302  000723                          BR      18$
3174
3175  004304  005737  001630          23$:    TST     DRVBIT          ;TEST IF DRVBIT IS NEGATIVE
3176  004310  100320                          BPL     18$             ;NO - SKIP
3177  004312  010137  001630                  MOV     R1,DRVBIT       ;STORE DRIVE SELECT BIT
3178  004316  010037  001240                  MOV     R0,$TMP7        ;STORE DRIVE NUMBER TO BE TESTED
3179  004322  000713                          BR      18$
3180
3181  004324  013737  001240  001626  21$:    MOV     $TMP7,DRVNUM    ;LOAD LOWEST # DRIVE PRESENT INTO DRVNUM
3182
3183  004332  023727  001624  000002          CMP     SRTFLG,#2       ;TEST IF CLOCK ADJUST START
3184  004340  001002                          BNE     25$             ;NO - SKIP
3185  004342  000137  042332                  JMP     ADJCLK          ;GO TO ADJUST CLOCK ROUTINE
3186
3187  004346                          25$:
3188
3189                                  ;;*****************************************************************
3190                                  ;*TEST 4        RELEASE ALL DRIVES
3191                                  ;*
3192                                  ;*      RELEASE ALL DRIVES.  MAKE SURE NO ERROR
3193                                  ;*      SETS AND STATUS VALID IS RESET.
3194                                  ;*
3195                                  ;;*****************************************************************
3196  004346  000004          TST4:   SCOPE
3197  004350  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3198  004356  104416                          TSSINIT                 ;INITIALIZE SUBSYSTEM
3199  004360  104003                          ERROR   3               ;BAD INIT
3200
3201  004362  013737  001626  001610          MOV     DRVNUM,L.CS2    ;SET DRIVE NUMBER
3202  004370  012737  000101  001600          MOV     #SELDRV,L.CS1   ;SET DRIVE SELECT
3203
3204  004376  104417                          TLOADRK                 ;LOAD RK REGS
3205  004400  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3206  004402  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3207
3208  004404  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3209  004406  104004                          ERROR   4 ;OR5,6,7      ;REPORT ANY ERRORS
3210
3211  004410  012737  000010  001610          MOV     #RLS,L.CS2      ;SET DRIVE RELEASE,STILL SET FOR SELECT
3212
3213  004416  104417                          TLOADRK                 ;LOAD RK REGS
3214  004420  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3215  004422  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3216
3217  004424  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3218  004426  104004                          ERROR   4  ;OR 5, 6, OR 7 ;REPORT ALL ERRORS
3219  004430  032737  100000  001552          BIT     #SVAL,T.DS      ;DID SVAL RESET?
3220  004436  001404                          BEQ     1$              ;YES - SKIP
3221  004440  012737  054500  001400          MOV     #EM49,EM3N      ;MESSAGE (SVAL NOT RESET W/RELEASE)
3222  004446  104003                          ERROR   3
3223  004450                          1$:
```

```
                                TSTLUP:
                                ;;*****************************************************************
                                ;*TEST 5         NON-STANDARD MESSAGES AND SVAL
                                ;*
                                ;*      PICK ONE OF THE AVAILABLE DRIVES AND GET
                                ;*      NON-STANDARD MESSAGES.  MAKE SURE NO
                                ;*      ERROR OCCURS AND STATUS VALID DOES NOT SET
                                ;*      AND THAT NON-STANDARD MESSAGES CAUSE STATUS
                                ;*      VALID TO RESET.
                                ;*
                                ;;*****************************************************************
004450  000004                  TST5:   SCOPE
004452  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
004460  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
004462  104003                          ERROR   3               ;BAD CLEAR MESSAGE
004464  012701  000001                  MOV     #1,R1           ;PRESET R1 FOR MESSAGE PAIR 1
004470  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRV NUMBER
004476  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD SELECT COMMAND
004504  005037  001616          1$:     CLR     L.MR1           ;LOAD FOR STANDARD STATUS
004510  104417                          TLOADRK                 ;LOAD RK
004512  104423                          TWAT16                  ;WAIT FOR INTERRUPT
004514  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
004516  104421                          TCHKOP                  ;CHECK OPERATION
004520  104004                          ERROR   4 ;5,6 OR 7     ;REPORT ALL ERRORS

004522  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
004530  001007                          BNE     2$              ;YES-SKIP

004532  012737  055235  001460          MOV     #EMSVAL,EM11N
004540  012737  046676  057702          MOV     #EMSELD,DF011A
004546  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRIVE SELECT"

004550  010137  001616          2$:     MOV     R1,L.MR1        ;LOAD MESSAGE PAIR SELECT

004554  104417                          TLOADRK                 ;LOAD RK
004556  104423                          TWAT16                  ;WAIT FOR INTERRUPT
004560  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

004562  104421                          TCHKOP                  ;CHECK OPERATION
004564  104004                          ERROR   4 ;5,6, OR 7    ;REPORT ALL ERRORS

004566  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID RESET
004574  001407                          BEQ     3$              ;YES-SKIP

004576  012737  055235  001470          MOV     #EMSVAL,EM12N
004604  012737  055252  057702          MOV     #EMNZPR,DF011A
004612  104012                          ERROR   12              ;"SVAL NOT RESET RESULT OF SEL W/ NON-0 PAIR"

004614  022701  000003          3$:     CMP     #3,R1           ;WAS PAIR 3 SELECTED?
004620  001402                          BEQ     4$              ;YES-SKIP
004622  005201                          INC     R1              ;BUMP TO NEXT PAIR
004624  000727                          BR      1$              ;SKIP TO DO IT.
004626                          4$:
                                ;;*****************************************************************
                                ;*TEST 6         WRITING CS2 AND STATUS VALID
```

```
3280                                    ;*
3281                                    ;*      SELECT AN AVAILABLE DRIVE.  MAKE SURE STATUS
3282                                    ;*      VALID IS SET.  WRITE COMMAND AND STATUS REGISTER 2.
3283                                    ;*      MAKE SURE STATUS VALID RESETS.
3284                                    ;*
3285                                    ;;*****************************************************************
3286  004626  000004                    TST6:   SCOPE
3287  004630  012737  000062  001262            MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3288  004636  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
3289  004640  104003                            ERROR   3               ;BAD INIT ERROR
3290
3291  004642  013737  001626  001610            MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3292  004650  012737  000101  001600            MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3293
3294  004656  104417                            TLOADRK                 ;LOAD RK
3295  004660  104423                            TWAIT6                  ;WAIT FOR INTERRUPT
3296  004662  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3297
3298  004664  104421                            TCHKOP                  ;CHECK OPERATION
3299  004666  104004                            ERROR   4 ;5,6, OR 7    ;REPORT ALL ERRORS
3300
3301  004670  032737  100000  001552            BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
3302  004676  001007                            BNE     1$              ;YES-SKIP
3303
3304  004700  012737  055235  001460            MOV     #EMSVAL,EM11N
3305  004706  012737  046676  057702            MOV     #EMSELD,DF011A
3306  004714  104011                            ERROR   11              ;"SVAL NOT SET RESULT OF DRV SELECT"
3307
3308  004716  013762  001626  000010  1$:       MOV     DRVNUM,RKCS2(R2)  ;WRITE CS2 TO RESET SVAL
3309
3310  004724  104420                            TGETRK                  ;GET RK REGS.
3311
3312  004726  032737  100000  001552            BIT     #SVAL,T.DS      ;TEST SVAL RESET
3313  004734  001407                            BEQ     2$              ;YES-SKIP
3314
3315  004736  012737  055235  001470            MOV     #EMSVAL,EM12N
3316  004744  012737  055313  057702            MOV     #EMWCS2,DF011A
3317  004752  104012                            ERROR   12              ;"SVAL NOT RESET BY WRITING CS2"
3318  004754                            2$:
3319
```

```
                                .SBTTL  **CONTROLLER ERROR TESTS

                                ;;*********************************************************************
                                ;*TEST 7         DRIVE TYPE ERROR
                                ;*
                                ;*       CREATE A DRIVE TYPE ERROR.  MAKE SURE DRIVE
                                ;*       TYPE ERROR SETS AND STATUS VALID SETS.
                                ;*
                                ;;*********************************************************************
004754  000004                  TST7:   SCOPE
004756  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
004764  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
004766  104003                          ERROR   3               ;BAD INIT ERROR

004770  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
004776  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
005004  052737  002000  001600          BIS     #CDT,L.CS1      ;LOAD DRIVE TYPE

005012  104417                          TLOADRK                 ;LOAD RK
005014  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
005016  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

005020  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
005022  000040                          .WORD   000040          ;DRIVE TYPE ERROR
005024  000000                          .WORD   0
005026  000000                          .WORD   0
005030  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DIFFERENCES (NO ERRORS,
                                                                ;ADDITIONAL ERRORS, DIFFERENT ERRORS)
005032  032737  100000  001552          BIT     #SVAL, T.DS     ;TEST IF SVAL SET
005040  001007                          BNE     1$              ;YES-SKIP

005042  012737  055235  001460          MOV     #EMSVAL,EM11N
005050  012737  053751  057702          MOV     #EMDTPE,DF011A
005056  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV TYPE ERR"
005060                          1$:
                                ;;*********************************************************************
                                ;*TEST 10        STATUS VALID AND PARITY ERROR
                                ;*
                                ;*       ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY.
                                ;*       MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION,
                                ;*       DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT,
                                ;*       AND STATUS VALID SET, ISSUE A CONTROLLER
                                ;*       CLEAR.  MAKE SURE DRIVE INTERRUPT AND ATTENTION
                                ;*       ARE STILL SET.  SELECT DRIVE AGAIN WITH GOOD
                                ;*       PARITY.  MAKE SURE ATTENTION, DRIVE STATUS
                                ;*       CHANGE, DRPAR, CONTROLLLER ERROR, DRIVE INTERRUPT,
                                ;*       AND STATUS VALID ARE SET AND SPAR IS RESET.
                                ;*       ISSUE A CONTROLLER CLEAR.  GET NON-STANDARD MESSAGES
                                ;*       AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION
                                ;*       ARE SET.  CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT
                                ;*       FOR ALL AVAILIABLE DRIVES.
                                ;*
                                ;;*********************************************************************
005060  000004                  T10:    SCOPE
005062  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
005070  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
```

```
005072  104003                              ERROR   3                   ;BAD INIT ERROR

005074  013737  001626  001610              MOV     DRVNUM,L.CS2        ;LOAD DRIVE NUMBER
005102  012737  000101  001600              MOV     #SELDRV,L.CS1       ;LOAD DRIVE SELECT
005110  012737  000020  001616              MOV     #PAT,L.MR1          ;LOAD EVEN PARITY BIT

005116  104417                              TLOADRK                     ;LOAD RK REGS-SELECT W/EVEN PARITY
005120  104423                              TWAT16                      ;WAIT FOR INTERRUPT
005122  104002                              ERROR   2                   ;TO SLOW/NOT COMPLETE ERROR

005124  104422                              TCHKWE                      ;CHECK OPERATION FOR EXPECTED ERROR
005126  000011                              DRPARERR!SPARERR            ;DRIVE SELECTED DRIVE BUS PARITY ERROR
005130  000000                              .WORD   0                   ;CONTROLLER DETECTED DRIVE BUS PARITY ERROR
005132  000000                              .WORD   0
005134  104004                              ERROR   4 ; OR 5,6,7        ;REPORT ANY DIFFERENCES

005136  012700  000400                      MOV     #BIT8,R0            ;ROUTINE TO DETERMINE WHICH BIT

005142  013701  001626                      MOV     DRVNUM,R1           ;SHOULD BE SET IN ASOF TO INDICATE
005146  001403                              BEQ     3$                  ;DRIVE ATTENTION. R0 WILL HAVE THE
005150  006300                      2$:     ASL     R0                  ;BIT THAT SHOULD BE SET FOR THE DRIVE
005152  005301                              DEC     R1                  ;IN USE
005154  001375                              BNE     2$

005156  030037  001556              3$:     BIT     R0,T.ASOF           ;TEST ATTENTION SET
005162  001007                              BNE     4$                  ;YES-SKIP
005164  012737  055421  001460              MOV     #EMDA,EM11N
005172  012737  053666  057702              MOV     #EMDPAR,DF011A
005200  104011                              ERROR   11                  ;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
005202  032737  040000  001540      4$:     BIT     #DI,T.CS1           ;TEST DRIVE INTERRUPT SET
005210  001007                              BNE     5$                  ;YES-SKIP
005212  012737  055355  001460              MOV     #EMDI,EM11N
005220  012737  053666  057702              MOV     #EMDPAR,DF011A
005226  104011                              ERROR   11                  ;"DRV INT NOT SET RESULT OF DRV PARITY ERR"

005230  032737  040000  001552      5$:     BIT     #DSC,T.DS           ;TEST DRIVE STATUS CHANGE SET
005236  001007                              BNE     6$                  ;YES-SKIP
005240  012737  055375  001460              MOV     #EMDSC,EM11N
005246  012737  053666  057702              MOV     #EMDPAR,DF011A
005254  104011                              ERROR   11                  ;"DSC NOT SET RESULT OF DRV PARITY ERR"

005256  032737  100000  001552      6$:     BIT     #SVAL,T.DS          ;TEST STATUS VALID SET
005264  001007                              BNE     7$                  ;YES-SKIP
005266  012737  055235  001460              MOV     #EMSVAL,EM11N
005274  012737  053666  057702              MOV     #EMDPAR,DF011A

005302  104011                              ERROR   11                  ;"SVAL NOT SET RESULT OF DRV PAR ERR"

005304  005037  001616              7$:     CLR     L.MR1               ;CLEAR PAT IN MR1

005310  052737  100000  001600              BIS     #CCLR,L.CS1         ;CLEAR CONTROLLER
005316  104417                              TLOADRK                     ;LOAD RK REGS TO DO CLEAR


005320  104421                              TCHKOP                      ;CHECK NO ERRORS SET
005322  104004                              ERROR   4 ; OR 5,6,7        ;REPORT ALL ERRORS STILL SET
```

```
                                                BIT     RQ,T.ASOF       ;TEST ATTENTION STILL SET
005324  030037  001556                          BNE     8$              ;YES-SKIP
005330  001007                                  MOV     #EMDA,EM14N
005332  012737  055421  001510                  MOV     #EMCCLR,DFD11A
005340  012737  055441  057702                  ERROR   14              ;"DRV ATT RESET RESULT OF CONT CLR"
005346  104014
                                                BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT STILL SET
005350  032737  040000  001540  8$:             BNE     9$              ;YES-SKIP
005356  001007                                  MOV     #EMDI,EM14N
005360  012737  055355  001510                  MOV     #EMCCLR,DFD11A
005366  012737  055441  057702                  ERROR   14              ;"DRV INT RESET RESULT OF CONTROLLER CLR"
005374  104014
                                                MOV     #SELDRV,L.CS1   ;LOAD SELECT DRIVE
005376  012737  000101  001600  9$:
                                                TLOADRK                 ;LOAD RK REGS
005404  104417                                  TWAIT6                  ;WAIT FOR INTERRUPT
005406  104423                                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
005410  104002
                                                TCHKWE                  ;CHECK THAT ERRORS ARE SET
005412  104422                                  .WORD   000010          ;DPAR SET (NO SPAR 1, REPORT
005414  000010                                  .WORD   0               ;ANY DISCREPENCIES
005416  000000                                  .WORD   0
005420  000000                                  ERROR   4 ; OR 5,6,7
005422  104004                                  BIT     RQ,T.ASOF       ;TEST ATTENTION STILL SET.
005424  030037  001556                          BNE     10$             ;YES-SKIP
005430  001007                                  MOV     #EMDA,EM11N
005432  012737  055421  001460                  MOV     #EMDPAR,DFD11A
005440  012737  053666  057702                  ERROR   11              ;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
005446  104011
                                                BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT STILL SET
005450  032737  040000  001540  10$:            BNE     11$             ;YES-SKIP
005456  001007                                  MOV     #EMDI,EM11N
005460  012737  055355  001460                  MOV     #EMDPAR,DFD11A
005466  012737  053666  057702                  ERROR   11              ;"DRV INT NOT SET RESULT OF DRV  PARITY ERR"
005474  104011
                                                BIT     #DSC,T.DS       ;TEST STATUS CHANGE STILL SET
005476  032737  040000  001552  11$:            BNE     12$             ;YES-SKIP
005504  001007                                  MOV     #EMDSC,EM11N
005506  012737  055375  001460                  MOV     #EMDPAR,DFD11A
005514  012737  053666  057702                  ERROR   11              ;"DSC NOT SET RESULT OF DRV PARITY ERR"
005522  104011
                                                BIT     #SVAL,T.DS      ;TEST STATUS VALID STILL SET
005524  032737  100000  001552  12$:            BNE     13$             ;YES-SKIP
005532  001007                                  MOV     #EMSVAL,EM11N
005534  012737  055235  001460                  MOV     #EMDPAR,DFD11A
005542  012737  053666  057702                  ERROR   11              ;"SVAL NOT SET RESULT OF DRV PARITY ERR"
005550  104011
                                                BIS     #CCLR,L.CS1     ;CLEAR CONTROLLER
005552  052737  100000  001600  13$:
                                                TLOADRK                 ;LOAD RK REGS (DO CLEAR)
005560  104417
                                                MOV     #1,R1           ;SET COUNTER TO SELECT STATUS PAIR
005562  012701  000001                          MOV     #SELDRV,..CS1   ;LOAD DRIVE SELECT
005566  012737  000101  001600
```

```
003488  005574  010137  001616          14$:    MOV     R1,L.MR1        ;LOAD STATUS PAIR SELECTION
003489  005600  104417                          TLOADRK                 ;LOAD RK REGS
003490  005602  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
003491  005604  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
003492
003493  005606  104421                          TCHKOP                  ;CHECK IF ANY ERRORS SET
003494  005610  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS SET.
003495
003496  005612  030037  001556                  BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET
003497  005616  001007                          BNE     15$             ;YES-SKIP
003498  005620  012737  055421  001410          MOV     #EMDA,EM4N
003499  005626  012737  055252  057702          MOV     #EMNZPR,DF011A
003500  005634  104014                          ERROR   14              ;"ATTENTION RESET RESULT OF NON-0 PAIR SEL"
003501
003502  005636  032737  040000  001540  15$:    BIT     #DI,T.CS1
003503  005644  001007                          BNE     16$
003504  005646  012737  055355  001510          MOV     #EMDI,EM14N
003505  005654  012737  055252  057702          MOV     #EMNZPR,DF011A
003506  005662  104014                          ERROR   14              ;"DRV INT RESET RESULT OF NON-0 PAIR SELECT"
003507
003508  005664  005201                  16$:    INC     R1              ;BUMP PAIR SELECT
003509  005666  022701  000004                  CMP     #4,R1           ;ALL PAIRS DONE?
003510  005672  001340                          BNE     14$             ;NO-LOOP
003511
003512  005674  005037  001616                  CLR     L.MR1           ;CLEAR MR1
003513
003514  005700  012737  000105  001600          MOV     #CLEAR,L.CS1    ;LOAD DRIVE CLEAR
003515
003516  005706  104417                          TLOADRK                 ;DO DRIVE CLEAR
003517  005710  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
003518  005712  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
003519
003520  005714  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
003521  005716  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
003522
003523  005720  012701  000020                  MOV     #20,R1          ;SET COUNT FOR SHORT WAIT
003524  005724  005301                  17$:    DEC     R1              ;TO ALLOW CONTROLLER TIME TO POLL
003525  005726  001376                          BNE     17$             ;DRIVES
003526
003527  005730  104420                          TGETRK                  ;GET RK REGS
003528  005732  030037  001556                  BIT     R0,T.ASOF       ;TEST ATTENTION RESET
003529  005736  001407                          BEQ     18$             ;YES-SKIP
003530  005740  012737  055421  001470          MOV     #EMDA,EM12N
003531  005746  012737  046724  057702          MOV     #EMDCLR,DF011A
003532  005754  104012                          ERROR   12              ;"ATTENTION NOT RESET RESULT OF DRV CLEAR
003533
003534  005756  032737  040000  001540  18$:    BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT RESET
003535  005764  001407                          BEQ     19$             ;YES-SKIP
003536  005766  012737  055355  001470          MOV     #EMDI,EM12N
003537  005774  012737  046724  057702          MOV     #EMDCLR,DF011A
003538  006002  104012                          ERROR   12              ;"DRV INT NOT RESET RESULT OF DRIVE CLR"
003539
003540  006004                          19$:
003541                                  ;;*****************************************************************
003542                                  ;*TEST 11        UNIT FIELD ERROR ON RELEASE
003543                                  ;*
```

```
                                        ;*      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.
                                        ;*
                                        ;;*****************************************************************
006004  000004                        TST11:  SCOPE
006006  012737  000062  001262                MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
006014  104416                              TSSINIT                 ;CLEAR SUBSYSTEM
006016  104002                              ERROR   2               ;BAD INIT ERROR

006020  013737  001626  001610              MOV     DRVNUM,L.CS2    ;SELECT A DRIVE
006026  012737  000101  001600              MOV     #SELDRV,L.CS1   ;DO DRIVE SELECT

006034  104417                              TLOADRK                 ;LOAD RK
006036  104423                              TWAT16                  ;WAIT FOR INTERRUPT
006040  104002                              ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

006042  104421                              TCHKOP                  ;CHECK FOR ANY ERRORS
006044  104004                              ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS.

006046  052737  000010  001610              BIS     #RLS,L.CS2      ;LOAD RELEASE
006054  012737  000040  001616              MOV     #DMD,L.MR1      ;SET DIAGNOSTIC MODE

006062  104417                              TLOADRK                 ;LOAD RK

006064  004437  035050                      JSR     R4,MCLOCK       ;CALL MAINT CLOCK
006070     023                              .BYTE   ^D19            ;NUMBER OF PHASES
006071     002                              .BYTE   2               ;NUMBER OF CLOCK XISTIONS

006072  042762  000040  000026              BIC     #DMD,RKMR1(R2)  ;CLEAR DIAG MODE

006100  104423                              TWAT16                  ;WAIT FOR INTERRUPT
006102  104002                              ERROR   2               ;TO SLOW/NOT COMPLETED

006104  104422                              TCHKWE                  ;CHECK OPERATION WITH ERROR
006106  000000                              .WORD   0
006110  000000                              .WORD   0
006112  000004                              .WORD   UFERR           ;UNIT FIELD ERROR
006114  104004                              ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES

006116  104416                              TSSINIT                 ;CLEAR SUBSYSTEM TO INSURE UFE RESETS
006120  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.
                                        ;*
                                        ;;*****************************************************************
006122  000004                        TST12:  SCOPE
```

```
3600  006124  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3601  006132  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3602  006134  104003                          ERROR   3               ;BAD INIT ERROR
3603
3604  006136  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3605  006144  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3606
3607  006152  104417                          TLOADRK                 ;LOAD RK
3608  006154  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3609  006156  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3610
3611  006160  104421                          TCHKOP                  ;CHECK FOR ANY ERROR
3612  006162  104004                          ERROR   4 ; OR 5.6,7    ;REPORT ALL ERRORS
3613
3614  006164  012737  000043  001616          MOV     #DMD!BIT1!BIT0,L.MR1    ;LOAD DIAG MODE & MSG PAIR 3
3615  006172  005037  001610                  CLR     L.CS2           ;LOAD FOR DRIVE 0
3616
3617  006176  104417                          TLOADRK                 ;LOAD RK
3618
3619  006200  004437  035050                  JSR     R4,MCLOCK       ;CALL MAINTENANCE CLOCK
3620  006204     026                          .BYTE   ^D22            ;THROUGH PHASE 6
3621  006205     002                          .BYTE   2               ;PLUS 2 TRANSITIONS
3622
3623  006206  042762  000040  000026          BIC     #DMD,RKMR1(R2)  ;CLEAR DIAG MODE
3624
3625  006214  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3626  006216  104002                          ERROR   2               ;TO SLOW/NOT COMPLETED ERROR
3627
3628  006220  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
3629  006222  000000                          .WORD   0
3630  006224  000000                          .WORD   0
3631  006226  000004                          .WORD   UFERR           ;UNIT FIELD ERROR SHOULD SET
3632  006230  104004                          ERROR   4 ; OR 5.6,7    ;REPORT ANY DISCREPENCIES
3633
3634                                  .SBTTL  **ATTENTION HANDLING BY CONTROLLER
3635
3636                                  ;;*****************************************************************
3637                                  ;*TEST 13       DOUBLE INTERRUPT
3638                                  ;*
3639                                  ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.
3640                                  ;*      MAKE SURE STATUS VALID IS SET.  CHECK THAT SECOND
3641                                  ;*      INTERRUPT OCCURS.  AFTER SECOND INTERRUPT
3642                                  ;*      CHECK THAT STATUS VALID IS RESET.  ISSUE SELECT
3643                                  ;*      AND MAKE SURE STATUS VALID SETS.  CLEAR DRIVE.
3644                                  ;*      CHECK THAT DRIVE STATUS CHANGE SETS
3645                                  ;*      (BIT 14 OF DRIVE STATUS
3646                                  ;*      REGISTER)
3647                                  ;*
3648                                  ;;*****************************************************************
3649  006232  000004                  TST13:  SCOPE
3650  006234  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3651  006242  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3652  006244  104003                          ERROR   3               ;BAD INIT ERROR
3653
3654  006246  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3655  006254  012737  000113  001600          MOV     #RECAL,L.CS1    ;LOAD RECAL
```

```
3656
3657  006262  104417                          TLOADRK                 ;LOAD RK
3658  006264  104423                          TWAT16                  ;WAIT FOR 1ST INTERRUPT
3659  006266  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3660  006270  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
3661  006274  104420                          TGETRK                  ;GET RK REGS
3662  006276  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAL SET
3663  006304  001010                          BNE     1$              ;YES-SKIP
3664  006306  012737  055235  001460          MOV     #EMSVAL,EM11N
3665  006314  012737  046765  057702          MOV     #EMRCAL,DF011A
3666  006322  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF RECAL"
3667  006324  000463                          BR      50$             ;ABORT TEST
3668
3669  006326  104437                  1$:     TWAT8S                  ;WAIT FOR INTERRUPT
3670  006330  000401                          BR      2$              ;NO INTERRUPT RETURN
3671  006332  000404                          BR      3$              ;INTERRUPT RETURN
3672
3673  006334  012737  054550  001370  2$:     MOV     #EM50,EM2N      ;ALTER MESSAGE "NO 2ND INTERRUPT OR IT WAS LATE"
3674  006342  104002                          ERROR   2
3675
3676  006344  104420                  3$:     TGETRK                  ;GET RK REGS
3677  006346  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAC SET NOW
3678  006354  001410                          BEQ     4$              ;NO-SKIP
3679  006356  012737  055235  001470          MOV     #EMSVAL,EM12N
3680  006364  012737  055462  057702          MOV     #EM2INT,DF011A
3681  006372  104012                          ERROR   12              ;"SVAL NOT RESET RESULT OF SECOND TEST"
3682  006374  000437                          BR      50$
3683
3684  006376  032737  040000  001552  4$:     BIT     #DSC,T.DS       ;TEST DSC SET BY ATTENTION
3685  006404  001010                          BNE     5$              ;YES-SKIP
3686  006406  012737  055375  001460          MOV     #EMDSC,EM11N
3687  006414  012737  055462  057702          MOV     #EM2INT,DF011A
3688  006422  104011                          ERROR   11              ;"DSC NOT SET RESULT OF SECOND INTERRUPT"
3689  006424  000423                          BR      50$
3690
3691  006426  012737  000101  001600  5$:     MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3692
3693  006434  104417                          TLOADRK                 ;LOAD RK REGS
3694  006436  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3695  006440  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3696
3697  006442  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3698
3699  006444  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
3700
3701  006446  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAC SET
3702  006454  001007                          BNE     50$             ;YES-SKIP
3703  006456  012737  055235  001460          MOV     #EMSVAL,EM11N
3704  006464  012737  046676  057702          MOV     #EMSELD,DF011A
3705  006472  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV SEL.
3706  006474                          50$:
3707                                  ;;*****************************************************************
3708                                  ;*TEST 14       SINGLE INTERRUPT FROM ATTENTION
3709                                  ;*
3710                                  ;*      DO A SEEK TO CYLINDER 0.  WAIT FOR INTERRUPT FROM
3711                                  ;*      DRIVE ATTENTION.  LOWER PRIORITY AGAIN AND MAKE
```

```
                                        ;*      SURE ANOTHER INTERRUPT DOES NOT OCCUR.  CLEAR DRIVE.
                                        ;*
                                        ;;*****************************************************************
006474  000004                  TST14:  SCOPE
006476  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
006504  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
006506  104003                          ERROR   3               ;BAD INIT ERROR

006510  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
006516  012737  000117  001600          MOV     #SEEK,L.CS1     ;LOAD SEEK DCYL LEFT AT 0.

006524  104417                          TLOADRK                 ;LOAD RK REGS
006526  104423                          TWAT16                  ;WAIT FOR INTERRUPT
006530  104002                          ERROR   2               ;TO SLOW/NOT COMPLETED ERROR

006532  104420                          TGETRK                  ;GET RK REGS

006534  032737  040000  001540          BIT     #DI,T.CS1       ;TEST DI SET
006542  001010                          BNE     2$              ;YES-SKIP
006544  012737  055355  001460          MOV     #EMDI,EM11N
006552  012737  055503  057702          MOV     #EMSKSF,DF011A
006560  104011                          ERROR   11              ;"DI NOT SET RESULT OF SEEK TO SELF"
006562  000417                          BR      50$

006564  012700  000031          2$:     MOV     #25.,R0         ;LOAD AND DECREMENT A COUNT TO
006570  005300                  3$:     DEC     R0              ;ZERO. GIVE CONTROLLER A CHANCE TO
006572  001376                          BNE     3$              ;INTERRJPT AGAIN.  ERROR IF IT DOES.

006574  022737  000001  001662          CMP     #1,INTSET       ;CHECK ONLY ONE INTERRUPT OCCURRED
006602  001407                          BEQ     50$             ;YES-SKIP
006604  012737  055655  001500          MOV     #EMMI,EM13N
006612  012737  055503  057702          MOV     #EMSKSF,DF011A
006620  104013                          ERROR   13              ;"MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"

006622  104421                  50$:    TCHKOP                  ;CHECK FOR ANY ERRORS
006624  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL ERRORS
                                        ;;*****************************************************************
                                        ;*TEST 15       RESET ATTENTIONS WITH UNIBUS INIT
                                        ;*
                                        ;*      DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
                                        ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
                                        ;*
                                        ;;*****************************************************************
006626  000004                  TST15:  SCOPE
006630  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
006636  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
006640  104003                          ERROR   3               ;BAD INIT ERROR

006642  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
006650  012737  000117  001600          MOV     #SEEK,L.CS1     ;LOAD SEEK (TO SELF-0)

006656  104417                          TLOADRK                 ;LOAD RK REGS
006660  104423                          TWAT16                  ;WAIT FOR INTERRUPT
006662  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
```

```
006664  104420                                  TGETRK                          ;GET RK REGS

006666  032737  040000  001540                  BIT     #DI,T.CS1               ;TEST DI SET
006674  001010                                  BNE     1$                      ;YES-EXIT
006676  012737  055355  001460                  MOV     #EMDI,EM11N
006704  012737  055503  057702                  MOV     #EMSKSF,DF011A
006712  104011                                  ERROR   11                      ;"DI NOT SET RESULT OF SEEK TO SELF
006714  000450                                  BR      50$

006716  005037  001662                  1$:     CLR     INTSET                  ;CLEAR INTERRUPT COUNTER
006722  000005                                  RESET                           ;DO UNIBUS RESET
006724  004737  043664                          JSR     PC,$TKINT               ;RESET KEYBOARD INTERRUPT

006730  005037  001660                          CLR     LCLKTK                  ;CLEAR TICK COUNTER
006734  004737  034214                  5$:     JSR     PC,MYTIME               ;CALL TIMER
006740  022737  000012  001660                  CMP     #10.,LCLKTK             ;COUNT 10 TICKS (MILLISECONDS)?
006746  001372                                  BNE     5$                      ;NO - LOOP

006750  012762  000100  000000                  MOV     #IE,RKCS1(R2)           ;SET IE FOR ANY STRAY INTERRUPTS
006756  004737  033704                          JSR     PC,OPTTST               ;SET UP OPTIONS AGAIN

006762  104423                                  TWAT16                          ;WAIT 16 MS FOR AN INTERRUPT
006764  000410                                  BR      2$                      ;NONE IS EXPECTED SO RETURN SHOULD BE
                                                                                ;HERE-BR TO CONTINUE TEST.
006766  012737  055520  001500                  MOV     #EMUXIT,EM13N           ;INT OCCURRED ON RESET
006774  012737  055562  057702                  MOV     #EMRSET,DF011A
007002  104013                                  ERROR   13                      ;"UNEXECUTED INTERRUPT RESULT OF RESET"
007004  000414                                  BR      50$
007006  104420                          2$:     TGETRK                          ;GET RK REGS
007010  032737  040000  001540                  BIT     #DI,T.CS1               ;TEST DI RESET
007016  001407                                  BEQ     50$                     ;YES-SKIP
007020  012737  055355  001470                  MOV     #EMDI,EM12N
007026  012737  055562  057702                  MOV     #EMRSET,DF011A
007034  104012                                  ERROR   12                      ;"DI NOT RESET RESULT OF RESET"

007036                                  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.
                                        ;*
                                        ;;*************************************************************
007036  000004                          TST16:  SCOPE
007040  012737  000062  001262                  MOV     #50.,$TIMES             ;;DO 50. ITERATIONS
007046  012701  000003                          MOV     #3,R1                   ;PRESET FOR SELECTING TRACK 3

007052  104416                                  TSSINIT                         ;CLEAR SUBSYSTEM
007054  104003                                  ERROR   3
```

```
3824
3825  007056  012737  000113  001600          MOV     #RECAL,L.CS1    ;SET UP TO RECAL
3826  007064  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE
3827
3828  007072  104417                          TLOADRK                 ;LOAD RK REGS
3829
3830  007074  104423                          TWAT16                  ;WAIT FOR 1ST INTERRUPT
3831  007076  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3832
3833  007100  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
3834
3835  007104  104437                          TWAT8S                  ;WAIT FOR INTERRUPT
3836  007106  104002                          ERROR   2
3837
3838  007110  012737  007116  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP ON ERROR
3839
3840  007116  104416                  1$:     TSSINIT                 ;CLEAR SUBSYSTEM
3841  007120  104003                          ERROR   3               ;BAD INIT ERROR
3842
3843  007122  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3844  007130  012737  000117  001600          MOV     #SEEK,L.CS1     ;LOAD SEEK
3845  007136  110137  001607                  MOVB    R1,L.DT         ;LOAD TRACK
3846
3847  007142  104417                          TLOADRK                 ;LOAD RK REGS
3848  007144  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3849  007146  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3850
3851  007150  032701  000001                  BIT     #BIT0,R1        ;TEST IF HEAD ADDRESS HAS BIT 0
3852  007154  001403                          BEQ     2$              ;NO - SKIP
3853  007156  032701  000002                  BIT     #BIT1,R1        ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
3854  007162  001007                          BNE     3$              ;YES-GO CHECK BOTH IDAE AND SKI SET
3855
3856  007164  104422                  2$:     TCHKWE                  ;CHECK OPERATION WITH ERROR
3857  007166  002000                          IDAERR                  ;ILLEGAL DISK ADDRESS ERROR
3858  007170  000000                          0
3859  007172  000000                          0
3860  007174  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL DISCREPANCIES
3861  007176  104415                          SCOP1                   ;LOCAL LOOP ON ERROR
3862  007200  000406                          BR      4$
3863
3864  007202  104422                  3$:     TCHKWE                  ;CHECK OPERATION WITH ERROR
3865  007204  002002                          IDAERR!SKIERR           ;ILLEGAL DISK ADDRESS ERROR
3866  007206  000000                          0                       ;SEEK INCOMPLETE
3867  007210  000000                          0
3868  007212  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ANY DISCREPANCIES
3869  007214  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
3870
3871  007216  005201                  4$:     INC     R1              ;ELSE BUMP TO NEXT ILLEGAL TRACK
3872  007220  022701  000010                  CMP     #8.,R1          ;ALL ILLEGAL TRACKS SELECTED?
3873  007224  001334                          BNE     1$              ;NO-LOOP
3874                          ;;*****************************************************************
3875                          ;*TEST 17        ILLEGAL DISK ADDRESS (PART 2)
3876                          ;*
3877                          ;*      ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE
3878                          ;*      ILLEGAL DISK ADDRESS ERROR SETS.  CLEAR CONTROLLER AND DRIVE
3879                          ;*
```

```
                                ;;*****************************************************************
007226  000004                 TST17:  SCOPE
007230  012737  000062  001262         MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
007236  104416                         TSSINIT                 ;CLEAR SUBSYSTEM
007240  104003                         ERROR   3               ;BAD INIT ERROR

007242  012737  000113  001600         MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
007250  013737  001626  001610         MOV     DRVNUM,L.CS2    ;LOAD DRIVE

007256  104417                         TLOADRK                 ;LOAD RK REGS

007260  104423                         TWAT16                  ;WAIT FOR 1ST INTERRUPT
007262  104002                         ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

007264  005037  001662                 CLR     INTSET          ;CLEAR INTERRUPT FLAG

007270  104437                         TWAT8S                  ;WAIT FOR INTERRUPT
007272  104002                         ERROR   2

007274  012737  007302  001110         MOV     #1$,$LPERR      ;SET LOOP TO BYPASS RECAL

007302  104416                 1$:     TSSINIT                 ;CLEAR SUBSYSTEM
007304  104003                         ERROR   3

007306  013737  001626  001610         MOV     DRVNUM, L.CS2   ;LOAD DRIVE NUMBER
007314  012737  000117  001600         MOV     #SEEK,L.CS1     ;LOAD SEEK
007322  012737  001000  001614         MOV     #1000, L.DCYL   ;LOAD ILLEGAL CYLINDER

007330  104417                         TLOADRK                 ;LOAD RK REGS
007332  104423                         TWAT16                  ;WAIT FOR INTERRUPT
007334  104002                         ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

007336  104422                         TCHKWE                  ;CHECK OPERATION WITH ERROR
007340  002000                         .WORD   IDAERR          ;DISK ADDRESS ERROR
007342  000000                         .WORD   0
007344  000000                         .WORD   0
007346  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 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.
                                        ;*
                                        ;;*****************************************************************
007350  000004                  TST20:  SCOPE
007352  012737  000001  001262          MOV     #1,$TIMES       ;;DO 1 ITERATION
007360  105037  007432                  CLRB    2$              ;CLEAR SECTOR POINTER
007364  005737  001304                  TST     $PASS           ;TEST IF FIRST PASS
007370  001402                          BEQ     15$             ;NO - SKIP
007372  000137  010136                  JMP     28$             ;ELSE EXITT TEST
007376                          15$:
007376  005000                          CLR     R0
007400  005005                          CLR     R5              ;CLEAR R5 FOR BAD SECTOR COUNTING
007402  013703  001640                  MOV     BSF26P,R3       ;SET POINT IN TO STORE BS 26 SECT FORMAT
007406  012737  007414  001110          MOV     #1$,$LPERR      ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
007414  104416                  1$:     TSSINIT                 ;CLEAR SUBSYSTEM
007416  104003                          ERROR   3               ;BAD INIT ERROR
                                
007420  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS WITH
007424  000121                          RDDATA                  ; READ DATA
007426  177400                          -400                    ; WORD COUNT
007430  060604                          IBUFF                   ; BUFFER ADDRESS
007432     000                  2$:     .BYTE   0               ; SECTOR ADDRESS
007433     002                          .BYTE   2               ; TRACK ADDRESS
007434  000632                          632                     ; CYLINDER ADDRESS
                                
007436  104417                          TLOADRK                 ;LOAD "L" REGS INTO RK
007440  104431                          TWAT112                 ;WAIT FOR INTERRUPT
007442  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
007444  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
007446  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
007450  104415                          SCOP1                   ;LOOP TO 1$ IF SW 9 SET
                                
007452  105737  001103                  TSTB    $ERFLG          ;TEST FOR ERROR IN OPERATION
007456  001477                          BEQ     7$              ;NO-SKIP
007460  005700                          TST     R0              ;GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
007462  001020                          BNE     3$              ;NO-SKIP
007464  062737  000002  007432          ADD     #2,2$           ;NEXT SECTOR ADDRESS
007472  122737  000012  007432          CMPB    #10.,2$         ;PAST APPLICABLE SECTORS?
007500  001063                          BNE     6$              ;NO-SKIP
007502  012737  054647  001360          MOV     #EM51,EM1N
007510  104001                          ERROR   1               ;"CANNOT READ BS FILES
007512  043737  001630  001354  25$:    BIC     DRVBIT,$DEVM    ;CLEAR DRIVE FROM DRIVE MAP
007520  000137  027236                  JMP     NEWDRV          ;ABORT TEST PASS.
                                
007524  022700  000001          3$:     CMP     #1,R0           ;GETTING A BS SOFT SECTOR 26 SECT FORMAT?
007530  001014                          BNE     4$              ;NO-SKIP
007532  062737  000002  007432          ADD     #2,2$           ;NEXT SECTOR ADDRESS
007540  122737  000026  007432          CMPB    #22.,2$         ;PAST APPLICABLE SECTORS?
007546  001040                          BNE     6$              ;NO-SKIP
007550  012737  054647  001360          MOV     #EM51,EM1N
007556  104001                          ERROR   1               ;"CANNOT READ BS FILES"
```

```
007560  000754                                  BR      25$
007562  022700  000002                  4$:     CMP     #2,R0           ;GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
007566  001014                                  BNE     5$              ;NO-SKIP
007570  062737  000002  007432                  ADD     #2,2$           ;NEXT SECTOR ADDRESS
007576  122737  000013  007432                  CMPB    #11.,2$         ;PAST APPLICABLE SECTORS?
007604  001021                                  BNE     6$              ;NO-SKIP
007606  012737  054647  001360                  MOV     #EM51,EM1N
007614  104001                                  ERROR   1               ;"CANNOT READ BS FILES"
007616  000735                                  BR      25$
007620  062737  000002  007432          5$:     ADD     #2,2$           ;NEXT SECTOR (BS SOFT 24 SECT MODE)
007626  122737  000027  007432                  CMPB    #23.,2$         ;PAST APPLICABLE SECTORS?
007634  001005                                  BNE     6$              ;NO-SKIP
007636  012737  054647  001360                  MOV     #EM51,EM1N
007644  104001                                  ERROR   1               ;"CANNOT READ BS FILES
007646  000721                                  BR      25$
007650  105037  001103                  6$:     CLRB    $ERFLG          ;CLEAR ERROR FLAG
007654  000657                                  BR      1$              ;DO NEXT READ
007656  005737  060610                  7$:     TST     IBUFF+4         ;CHECK FOR ALIGNMENT PACK
007662  001405                                  BEQ     8$              ;NO-SKIP
007664  012737  054742  001360                  MOV     #EM52,EM1N
007672  104001                                  ERROR   1               ;"ALIGNMENT PACK. DRIVE ABORTING"
007674  000706                                  BR      25$
007676  012701  060614                  8$:     MOV     #IBUFF+10,R1    ;SET TO START OF BAD SECTOR DATA
007702  022711  177777                  9$:     CMP     #-1,(R1)        ;TEST IF WORD ALL ONES (END OF DATA)
007706  001417                                  BEQ     11$             ;YES-SKIP
007710  012123                                  MOV     (R1)+,(R3)+     ;STORE CYLINDER
007712  012123                                  MOV     (R1)+,(R3)+     ;       TRACK AND SECTOR
007714  005205                                  INC     R5              ;BUMP ERROR COUNTER
007716  022705  000025                          CMP     #21.,R5         ;DOES IT TOTAL 20 FOR THIS FORMAT?
007722  001367                                  BNE     9$              ;NO-TEST AND MORE NEXT ADDRESS
007724  012737  055020  001360                  MOV     #EM53,EM1N
007732  104001                                  ERROR   1               ;TO MANY BAD SECTORS
007734  043737  001630  001354          10$:    BIC     DRVBIT,$DEVM    ;CLEAR DRIVE FROM TESTING
007742  000137  027236                          JMP     NEWDRV          ;ABORT PASS
007746  005200                          11$:    INC     R0              ;BUMP TO NEXT
007750  022700  000001                          CMP     #1,R0           ;NOW TESTING BS SOFT 26 SECTOR MODE?
007754  001011                                  BNE     12$             ;NO-SKIP
007756  012723  177777                          MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
007762  010337  001644                          MOV     R3,BSS26P       ;SET POINTER TO BAD SECTOR SOFTWARE FIELD
007766  112737  000012  007432                  MOVB    #12,2$          ;SET TO FIRST SECTOR THIS MODE
007774  000137  007414                          JMP     1$              ;GO READ IT.
010000  022700  000002                  12$:    CMP     #2,R0           ;NOW TESTING BS FACT 24 SECTOR MODE?
010004  001014                                  BNE     13$             ;NO-SKIP
010006  012723  177777                          MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
010012  112737  000001  007432                  MOVB    #1,2$           ;SET TO FIRST SECTOR THIS MODE
010020  010537  001646                          MOV     R5,BS26CT       ;STORE TOTAL BS COUNT 26 SECTOR MODE
010024  005005                                  CLR     R5              ;CLEAR COUNTER FOR COUNTING 24 SECT BS
010026  013703  001636                          MOV     BSF24P,R3       ;SET POINTER FOR STORING BS
```

```
4048  010032  000137  007414          JMP     1$              ;GO READ
4049
4050  010036  022700  000003  13$:    CMP     #3,R0           ;NOW TESTING BS SOFT 24 SECTOR MODE
4051  010042  001011                  BNE     14$             ;NO-SKIP
4052  010044  012723  177777          MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
4053  010050  010337  001642          MOV     R3,BSS24P       ;STORE POINTER TO BSS 24 SECTOR MODE
4054  010054  112737  000013  007432  MOVB    #13,2$          ;GET START OF FIELDS BSS 24 SECT MODE
4055  010062  000137  007414          JMP     1$              ;GO READ IT
4056
4057  010066  012723  177777  14$:    MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
4058  010072  010537  001650          MOV     R5,BS24CT       ;STORE COUNT BSS 24 SECTOR MODE
4059
4060
4061  010076  012700  057734          MOV     #BS26,R0        ;GET START OF BAD SECTOR BUFFER
4062  010102  012703  060144          MOV     #BS24+52.,R3    ;GET END OF BUFFER
4063
4064  010106  022720  000312  27$:    CMP     #312,(R0)+      ;TEST IF ANY SECTORS BAD IN CYL 312
4065  010112  001403                  BEQ     26$             ;YES - GET OUT
4066
4067  010114  020003                  CMP     R0,R3           ;CHECK IF ALL OF BUFFER TESTED
4068  010116  001373                  BNE     27$             ;NO - LOOP
4069  010120  000406                  BR      28$             ;EXIT
4070
4071  010122  012737  055701  001360  26$:    MOV     #DRVABT,EMIN    ;"BAD SECTOR IN AREA FOR TEST"
4072  010130  104001                  ERROR   1
4073  010132  000137  007512          JMP     25$
4074
4075  010136                  28$:
4076
4077                                  ;;*****************************************************************
4078                                  ;*TEST 21       FORMAT IN 26 SECTOR FORMAT
4079                                  ;*
4080                                  ;*      FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
4081                                  ;*      FORMAT.  VERIFY FORMAT AND THAT DATA LATE DID NOT
4082                                  ;*      OCCUR WITH WRITE HEADER OR IN READING DATA
4083                                  ;*      BUFFER AFTER READ HEADER.
4084                                  ;*
4085                                  ;;*****************************************************************
4086  010136  000004          TST21:  SCOPE
4087  010140  012737  000062  001262  MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
4088  010146  012737  000312  001664  MOV     #312,REFMT      ;SET REFORMAT SWITCH
4089  010154  105037  010277          CLRB    10$             ;CLEAR SECTOR POINTER
4090  010160  105037  010203          CLRB    11$             ;CLEAR TRACK POINTER
4091  010164  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
4092  010166  104003                  ERROR   3               ;BAD INIT ERROR
4093
4094  010170  004437  034574  9$:     JSR     R4,LRLOAD       ;LOAD "L" REGS
4095  010174  000127                  WRHEAD                  ;WRITE HEADER
4096  010176  177676                  -102                    ;WORD CNT FOR 26 SECTOR MODE
4097  010200  062604                  OBUFF                   ;BUFFER
4098  010202     000                  .BYTE   0               ;SECTOR 0
4099  010203     000          11$:    .BYTE   0               ;TRACK 0
4100  010204  000312                  312                     ;CYL 0
4101
4102  010206  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA
4103  010212  000600                  000600                  ;BUILD HEADERS-NO BAD SECTORS
```

```
010214  012737  010224  001110          MOV     #111$,$LPERR    ;SET LOCAL LOOP ON ERROR
010222  000402                          BR      1$              ;SKIP INIT
010224                          111$:
010224  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
010226  104003                          ERROR   3               ;BAD INIT ERROR
010230  104417                  1$:     TLOADRK                 ;LOAD RK REGS
010232  104421                          TWAIT12                 ;WAIT FOR INTERRUPT
010234  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

010236  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
010240  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

010242  104415                          SCOP1                   ;INTERNAL LOOP TO 111$

010244  012737  010304  001110          MOV     #112$,$LPERR    ;SET LOCAL LOOP ON ERROR
010252  010203                          MOV     R2,R3           ;BUILD POINTER TO RKDB
010254  062703  000024                  ADD     #RKDB,R3
010260  012701  060604                  MOV     #IBUFF,R1       ;SET POINTER TO BUFFER
010264  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
010270  000126                          RDREAD                  ;READ HEADER
010272  000000                          0                       ;NO WORDS COUNT
010274  003000                          0                       ;NO BUFFER
010276      000                         .BYTE   0               ;SECTOR 0
010277      000                 10$:    .BYTE   0               ;  TRACK 0
010300  000312                          312                     ;  CYL 312

010302  000402                          BR      2$              ;SKIP INIT
010304                          112$:
010304  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
010306  104003                          ERROR   3               ;BAD INIT ERROR
010310  104417                  2$:     TLOADRK                 ;LOAD RK REGS
010312  104423                          TWAIT16                 ;WAIT FOR INTERRUPT
010314  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

010316  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
010320  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL ERRORS
010322  012700  000003                  MOV     #3,R0           ;SET COUNT
010326  011321                  5$:     MOV     (R3),(R1)+      ;GET RKDB
010330  104420                          TGETRK                  ;GET RK REGS
010332  032737  100000  001550          BIT     #DLT,T.CS2      ;TEST IF DATA LATE
010340  001410                          BEQ     3$              ;NO-SKIP
010342  012737  053442  053071          MOV     #EMDLT,EM13
010350  012737  055570  057702          MOV     #EMRDB,DF011A
010356  104013                          ERROR   13              ;"DATA LATE SET RESULT OF DB READ
010360  104415                          SCOP1                   ;LOCAL LOOP TO 112$

010362  032737  100000  001540  3$:     BIT     #CERR,T.CS1     ;TEST IF CONT ERROR SET
010370  001410                          BEQ     4$              ;NO-SKIP
010372  012737  055614  001500          MOV     #EMCERR,EM13N
010400  012737  055570  057702          MOV     #EMRDB,DF011A
010406  104013                          ERROR   13              ;"CERR SET RESULT OF READ DB
010410  104415                          SCOP1                   ;LOCAL LOOP TO 112$
010412  005300                  4$:     DEC     R0              ;DEC COUNT
010414  001344                          BNE     5$              ;LOOP IF NOT ZERO
010416  012737  010426  001110          MOV     #117$,$LPERR    ;SET LOCAL LOOP 117$
010424  000402                          BR      7$              ;SKIP INIT
```

```
4160  010426                          117$:
4161  010426  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4162  010430  104003                          ERROR   3               ;BAD INIT ERROR
4163  010432  004437  035114          7$:     JSR     R4,RDSTHD       ;GO READ & SEQUENCE HEADERS
4164  010436  104421                          TCHKOP                  ;CONTROLLER ERROR RETURN
4165  010440  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4166  010442  104013                          ERROR   13              ;"DATA LATE SET RESULT OF DATA BUFFER READ"
4167  010444  104002                          ERROR   2               ;"OPERATION TO SLOW" MESSAGE
4168                                                                  ;OR "HEADER 0 NOT FOUND" MESSAGE
4169
4170  010446  004437  040776                  JSR     R4,GENCOM
4171  010452  100200                          100200                  ;COMPARE IBUF & OBUFF (HEADERS)
4172  010454  000414                          BR      6$              ;GOOD RETURN-NO MISCOMPARES
4173  010456  104015                          ERROR   15              ;REPORT 1ST MSCOMPARES
4174
4175  010460  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
4176  010464  005300                  12$:    DEC     R0              ;DECREMENT IT
4177  010466  001407                          BEQ     6$              ;EXIT IF ZERO
4178  010470  004437  040776                  JSR     R4,GENCOM
4179
4180  010474  040000                          040000                  ;RESUME COMPARE
4181  010476  000403                          BR      6$              ;GOOD RETURN-NO MORE ERRORS
4182  010500  104016                          ERROR   16              ;REPORT NEXT ERROR LINE
4183  010502  000770                          BR      12$             ;LOOP
4184  010504  104415                          SCOPI                   ;LOCAL ERROR LOOP TO 117$
4185
4186  010506  105737  001607          6$:     TSTB    L.DT            ;WAS TRACK 1 JUST DONE?
4187  010512  001010                          BNE     9$              ;YES-SKIP
4188
4189  010514  112737  000001  010203          MOVB    #1,11$          ;CHANGE PARAM TO LOAD "L" WITH
4190  010522  112737  000001  010277          MOVB    #1,10$          ;TRACK 2
4191  010530  000137  010170                  JMP     9$              ;JUMP TO DO ENTIRE TEST ON TRK 1
4192
4193  010534                          8$:
4194
4195                                  .SBTTL  **HEADER RECOGNITION TESTS
4196
4197                                  ;;*****************************************************************
4198                                  ;*TEST 22       BAD SECTOR ERROR
4199                                  ;*
4200                                  ;*      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
4201                                  ;*      SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1
4202                                  ;*      (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
4203                                  ;*      AND ALL OTHER SECTORS GOOD.
4204                                  ;*
4205                                  ;*      ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0,
4206                                  ;*      SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS.  ISSUE A
4207                                  ;*      WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS.
4208                                  ;*      MAKE SURE BAD SECTOR ERROR SET.  ISSUE A WRITE DATA
4209                                  ;*      OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2.  MAKE
4210                                  ;*      SURE NO ERROR SETS.
4211                                  ;*
4212                                  ;;*****************************************************************
4213  010534  000004                  TST22:  SCOPE
4214  010536  012737  000062  001262          MOV     #50.,STIMES     ;;DO 50. ITERATIONS
4215  010544  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
```

```
4216  010552  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
4217  010554  104003                        ERROR   3               ;BAD INIT ERROR
4218
4219  010556  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
4220  010562  000127                        WRHEAD                  ;WRITE HEADER
4221  010564  177676                        -102                    ;WORD COUNT FOR 26 SECTOR MODE
4222  010566  062604                        OBUFF                   ;BUFFER ADDRESS
4223  010570     000                        .BYTE   0               ;SECTOR
4224  010571     000                        .BYTE   0               ;TRACK
4225  010572  000312                        312                     ;CYLINDER
4226
4227
4228  010574  004437  040776                JSR     R4,GENCOM       ;GENERATE HEADERS
4229  010600  000600                        600                     ;WITH NO BS BITS
4230
4231  010602  012700  062606                MOV     #OBUFF+2,R0     ;RESET BIT 15 IN WORD 2 OF
4232  010606  042720  100000                BIC     #BIT15,(R0)+    ;SECTOR 0 HEADER AND BIT 14
4233  010612  042720  100000                BIC     #BIT15,(R0)+    ;IN WORD 2 OF SECTOR 1 HEADER.
4234  010616  005720                        TST     (R0)+           ;ALSO CORRECT THE VRC
4235  010620  042720  040000                BIC     #BIT14,(R0)+
4236  010624  042710  040000                BIC     #BIT14,(R0)
4237
4238  010630  104417                        TLOADRK                 ;LOAD RK REGS
4239  010632  104431                        TWAIT112                ;WAIT FOR INTERRUPT
4240  010634  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4241
4242  010636  104421                        TCHKOP                  ;CHECK IF ANY ERRORS
4243  010640  104004                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4244  010642  012737  010650  001110        MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR
4245  010650  104416                4$:     TSSINIT
4246  010652  104003                        ERROR   3
4247  010654  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
4248  010660  000123                        WRDATA                  ;WRITE DATA
4249  010662  177400                        -400                    ;WORD COUNT
4250  010664  062604                        OBUFF                   ;BUS ADDRESS
4251  010666     000                5$:     .BYTE   0               ;SECT 0
4252  010667     000                        .BYTE   0               ;TRACK 0
4253  010670  000312                        312                     ;CYL 312
4254
4255  010672  104417                1$:     TLOADRK                 ;LOAD RK REGS
4256  010674  104424                        TWAIT32                 ;WAIT FOR INTERRUPT
4257  010676  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4258
4259  010700  022737  000002  010666        CMP     #2,5$           ;JUST READ SECTOR 2?
4260  010706  001415                        BEQ     6$              ;YES - SKIP
4261
4262  010710  104422                        TCHKWE                  ;CHECK OPERATION WITH ERROR
4263  010712  000000                        0
4264  010714  000100                        100                     ;EXPECTED BSE
4265  010716  000000                        0
4266  010720  104004                        ERROR   4 ; OR 5.6.7    ;REPORT ANY DISCREPENCIES
4267
4268  010722  104415                        SCOP1                   ;LOCAL ERROR LOOP TO 4$
4269
4270  010724  122737  000002  010666        CMPB    #2,5$           ;WAS SECTOR SET TO 2
4271  010732  001405                        BEQ     7$              ;YES-SKIP
```

```
 4272  010734  105237  010666          INCB    5$              ;BUMP TO NEXT SECTOR
 4273  010740  000743                  BR      4$              ;LOOP
 4274
 4275  010742  104421          6$:     TCHKOP                  ;CHECK FOR GOOD OPERATION
 4276  010744  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
 4277
 4278  010746                  7$:
 4279                          ;:*****************************************************************
 4280                          ;*TEST 23       HEADER VRC ERROR
 4281                          ;*
 4282                          ;*      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
 4283                          ;*      16 SECTORS WITH BAD HEADER VRC.  ISSUE A WRITE DATA
 4284                          ;*      OF EACH OF THE SECTORS WITH A BAD HEADER VRC.  MAKE
 4285                          ;*      SURE HEADER VRC ERROR SETS.  ISSUE A WRITE DATA TO
 4286                          ;*      A GOOD HEADER AND MAKE SURE NO ERROR OCCURS.
 4287                          ;*
 4288                          ;*****************************************************************
 4289  010746  000004          TST23:  SCOPE
 4290  010750  012737  000062  001262  MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
 4291  010756  012737  000312  001664  MOV     #312,REFMT      ;SET REFORMAT SWITCH
 4292  010764  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
 4293  010766  104003                  ERROR   3               ;BAD INIT ERROR
 4294
 4295  010770  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 4296  010774  000127                  WRHEAD                  ;WRITE HEADER
 4297  010776  177676                  -102                    ;WORD COUNT
 4298  011000  062604                  OBUFF                   ;BUFF ADD
 4299  011002     000                  .BYTE   0               ;SECT
 4300  011003     000                  .BYTE   0               ;TRACK
 4301  011004  000312                  312                     ;CYL
 4302
 4303  011006  004437  040776          JSR     R4,GENCOM
 4304  011012  000600                  600                     ;BUILD HEADERS NO BSE
 4305
 4306  011014  012700  062610          MOV     #OBUFF+4,R0     ;GET ADDRESS OF VRC HDR0
 4307  011020  012703  000001          MOV     #BIT0,R3        ;SET FOR BIT CHANGE SELECT
 4308  011024  030310          1$:     BIT     R3,(R0)         ;CHECK A VRC BIT
 4309  011026  001402                  BEQ     2$              ;SKIP IF ZERO
 4310  011030  040310                  BIC     R3,(R0)         ;ELSE CLEAR IT
 4311  011032  000401                  BR      3$              ;SKIP
 4312  011034  050310          2$:     BIS     R3,(R0)         ;IF ZERO SET IT
 4313  011036  062700  000006  3$:     ADD     #6,R0           ;BUMP TO NEXT VRC WORD
 4314
 4315  011042  006303                  ASL     R3              ;SHIFT THE SELECT
 4316  011044  001367                  BNE     1$              ;IF BIT NOT SHIFTED OUT-LOOP
 4317
 4318  011046  104417                  TLOADRK                 ;LOAD RK REGS
 4319  011050  104431                  TWAT112                 ;WAIT FOR INTERRUPT
 4320  011052  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 4321  011054  104421                  TCHKOP                  ;CHECK OPERATION COMPLETE
 4322  011056  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
 4323
 4324  011060  012737  011066  001110  MOV     #4$,$LPERR      ;SET LOCAL LOOP
 4325  011066  104416          4$:     TSSINIT                 ;CLEAR SUBSYSTEM
 4326  011070  104003                  ERROR   3               ;BAD INIT ERROR
 4327
```

```
 4328  011072  004437  034574              JSR     R4,LRLOAD       ;LOAD "L" REGS
 4329  011076  000123                      WRDATA                  ;WRITE DATA
 4330  011100  177400                      -400                    ;WORD COUNT
 4331  011102  062604                      OBUFF                   ;BUFFER ADD
 4332  011104     000              5$:     .BYTE   0               ;SECT
 4333  011105     000                      .BYTE   0               ;TRACK
 4334  011106  000312                      312                     ;CYL
 4335
 4336  011110  104417                      TLOADRK                 ;LOAD RK REG
 4337  011112  104424                      TWAIT32                 ;WAIT FOR INTERRUPT
 4338  011114  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 4339
 4340  011116  022737  000020  011104      CMP     #16.,5$         ;WAS THIS WRITE SECTOR 16?
 4341  011124  001415                      BEQ     6$              ;YES-SKIP
 4342                                                              ;ELSE
 4343  011126  104422                      TCHKWE                  ;CHECK OPERATION WITH ERROR
 4344  011130  000000                      0
 4345  011132  000040                      40                      ;HVRC  EM EXPECTED
 4346  011134  000000                      0
 4347  011136  104004                      ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
 4348
 4349  011140  104415                      SCOP1                   ;LOCAL LOOP TO 4$
 4350
 4351  011142  105237  011104              INCB    5$              ;BUMP SECTOR IN "L" REG
 4352  011146  022737  000016  011104      CMP     #16.,5$         ;IF SECTOR IS 16 OR LESS
 4353  011154  003744                      BLE     4$              ;LOOP
 4354  011156  000402                      BR      7$              ;ELSE EXIT
 4355  011160  104421              6$:     TCHKOP                  ;CHECK LAST OPERATION NO ERRORS
 4356  011162  104004                      ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
 4357
 4358  011164                      7$:
 4359                              ;;*****************************************************************
 4360                              ;*TEST 24       BAD SECTOR ERROR AND HVRC ERROR
 4361                              ;*
 4362                              ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
 4363                              ;*      BOTH A BAD SECTOR ERROR AND HEADER VRC.  ISSUE A WRITE DATA
 4364                              ;*      TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY HEADER VRC
 4365                              ;*      ERROR SETS.
 4366                              ;*
 4367                              ;;*****************************************************************
 4368  011164  000004              TST24:  SCOPE
 4369  011166  012737  000062  001262      MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
 4370  011174  012737  000312  001664      MOV     #312,REFMT      ;SET REFORMAT SWITCH
 4371  011202  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
 4372  011204  104003                      ERROR   3               ;BAD INIT ERROR
 4373
 4374  011206  004437  034574              JSR     R4,LRLOAD       ;LOAD "L" REG
 4375  011212  000127                      WRHEAD                  ;WRITE HEADER
 4376  011214  177676                      -102                    ;WORD CNT FOR 26 SECTOR MODE
 4377  011216  062604                      OBUFF                   ;BUFF ADD
 4378  011220     000                      .BYTE   0               ;SECTOR
 4379  011221     000                      .BYTE   0               ;TRACK
 4380  011222  000312                      312                     ;CYLINDER
 4381:
 4382  011224  004437  040776              JSR     R4,GENCOM
 4383  011230  000600                      600                     ;BUILD HEADERS-NO BSE
```

```
                                        BIC     #BIT15,OBUFF+2  ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.

                                        TLOADRK                 ;LOAD RK REGS
                                        TWAIT12                 ;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
                                        TWAIT32                 ;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.
                        ;*
                        ;**************************************************************
TST25:  SCOPE
                                        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
                                        MOV     #312,REFMT      ;SET REFORMAT SWITCH
                                        TSSINIT                 ;CLEAR SUBSYSTEM
                                        ERROR   3               ;BAD INIT ERROR

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

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

                                        BIS     #BIT9,OBUFF+200 ;CHANGE FORMAT IN SECTOR 25
                                        BIS     #BIT9,OBUFF+202 ;CORRECT THE VRC
```

```
4440  011372  104417                          TLOADRK                 ;LOAD RK REGS
4441  011374  104431                          TWAT112                 ;WAIT FOR INTERRUPT
4442  011376  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
4443
4444  011400  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4445  011402  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4446
4447  011404  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4448  011410  000123                          WRDATA                  ;WRITE DATA
4449  011412  177400                          -400                    ;400 WORDS
4450  011414  062604                          OBUFF                   ;BUFF ADD
4451  011416     025                          .BYTE   25              ;SECTOR 25
4452  011417     000                          .BYTE   0               ;TRACK 0
4453  011420  000312                          312                     ;CYL 312
4454
4455  011422  104417                          TLOADRK                 ;LOAD RK REGS
4456  011424  104425                          TWAT48                  ;WAIT FOR INTERRUPT
4457  011426  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
4458
4459  011430  104422                          TCHKWE                  ;CHECK OPERATION EXPECTED ERROR
4460  011432  000000                          0
4461  011434  000020                          20                      ;OPI EXPECTED
4462  011436  000000                          0
4463  011440  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
4464                                ;;*****************************************************************
4465                                ;*TEST 26        OPI WITH HVRC ERROR
4466                                ;*
4467                                ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
4468                                ;*      ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
4469                                ;*      ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312,
4470                                ;*      TRACK 0, SECTOR 17.  THAT BOTH OPERATION INCOMPLETE
4471                                ;*      AND HEADER VRC SET.
4472                                ;*
4473                                ;*****************************************************************
4474  011442  000004                TST26:  SCOPE
4475  011444  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
4476  011452  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4477  011460  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4478  011462  104003                          ERROR   3               ;BAD INIT ERROR
4479
4480  011464  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4481  011470  000127                          WRHEAD                  ;WRITE HEADER
4482  011472  177676                          -102                    ;WORD COUNT FOR 26 SECT MODE
4483  011474  062604                          OBUFF                   ;BUS ADDRESS
4484  011476     000                          .BYTE   0               ;SECTOR
4485  011477     000                          .BYTE   0               ;TRACK
4486  011500  000312                          312                     ;CYLINDER
4487
4488  011502  004437  040776                  JSR     R4,GENCOM
4489  011506  000600                          600                     ;BUILD HEADER- NO BSE ERRORS
4490
4491  011510  012700  062740                  MOV     #OBUFF+134,R0   ;GET ADDRESS 2ND WORD HDR 17(8)
4492  011514  052720  001000                  BIS     #BIT9,(R0)+     ;SET FORMAT 24 SECT PER TRACK
4493  011520  052720  001000                  BIS     #BIT9,(R0)+     ;SET VRC BIT
4494  011524  062700  000004                  ADD     #4,R0           ;BUMP TO HVRC WORD HDR 20(8)
4495  011530  032710  000001                  BIT     #BIT0,(R0)      ;TEST BIT 0
```

```
4496  011534  001403                          BEQ     1$              ;RESET-SKIP
4497  011536  042710  000001                  BIC     #BIT0,(R0)      ;CLEAR BIT
4498  011542  000402                          BR      2$
4499  011544  052710  000001          1$:     BIS     #BIT0,(PC)      ;SET BIT
4500                                          ;FORCE OPI AND HVRC ERROR
4501  011550  104417                  2$:     TLOADRK                 ;LOAD RK REGS
4502  011552  104431                          TWAIT12                 ;WAIT FOR INTERRUPT
4503  011554  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4504
4505  011556  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4506  011560  104004                          ERROR   4 ; OR 5,6,7    ;YES-REPORT ALL ERRORS
4507
4508  011562  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4509  011566  000123                          WRDATA                  ;WRITE DATA
4510  011570  177400                          -400                    ;400 WORDS
4511  011572  062604                          OBUFF                   ;BUFF ADDRESS
4512  011574     017                          .BYTE   17              ;SECT 17
4513  011575     000                          .BYTE   0               ;TRACK 0
4514  011576  000312                          312                     ;CYLINDER 312
4515
4516  011600  104417                          TLOADRK                 ;LOAD RK REGS
4517  011602  104425                          TWAIT48                 ;WAIT FOR INTERRUPT
4518  011604  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
4519
4520  011606  104422                          TCHKWE                  ;CHECK WITH EXPECTED ERROR
4521  011610  000000                          0
4522  011612  000060                          60                      ;HVRC ERR & OPI EXPECTED
4523  011614  000000                          0
4524  011616  104004                          ERROR   4  ;OR 5,6,7
4525                                  ;;*****************************************************************
4526                                  ;*TEST 27       HVRC IGNORE ON NON-ADDRESSED SECTOR
4527                                  ;*
4528                                  ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
4529                                  ;*      ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
4530                                  ;*      AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
4531                                  ;*      END OF THE OPERATION
4532                                  ;*
4533                                  ;;*****************************************************************
4534  011620  000004                  TST27:  SCOPE
4535  011622  012737  000100  001262          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
4536  011630  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4537
4538  011636  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4539  011640  104003                          ERROR   3               ;BAD INIT ERROR
4540
4541  011642  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGISTERS
4542  011646  000127                          WRHEAD                  ;WRITE HEADER
4543  011650  177676                          -102                    ;WORD COUNT FOR 26 SECTOR MODE
4544  011652  062604                          OBUFF                   ;BUFF ADD
4545  011654     000                          .BYTE   0               ;SECTOR
4546  011655     000                          .BYTE   0               ;TRACK
4547  011656  000312                          312                     ;CYLINDER
4548
4549  011660  004437  040776                  JSR     R4,GENCOM
4550  011664  000600                          600                     ;BUILD HEADERS-NO BSE ERRORS
4551
```

```
4552  011666  012700  062750          MOV     #OBUFF+144,R0   ;ADDRESS OF HEAD 20 HVRC WORD
4553  011672  012701  000002          MOV     #BIT1,R1        ;BIT 1 CONSTANT
4554  011676  030110                  BIT     R1,(R0)         ;TEST BIT 1 SET
4555  011700  001402                  BEQ     1$              ;RESET-SKIP
4556  011702  040110                  BIC     R1,(R0)         ;ELSE CLEAR BIT 1
4557  011704  000401                  BR      2$              ;SKIP
4558  011706  050110          1$:     BIS     R1,(R0)         ;SET BIT 1
4559
4560  011710  104417          2$:     TLOADRK                 ;LOAD RK REGS
4561  011712  104431                  TWAT112                 ;WAIT FOR INTERRUPT
4562  011714  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE
4563
4564  011716  104421                  TCHKOP                  ;CHECK FOR ANY ERROR
4565  011720  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4566
4567  011722  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGISTER
4568  011726  000123                  WRDATA                  ;WRITE DATA
4569  011730  177400                  -400                    ;WORD COUNT
4570  011732  062604                  OBUFF                   ;BUFF ADD
4571  011734     021                  .BYTE   21              ;SECTOR
4572  011735     000                  .BYTE   0               ;TRACK
4573  011736  000312                  312                     ;CYLINDER
4574  011740  104417                  TLOADRK                 ;LOAD Rk REGS
4575  011742  104424                  TWAT32                  ;WAIT FOR INTERRUPT
4576  011744  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4577
4578  011746  104421                  TCHKOP                  ;CHECK FOR ANY ERROR
4579  011750  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS.
4580
4581
4582                          .SBTTL  **DATA TRANSFER TESTS
4583
4584                          ;;*****************************************************************
4585                          ;*TEST 30       WRITE AND READ ONE SECTOR
4586                          ;*
4587                          ;*      FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 0
4588                          ;*      TO AGREED WITH BAD SECTOR INFORMATION.  ISSUE A WRITE DATA
4589                          ;*      OF ONE SECTOR ON CYLINDER 312, TRACK 0.  READ IT BACK TO
4590                          ;*      MAKE SURE IT AGREES WITH WHAT IS WRITTEN.
4591                          ;*
4592                          ;;*****************************************************************
4593  011752  000004          TST30:  SCOPE
4594  011754  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
4595  011762  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4596  011770  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4597  011772  104003                          ERROR   3               ;BAD INIT ERROR
4598
4599  011774  012737  000312  012022          MOV     #312,7$         ;PRESET CYL POINTER
4600  012002  105037  012021                  CLRB    2$              ;CLEAR TRACK POINTER
4601
4602  012006  004437  034574          1$:     JSR     R4,LRLOAD       ;LOAD "L" REG
4603  012012  000127                          WRHEAD                  ;WRITE HEADER
4604  012014  177676                          -102                    ;WORD COUNT FOR 26 SECTOR MODE
4605  012016  062604                          OBUFF                   ;BUFF ADDRESS
4606  012020     000                          .BYTE   0               ;SECTOR
4607  012021     000          2$:             .BYTE   0               ;TRACK
```

```
                                7$:     312                     ;CYLINDER
                                        JSR     R4,GENCOM
                                        1200                    ;BUILD HDRS-INCLUDE BAD SECTORS
                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT112                 ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                        TCHKOP                  ;CHECK FOR ANY ERRORS
                                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
                                        CMP     #313,7$         ;TEST IF DONE 313 TK 0
                                        BEQ     3$              ;YES - SKIP
                                        CMPB    2$,#2           ;DID WE JUST FORMAT TRACK 2
                                        BEQ     8$              ;YES-SKIP
                                        INCB    2$              ;BUMP TO NEXT TRACK
                                        BR      1$              ;GO FORMAT NEXT TRACK
                                8$:     CLRB    2$              ;CLEAR TRACK POINTER
                                        INC     7$              ;BUMP CYL TO 313
                                        BR      1$              ;GO FORMAT 313 TK 0
                                3$:     JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        WRDATA                  ;WRITE DATA
                                        -400                    ;ONE SECTOR WORD COUNT
                                        OBUFF                   ;BUFF ADDRESS
                                        .BYTE   12              ;SECTOR 12
                                        .BYTE   0               ;TRACK 0
                                        312                     ;CYLINDER 312
                                        JSR     R4,GENCOM
                                        1                       ;BUILD DATA PATTERN 1
                                        400                     ;400 WORDS LONG
                                        MOV     #4$,$LPERR      ;SET FOR LOCAL LOOP
                                4$:     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
                                        RDDATA                  ;READ DATA
                                        -400                    ;400 WORDS
                                        IBUFF                   ;BUFF ADD
                                        .BYTE   12              ;SECTOR 12
                                        .BYTE   0               ;TRACK 0
                                        312                     ;CYL 312
                                        TLOADRK                 ;LOAD RK
                                        TWAT32                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE
                                        TCHKOP                  ;CHECK FOR ANY ERRORS
                                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
```

```
                                                JSR     R4,GENCOM
                                                100001                  ;GO COMPARE DATA TO PATTERN 1
                                                400                     ;400 WORDS LONG
                                                BR      6$              ;GOOD RETURN-NO DATA ERRORS
                                                ERROR   15              ;ERROR RETURN

                                                MOV     ERRLMT,R0       ;GET ERROR LIMIT
                                        5$:     DEC     R0              ;DEC LIMIT
                                                BEQ     6$              ;EXIT IF 0
                                                JSR     R4,GENCOM
                                                040000                  ;CONTINUE COMPARE
                                                BR      6$              ;EXIT IF NO MORE ERRORS
                                                ERROR   16              ;ELSE REPORT MISCOMPARE
                                                BR      5$              ;LOOP
                                        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.
                                        ;*
                                        ;;*****************************************************************
                                        TST31:  SCOPE
                                                MOV     #50.,$TIMES     ;;DO 50. 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   12              ;SECTOR 12
                                                .BYTE   1               ;TRACK 1
                                                312                     ;CYLINDER 312

                                                BIS     #BAI,L.CS2      ;SET INCREMENT INHIBIT
                                                JSR     R4,GENCOM       ;BUILD PATTERN
                                                16                      ;PATTERN 16
                                                400                     ;400 WORDS

                                                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,LRLOAD       ;LOAD "L" REGS
                                                RDDATA                  ;RDDATA
                                                -400                    ;-400 WORDS
                                                IBUFF                   ;IBUFF IS BUFF ADDRESS
```

```
4720  012342  012                       .BYTE   12              ;SECTOR 12
4721  012343  001                       .BYTE   1               ;TRACK 1
4722  012344  000312                    312                     ;CYLINDER 312
4723
4724  012346  012700  000377            MOV     #377,R0         ;SET COUNT TO SET OBUFF TO BE
4725  012352  012701  062606            MOV     #OBUFF+2,R1     ;ALL THE FIRST WORD OF PATTERN
4726  012356  012703  062604            MOV     #OBUFF,R3
4727
4728  012362  011321              1$:    MOV     (R3),(R1)+      ;MOV THE WORD
4729  012364  005300                     DEC     R0
4730  012366  001375                     BNE     1$              ;LOOP UNTIL ALL WORDS SET
4731
4732  012370  104417                     TLOADRK                 ;LOAD RK REGS
4733  012372  104424                     TWAT32                  ;WAIT FOR INTERRUPT
4734  012374  104002                     ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4735
4736  012376  104421                     TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4737  012400  104004                     ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4738
4739  012402  004437  040776            JSR     R4,GENCOM       ;COMPARE THE DATA
4740  012406  100000                     100000
4741  012410  000400                     400
4742  012412  000413                     BR      2$
4743  012414  104015                     ERROR   15
4744  012416  013700  001634            MOV     ERRLMT,R0       ;GET ERROR LIMIT
4745  012422  005300              64$:   DEC     R0              ;DECREMENT COUNT
4746  012424  001406                     BEQ     65$             ;IF ZERO - EXIT
4747  012426  004437  040776            JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4748  012432  040000                     40000
4749  012434  000402                     BR      65$             ;NO MORE ERRORS - EXIT
4750  012436  104016                     ERROR   16              ;REPORT NEXT ERROR
4751  012440  000770                     BR      64$             ;LOOP
4752  012442                        65$:
4753
4754  012442                        2$:
4755                                ;;*****************************************************************
4756                                ;*TEST 32       WRITE DATA ADDRESS GREATER THAN 32K
4757                                ;*
4758                                ;*      ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
4759                                ;*      MAKE SURE CORRECT DATA IS ON DISK.
4760                                ;*
4761                                ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
4762                                ;*              OF MEMORY IS PRESENT.
4763                                ;*
4764                                ;;*****************************************************************
4765  012442  000004                TST32:  SCOPE
4766  012444  012737  000062  001262        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
4767  012452  123727  001326  000001        CMPB    $MAMS1,#1       ;TEST IF >32K MEM
4768  012460  002016                        BGE     2$              ;YES-SKIP
4769
4770  012462  032737  000002  001656        BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT ALREADY MADE
4771  012470  001011                        BNE     1$              ;YES -SKIP
4772
4773  012472  104401  050411                TYPE    ,OPR011         ;"INSUFFICIENT MEMORY DATA TRANSFER WITH
4774  012476  104401  050554                TYPE    ,OPR012         ;ADDRESS >32K
4775  012502  052737  000002  001656        BIS     #MEMSZB,OPTFLG  ;SET FLAG
```

```
4776  012510  104401  050570                    TYPE    ,OPR015         ;BYPASSED"
4777  012514  000470                1$:         BR      4$              ;EXIT
4778  012516  012737  012524  001110  2$:       MOV     #5$,$LPERR      ;SET LOCAL LOOP ON ERROR ADDRESS
4779  012524                          5$:
4780  012524  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
4781  012526  104003                            ERROR   3               ;BAD INIT ERROR
4782
4783  012530  004437  034574                    JSR     R4,LRLOAD       ;LOAD "L" REGS
4784  012534  000123                            WRDATA                  ;WRITE DATA
4785  012536  177400                            -400                    ;400 WORDS
4786  012540  177770                            177770                  ;BUS ADDRESS IN 32K -10 BYTES
4787  012542     016                            .BYTE   16              ;SECTOR 16
4788  012543     000                            .BYTE   0               ;TRACK 0
4789  012544  000312                            312                     ;CYLINDER 312
4790  012546  004437  040776                    JSR     R4,GENCOM       ;GENERATE DATA
4791  012552  010010                            10010                   ;PATTERN 10, MEM. MANAGEMENT FOR DEST.
4792  012554  001777                            1777                    ;RELOCATION ARGUMENT
4793  012556  000400                            400                     ;400 WORDS
4794
4795  012560  104417                            TLOADRK                 ;LOAD RK REGS
4796  012562  104430                            TWAT96                  ;WAIT FOR INTERRUPT
4797  012564  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4798
4799  012566  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4800  012570  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4801  012572  104415                            SCOP1                   ;LOCAL LOOP ON ERROR TO 5$
4802
4803  012574  004437  040776                    JSR     R4,GENCOM       ;CLEAR IBUFF TO 1'S.
4804  012600  002007                            2007
4805  012602  001000                            1000
4806
4807  012604  004437  034574                    JSR     R4,LRLOAD       ;LOAD "L" REGS
4808  012610  000121                            RDDATA                  ;RDDATA
4809  012612  177400                            -400                    ;-400 WORDS
4810  012614  060604                            IBUFF                   ;IBUFF IS BUFF ADDRESS
4811  012616     016                            .BYTE   16              ;SECTOR 16
4812  012617     000                            .BYTE   0               ;TRACK 0
4813  012620  000312                            312                     ;CYLINDER 312
4814  012622  104417                            TLOADRK                 ;LOAD RK REGS
4815  012624  104424                            TWAT32                  ;WAIT FOR INTERRUPT
4816  012626  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4817  012630  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4818  012632  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4819  012634  004437  040776                    JSR     R4,GENCOM       ;COMPARE
4820  012640  110000                            110000                  ;MEMORY MANAGEMENT FOR DESTINATION
4821  012642  001777                            1777                    ;RELOCATION ARGUMENT
4822  012644  000400                            400                     ;400 WORDS
4823  012646  000413                            BR      4$              ;NO ERROR-SKIP
4824  012650  104015                            ERROR   15              ;REPORT FIRST MISCOMPARE
4825  012652  013700  001634                    MOV     ERRLMT,R0       ;GET ERROR LIMIT
4826  012656  005300                64$:        DEC     R0              ;DECREMENT COUNT
4827  012660  001406                            BEQ     65$             ;IF ZERO - EXIT
4828  012662  004437  040776                    JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4829  012666  050000                            50000
4830  012670  000402                            BR      65$             ;NO MORE ERRORS - EXIT
4831  012672  104016                            ERROR   16              ;REPORT NEXT ERROR
```

```
                        000770                  BR      64$             ;LOOP
                                        65$:
                                        4$:
                                        ;;*****************************************************************
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
012676  000004                T$T33:  SCOPE
012700  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
012706  123727  001326  000001          CMPB    $MAMSI,#1       ;CHECK IF >32k MEMORY
012714  002001                          BGE     2$              ;YES-SKIP
012716  000462                1$:       BR      5$              ;EXIT
012720  012737  012726  001110  2$:     MOV     #3$,$_PERR      ;SET LOCAL ERROR LOOP
012726                          3$:
012726  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
012730  104003                          ERROR   3               ;BAD INIT ERROR
012732  004437  034574                  JSR     R4,LR_LOAD      ;LOAD "L" REGS
012736  000123                          WRDATA                  ;WRDATA
012740  177400                          -400                    ;-400 WORDS
012742  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
012744     017                          .BYTE   17              ;SECTOR 17
012745     000                          .BYTE   0               ;TRACK 0
012746  000312                          312                     ;CYLINDER 312
012750  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA IN OBUFF
012754  000011                          11                      ;PATTERN 11
012756  000400                          400                     ;400 WORDS
012760  104417                          TLOADRK                 ;LOAD RK REGS
012762  104430                          TWAT96                  ;WAIT FOR INTERRUPT
012764  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
012766  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
012770  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
012772  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REG
012776  000121                          RDDATA                  ;READ DATA
013000  177400                          -400                    ;400 WORDS
013002  177770                          177770                  ;ACCROSS 32K BOUNDARY
013004     017                          .BYTE   17              ;SECTOR 17
013005     000                          .BYTE   0               ;TRACK 0
013006  000312                          312                     ;CYL 312
013010  104417                          TLOADRK                 ;LOAD RK REGS
013012  104424                          TWAT32                  ;WAIT FOR INTERRUPT
013014  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
013016  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
013020  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
```

```
                                T33     READ DATA ADDRESS GREATER THAN 32k

 013022  004437  040776                        JSR     R4,GENCOM       ;COMPARE DATA
 013026  120000                                120000                  ;MEMORY MANAGEMENT WITH SOURCE
 013030  001777                                1777                    ;RELOCATION ARGUMENT
 013032  000400                                400                     ;COMPARE 400 WORDS
 013034  000413                                BR      5$              ;NO MISCOMPARE-EXIT
 013036  104015                                ERROR   15              ;REPORT FIRST MISCOMPARE
 013040  013700  001634                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
                                64$:    DEC     R0                      ;DECREMENT COUNT
 013044  005300                                BEQ     65$             ;IF ZERO - EXIT
 013046  001406                                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
 013050  004437  040776                        60000
 013054  060000                                BR      65$             ;NO MORE ERRORS - EXIT
 013056  000402                                ERROR   16              ;REPORT NEXT ERROR
 013060  104016                                BR      64$             ;LOOP
 013062  000770
                                65$:
 013064                         5$:
                                ;*****************************************************************
                                ;*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.
                                ;*
                                ;*****************************************************************
 013064  000004                 T$T34:  SCOPE
 013066  012737  000C62  001262         MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
 013074  123727  001326  000002         CMPB    $MAMS1,#2       ;CHECK IF >64K MEMORY
 013102  002016                         BGE     2$              ;YES-SKIP
 013104  032737  000002  001656 6$:     BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
 013112  001011                         BNE     1$              ;NO-SKIP

 013114  104401  050411                 TYPE    .OPR011         ;"INSUFFICIENT MEMORY-DATA XFER WITH
 013120  104401  050560                 TYPE    .OPR013         ;ADDRESS >64K
 013124  104401  050570                 TYPE    .OPR015         ;BYPASSED"
 013130  052737  000002  001656         BIS     #MEMSZB,OPTFLG  ;SET FLAG
 013136  000467                 1$:     BR      5$

 013140  012737  013146  001110 2$:     MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR

 013146                         3$:
 013146  104416                         TSSINIT                 ;CLEAR SUBSYSTEM
 013150  104003                         ERROR   3               ;BAD INIT ERROR
 013152  004437  040776                 JSR     R4,GENCOM       ;GENERATE DATA, PATTERN 11
 013156  010011                         10011                   ;MEM MANAGEMENT ON DESTINATION
 013160  003777                         3777                    ;RELOCATION ARGUMENT
 013162  000400                         400                     ;400 WORDS

 013164  004437  034574                 JSR     R4,LRLOAD       ;LOAD "L" REGS
 013170  000523                         WRDATA!BA16             ;WRITE DATA AND SET BA16
 013172  177400                         -400                    ;400 WORDS
 013174  177770                         177770                  ;ACCROSS 64K BOUNDARY
 013176     020                         .BYTE   20              ;SECTOR 20
 013177     000                         .BYTE   0               ;TRACK 0
 013200  000312                         312                     ;CYLINDER 312
```

```
4944
4945  013202  104417                          TLOADRK                 ;LOAD RK REGS
4946  013204  104430                          TWAT96                  ;WAIT FOR INTERRUPT
4947  013206  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4948
4949  013210  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4950  013212  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4951  013214  004437  040776                  JSR     R4,GENCOM        CLEAR IBUFF TO 1'S
4952  013220  002007                          2007
4953  013222  001000                          1000
4954
4955  013224  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4956  013230  000121                          RDDATA                  ;RDDATA
4957  013232  177400                          -400                    ;-400 WORDS
4958  013234  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
4959  013236     020                          .BYTE   20              ;SECTOR 20
4960  013237     000                          .BYTE   0               ;TRACK 0
4961  013240  000312                          312                     ;CYLINDER 312
4962  013242  104417                          TLOADRK                 ;LOAD RK REGS
4963  013244  104424                          TWAT32                  ;WAIT FOR INTERRUPT
4964  013246  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4965
4966  013250  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4967  013252  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4968  013254  004437  040776                  JSR     R4,GENCOM       ;CHECK DATA
4969  013260  110000                          110000                  ;MEMORY MANAGEMENT WITH DESTINATION
4970  013262  003777                          3777                    ;RELOCATION ARGUMENT
4971  013264  000400                          400                     ;400 WORDS
4972  013266  000413                          BR      5$              ;NO MISCOMPARES-SKIP
4973  013270  104015                          ERROR   15              ;REPORT FIRST ERROR
4974
4975  013272  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
4976  013276  005300                  64$:    DEC     R0              ;DECREMENT COUNT
4977  013300  001406                          BEQ     65$             ;IF ZERO - EXIT
4978  013302  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4979  013306  050000                          50000
4980  013310  000402                          BR      65$             ;NO MORE ERRORS - EXIT
4981  013312  104016                          ERROR   16              ;REPORT NEXT ERROR
4982  013314  000770                          BR      64$             ;LOOP
4983  013316                          65$:
4984  013316                          5$:
4985                                          ;;*****************************************************************
4986                                          ;*TEST 35       READ DATA ADDRESS GREATER THAN 64K
4987                                          ;*
4988                                          ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
4989                                          ;*      CHECK MEMORY FOR CORRECT TRANSFER.
4990                                          ;*
4991                                          ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
4992                                          ;*              OF MEMORY IS PRESENT.
4993                                          ;*
4994                                          ;;*****************************************************************
4995  013316  000004                  TST35:  SCOPE
4996  013320  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
4997  013326  123727  001326  000002          CMPB    $MAMS1,#2       ;CHECK IF >64K MEMORY
4998  013334  002001                          BGE     2$              ;YES-SKIP
4999  013336  000462                  1$:     BR      5$              ;EXIT
```

```
5000
5001  013340  012737  000032  001110  2$:     MOV     #32,$LPERR      ;SET LOCAL LOOP ON ERROR
5002
5003  013346                          3$:
5004  013346  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5005  013350  104003                          ERROR   3               ;BAD INIT ERROR
5006  013352  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5007  013356  000123                          WRDATA                  ;WRDATA
5008  013360  177400                          -400                    ;-400 WORDS
5009  013362  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5010  013364     021                          .BYTE   21              ;SECTOR 21
5011  013365     000                          .BYTE   0               ;TRACK 0
5012  013366  000312                          312                     ;CYLINDER 312
5013  013370  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
5014  013374  000012                          12                      ;PATTERN 12
5015  013376  000400                          400                     ;400 WORDS
5016
5017  013400  104417                          TLOADRK                 ;LOAD RK REGS
5018  013402  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5019  013404  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5020
5021  013406  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5022  013410  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5023  013412  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5024  013416  000521                          RDDATA!BA16             ;READ DATA AND SET BA16
5025  013420  177400                          -400                    ;400 WORDS
5026  013422  177770                          177770                  ;ACCROSS 64K BOUNDARY
5027  013424     021                          .BYTE   21              ;FROM SECTOR 21
5028  013425     000                          .BYTE   0               ;TRACK 0
5029  013426  000312                          312                     ;CYLINDER 312
5030
5031  013430  104417                          TLOADRK                 ;LOAD RK REGS
5032  013432  104424                          TWAT32                  ;WAIT FOR INTERRUPT
5033  013434  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5034
5035  013436  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5036  013440  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5037  013442  004437  040776                  JSR     R4,GENCOM       ;COMPARE DATA
5038  013446  120000                          120000                  ;MEM MANAGEMENT WITH SOURCE
5039  013450  003777                          3777                    ;RELOCATION ARGUMENT
5040  013452  000400                          400                     ;400 WORDS
5041  013454  000413                          BR      5$              ;NO MISCOMPARES-SKIP
5042  013456  104015                          ERROR   15              ;REPORT FIRST ERROR
5043
5044  013460  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5045  013464  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5046  013466  001406                          BEQ     65$             ;IF ZERO - EXIT
5047  013470  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5048  013474  060000                          60000
5049  013476  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5050  013500  104016                          ERROR   16              ;REPORT NEXT ERROR
5051  013502  000770                          BR      64$             ;LOOP
5052  013504                          65$:
5053
5054  013504                          5$:
5055                                  ;;*****************************************************************
```

```
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
013504  000004                  TST36:  SCOPE
013506  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
013514  123727  001326  000003          CMPB    $MAMS1,#3       ;CHECK IF >96K MEMORY
013522  002016                  BGE     3$              ;YES-SKIP
013524  032737  000002  001656  1$:     BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
013532  001011                  BNE     2$              ;NO-SKIP
                                
013534  104401  050411          TYPE    .OPR011         ;"INSUFFICIENT MEMORY-DATA TRANSFET WITH
013540  104401  050564          TYPE    .OPR014         ;ADDRESS >96K BYPASSED"
013544  104401  050570          TYPE    .OPR015
013550  052737  000002  001656  BIS     #MEMSZB,OPTFLG  ;SET REPORT FLAG
013556  000463                  2$:     BR      6$

013560  012737  013566  001110  3$:     MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR

013566                          4$:
013566  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
013570  104003                  ERROR   3               ;BAD INIT ERROR
013572  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REG
013576  001123                  WRDATA!BA17             ;WRITE DATA AND BA17
013600  177400                  -400                    ;400 WORDS FROM
013602  177770                  177770                  ;ACCROSS 96K BOUNDARY
013604     022                  .BYTE   22              ;TO SECTOR 22
013605     000                  .BYTE   0               ;TRACK 0
013606  000312                  312                     ;CYL 312
013610  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA
013614  010013                  10013                   ;PATTERN 13 MEM MAN WITH DEST.
013616  005777                  5777                    ;RELOCATION ARGUMENT
013620  000400                  400                     ;400 WORDS

013622  104417                  TLOADRK                 ;LOAD RK REGS
013624  104430                  TWAT96                  ;WAIT FOR INTERRUPT
013626  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

013630  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
013632  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

013634  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
013640  000121                  RDDATA                  ;RDDATA
013642  177400                  -400                    ;-400 WORDS
013644  060604                  IBUFF                   ;IBUFF IS BUFF ADDRESS
013646     022                  .BYTE   22              ;SECTOR 22
013647     000                  .BYTE   0               ;TRACK 0
013650  000312                  312                     ;CYLINDER 312

013652  104417                  TLOADRK                 ;LOAD RK REGS
013654  104424                  TWAT32                  ;WAIT FOR INTERRUPT
```

```
013656  104002                          ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR

013660  104421                          TCHKOP                          ;CHECK OPERATION FOR ANY ERRORS
013662  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

013664  004437  040776                  JSR     R4,GENCOM               ;COMPARE DATA
013670  110000                          110000                          ;MEM MANAGEMENT WITH DESTINATION
013672  005777                          5777                            ;RELOCATION ARGUMENT
013674  000400                          400                             ;400 WORDS
013676  000413                          BR      6$                      ;NO MISCOMPARES-BRANCH
013700  104015                          ERROR   15                      ;REPORT 1ST ERROR

013702  013700  001634                  MOV     ERRLMT,R0               ;GET ERROR LIMIT
013706  005300                  64$:    DEC     R0                      ;DECREMENT COUNT
013710  001406                          BEQ     65$                     ;IF ZERO - EXIT
013712  004437  040776                  JSR     R4,GENCOM               ;CONTINUE DATA COMPARE
013716  050000                          50000
013720  000402                          BR      65$                     ;NO MORE ERRORS - EXIT
013722  104016                          ERROR   16                      ;REPORT NEXT ERROR
013724  000770                          BR      64$                     ;LOOP
013726                          65$:

013726                          6$:
                                ;;*********************************************************************
                                ;*TEST 37        READ DATA ADDRESS GREATER THAN 96K
                                ;*
                                ;*       ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
                                ;*       CHECK MEMORY FOR CORRECT TRANSFER.
                                ;*
                                ;*       NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
                                ;*               OF MEMORY IS PRESENT.
                                ;*
                                ;;*********************************************************************
013726  000004                  TST37:  SCOPE
013730  012737  000062  001262          MOV     #50.,$TIMES             ;;DO 50. ITERATIONS
013736  123727  001326  000003          CMPB    $MAMS1,#3               ;CHECK IF >96K MEMORY
013744  002001                          BGE     3$                      ;YES-SKIP
013746  000462                  2$:     BR      6$

013750  012737  013756  001110  3$:     MOV     #4$,$LPERR              ;SET LOCAL LOOP ON ERROR

013756                          4$:
013756  104416                          TSSINIT                         ;CLEAR SUBSYSTEM
013760  104003                          ERROR   3                       ;BAD INIT ERROR

013762  004437  034574                  JSR     R4,LRLOAD               ;LOAD "L" REGS
013766  000123                          WRDATA                          ;WRDATA
013770  177400                          -400                            ;-400 WORDS
013772  062604                          OBUFF                           ;OBUFF IS BUFF ADDRESS
013774     005                          .BYTE   5                       ;SECTOR 5
013775     000                          .BYTE   0                       ;TRACK 0
013776  000312                          312                             ;CYLINDER 312
014000  004437  040776                  JSR     R4,GENCOM               ;GENERATE DATA
014004  000014                          14                              ;PATTERN 14
014006  000400                          400                             ;400 WORDS
```

```
5168  014010  104417                    TLOADRK                 ;LOAD RK REGS
5169  014012  104430                    TWAT96                  ;WAIT FOR INTERRUPT
5170  014014  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5171
5172  014016  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5173  014020  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5174  014022  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
5175  014026  001121                    RDDATA!BA17             ;READ DATA WITH BA17 SET
5176  014030  177400                    -400                    ;400 WORDS
5177  014032  177770                    177770                  ;ACCROSS 96K BOUNDARY
5178  014034     005                    .BYTE   5               ;FROM SECTOR 5
5179  014035     000                    .BYTE   0               ;TRACK 0
5180  014036  000312                    312                     ;CYL 312
5181
5182  014040  104417                    TLOADRK                 ;LOAD RK REGS
5183  014042  104424                    TWAT32                  ;WAIT FOR INTERRUPT
5184  014044  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5185
5186  014046  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5187  014050  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5188  014052  004437  040776            JSR     R4,GENCOM       ;COMPARE DATA
5189  014056  120000                    120000                  ;MEM MANAGEMENT WITH SOURCE
5190  014060  005777                    5777                    ;RELOCATION ARGUMENT
5191  014062  000400                    400                     ;400 WORDS
5192  014064  000413                    BR      6$              ;NO MISCOMPARES-SKIP
5193  014066  104015                    ERROR   15              ;REPORT FIRST ERROR
5194
5195  014070  013700  001634            MOV     ERRLMT,R0       ;GET ERROR LIMIT
5196  014074  005300              64$:  DEC     R0              ;DECREMENT COUNT
5197  014076  001406                    BEQ     65$             ;IF ZERO - EXIT
5198  014100  004437  040776            JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5199  014104  060000                    60000
5200  014106  000402                    BR      65$             ;NO MORE ERRORS - EXIT
5201  014110  104016                    ERROR   16              ;REPORT NEXT ERROR
5202  014112  000770                    BR      64$             ;LOOP
5203  014114                      65$:
5204
5205  014114                      6$:
5206                                    ;;*****************************************************************
5207                                    ;*TEST 40       PARTIAL SECTOR WRITE DATA
5208                                    ;*
5209                                    ;*      ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312,
5210                                    ;*      HEAD 0, SECTOR 0.  MAKE SURE THE SECTOR WAS
5211                                    ;*      ZERO FILLED CORRECTLY.
5212                                    ;*
5213                                    ;;*****************************************************************
5214  014114  000004              TST40:  SCOPE
5215  014116  012737  000062  001262    MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5216  014124  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
5217  014126  104003                    ERROR   3               ;BAD INIT ERROR
5218
5219  014130  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REG
5220  014134  000123                    WRDATA                  ;WRITE DATA
5221  014136  177675                    -103                    ;WORD COUNT PARTIAL SECTOR
5222  014140  062604                    OBUFF                   ;BUFF ADDRESS
5223  014142     007                    .BYTE   7               ;SECTOR 7
```

```
5224  014143  000                     .BYTE   0               ;TRACK 0
5225  014144  000312                  312                     ;CYLINDER 312
5226
5227  014146  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA
5228  014152  000003                  3                       ;PATTERN 3
5229  014154  000400                  400                     ;400 WORDS
5230
5231  014156  104417                  TLOADRK                 ;LOAD RK REGS
5232  014160  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5233  014162  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5234
5235  014164  104421                  TCHKOP                  ;CHECK FOR ANY ERROR
5236  014166  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERROR
5237
5238  014170  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5239
5240  014174  000121                  RDDATA                  ;READ DATA
5241  014176  177400                  -400                    ;ONE FULL SECTOR
5242  014200  060604                  IBUFF                   ;BUFF ADDRESS
5243  014202  007                     .BYTE   7               ;SECTOR 7
5244  014203  000                     .BYTE   0               ;TRACK 0
5245  014204  000312                  312                     ;CYLINDER 312
5246
5247  014206  004437  040776          JSR     R4,GENCOM
5248  014212  002007                  2007                    ;CLEAR IBUFF TO ALL ONES
5249  014214  000400                  400
5250
5251  014216  104417                  TLOADRK                 ;LOAD RK REGS
5252  014220  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5253  014222  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE
5254
5255  014224  104421                  TCHKOP                  ;CHECK FOR ANY ERRORS
5256  014226  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
5257
5258  014230  012701  063012          MOV     #OBUFF+206,R1   ;CLEAR THE LAST 205 WORDS
5259  014234  012700  000275          MOV     #275,R0         ;OF THE OUTPUT BUFFER TO ZERO
5260  014240  005021          1$:     CLR     (R1)+           ;TO VERIFY THE PARTIAL SECTOR
5261  014242  005300                  DEC     R0              ;WRITE 0 FILLED THE SECTOR
5262  014244  001375                  BNE     1$
5263  014246  004437  040776          JSR     R4,GENCOM
5264  014252  100000                  100000                  ;COMPARE OBUFF & IBUFF.
5265  014254  000400                  400                     ;ALL 400 WORDS
5266  014256  000413                  BR      3$              ;NO ERRORS-EXIT
5267  014260  104015                  ERROR   15              ;REPORT FIRST COMPARE ERROR
5268
5269  014262  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5270  014266  005300          2$:     DEC     R0              ;DECREMENT IT
5271  014270  001406                  BEQ     3$              ;IF ZERO-EXIT
5272  014272  004437  040776          JSR     R4,GENCOM
5273  014276  040000                  40000                   ;CONTINUE COMPARE
5274  014300  000402                  BR      3$              ;NO MORE ERRORS-EXIT
5275  014302  104016                  ERROR   16              ;REPORT NEXT COMPARE ERROR
5276  014304  000770                  BR      2$              ;LOOP
5277
5278  014306          3$:
5279                  ;;*****************************************************************
```

```
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
014306  000004                  TST41:  SCOPE
014310  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
014316  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
014320  104003                          ERROR   3               ;BAD INIT ERROR

014322  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
014326  000123                          WRDATA                  ;WRDATA
014330  177400                          -400                    ;-400 WORDS
014332  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
014334     017                          .BYTE   17              ;SECTOR 17
014335     000                          .BYTE   0               ;TRACK 0
014336  000312                          312                     ;CYLINDER 312
014340  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
014344  000004                          4                       ;PATTERN 4
014346  000400                          400                     ;400 WORDS

014350  104417                          TLOADRK                 ;LOAD RK REGS
014352  104430                          TWAT96                  ;WAIT FOR INTERRUPT
014354  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

014356  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
014360  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

014362  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
014366  000121                          RDDATA                  ;RDDATA
014370  177675                          -103                    ;-103 WORDS
014372  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
014374     017                          .BYTE   17              ;SECTOR 17
014375     000                          .BYTE   0               ;TRACK 0
014376  000312                          312                     ;CYLINDER 312
014400  004437  040776                  JSR     R4,GENCOM
014404  002007                          2007                    ;CLEAR IBUFF
014406  000400                          400

014410  104417                          TLOADRK                 ;LOAD RK REGS
014412  104424                          TWAT32                  ;WAIT FOR INTERRUPT
014414  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
014416  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
014420  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

014422  012700  063012                  MOV     #OBUFF+206,R0   ;AFTER THE LAST 205 WORDS OF
014426  012701  000275                  MOV     #275,R1         ;THE OUTPUT BUFFER TO ALL ONES.
014432  012720  177777          1$:     MOV     #-1,(R0)+       ;THESE SHOULD ALL BE ONES IN
014436  005301                          DEC     R1              ;IBUFF BECAUSE THE PARTIAL
014440  001374                          BNE     1$              ;READ FILLED ONLY 103 WORDS.
014442  004437  040776                  JSR     R4,GENCOM       ;GO COMPARE IBUFF & OBUFF
014446  100000                          100000
```

```
5336  014450  000400                          400                     ;ALL 400 WORDS
5337  014452  000413                          BR      3$              ;NO ERRORS-EXIT
5338  014454  104015                          ERROR   15              ;REPORT FIRST COMPARE ERROR
5339
5340  014456  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5341  014462  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5342  014464  001406                          BEQ     65$             ;IF ZERO - EXIT
5343  014466  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5344  014472  040000                          40000
5345  014474  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5346  014476  104016                          ERROR   16              ;REPORT NEXT ERROR
5347  014500  000770                          BR      64$             ;LOOP
5348  014502                          65$:
5349
5350  014502                          3$:
5351                                  ;;*****************************************************************
5352                                  ;*TEST 42       WRITE DATA WITH NON-EXISTENT MEMORY
5353                                  ;*
5354                                  ;*      ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000.
5355                                  ;*      MAKE SURE NON-EXISTENT MEMORY SETS.
5356                                  ;*
5357                                  ;;*****************************************************************
5358  014502  000004                  TST42:  SCOPE
5359  014504  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5360  014512  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5361  014514  104003                          ERROR   3               ;BAD INIT ERROR
5362
5363  014516  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REG
5364  014522  001523                          BA16!BA17!WRDATA        ;BA16 & 17 SET WITH WRITE DATA
5365  014524  177777                          -1                      ;WORD COUNT OF 1
5366  014526  176000                          176000                  ;BUFF ADDRESS=IO PAGE BASE
5367  014530     013                          .BYTE   13              ;SECT 13
5368  014531     000                          .BYTE   0               ;TRACK 0
5369  014532  000312                          312                     ;CYLINDER 312
5370
5371  014534  104417                          TLOADRK                 ;LOAD RK REGS
5372  014536  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5373  014540  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5374  014542  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
5375  014544  000000                          0
5376  014546  000000                          0
5377  014550  000040                          NEMERR                  ;NON-EXISTENT MEMORY ERROR
5378  014552  104004                          ERROR   4 ;OR5,6,7      ;REPORT ANY DISCREPENCIES
5379  014554  012737  052762  001450          MOV     #EM11A,EM10N    ;SET UP ERROR MESSAGE
5380  014562  012737  047276  057662          MOV     #OPER42,DF010A  ;WITH SUPPORT MESSAGE
5381  014570  113700  001541                  MOVB    T.CS1+1,R0      ;GET UPPER CS1
5382  014574  042700  177774                  BIC     #177774,R0      ;CLEAR UNUSED BITS
5383  014600  022700  000003                  CMP     #3,R0           ;TEST IF BOTH UPPER BUS BITS SET
5384  014604  001406                          BEQ     1$              ;YES - SKIP
5385  014606  010037  001204                  MOV     R0,$REG11       ;SET UP FOR ERROR REPORT
5386  014612  012737  000003  001202          MOV     #3,$REG10
5387  014620  104010                          ERROR   10
5388  014622  022737  176002  001544  1$:     CMP     #176002,T.BA    ;TEST IF BUSS ADDRESS LOW OKAY
5389  014630  001412                          BEQ     2$              ;YES - SKIP
5390  014632  012737  052734  001450          MOV     #EM11,EM10N     ;SET UP MESSGAE
5391  014640  012737  176002  001202          MOV     #176002,$REG10  ;STORE VALUE FOR REPORT
```

```
014646  013737  001544  001204          MOV     T.BA,$REG11
014654  104010                          ERROR   10

014656  005737  001542          2$:     TST     T.WC            ;TEST IF WORD COUNT CORRECT
014662  001411                          BEQ     3$              ;YES - SKIP
014664  012737  052707  001450          MOV     #EM10,EM1ON     ;SET UP MESSAGE
014672  005037  001202                  CLR     $REG10
014676  013737  001542  001204          MOV     T.WC,$REG11
014704  104010                          ERROR 10

014706                          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.
                                ;*
                                ;*****************************************************************
014706  000004                  TST43:  SCOPE
014710  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
014716  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
014720  104003                          ERROR   3               ;BAD INIT ERROR

014722  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REG
014726  001521                          BA16!BA17!RDDATA        ;BA16 & 17 WITH READ DATA
014730  177777                          -1                      ;WORD COUNT OF 1
014732  176000                          176000                  ;BUFF ADDRESS=IO PAGE BASE
014734     013                          .BYTE   13              ;SECTOR 13
014735     000                          .BYTE   0               ;TRACK 0
014736  000312                          312                     ;CYL 312

014740  104417                          TLOADRK                 ;LOAD RK REGS
014742  104430                          TWAT96                  ;WAIT FOR INTERRUPT
014744  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
014746  104422                          TCHKWE                  ;CHECK OPERATION WITH ERRORS
014750  000000                          0
014752  000000                          0
014754  000040                          NEMERR                  ;NON-EXISTENT MEMORY ERROR
014756  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
014760  012737  052762  001450          MOV     #EM11A,EM1ON    ;SET MESSAGE
014766  012737  047352  057662          MOV     #OPER43,DF01DA  ;SET SUPPORT MESSAGE
014774  113700  001541                  MOVB    T.CS1+1,R0              ;GET UPPER CS1
015000  042700  177774                  BIC     #177774,R0      ;CLEAR UNWANTED BITS
015004  022700  000003                  CMP     #3,R0           ;TEST BOTH BUS 16 & 17 SET
015010  001406                          BEQ     1$              ;YES - SKIP
015012  012737  000003  001202          MOV     #3,$REG10       ;SET VALUES FOR REPORT
015020  010037  001204                  MOV     R0,$REG11
015024  104010                          ERROR   10

015026  022737  176002  001544  1$:     CMP     #176002,T.BA    ;TEST IF BUS ADDRESS CORRECT
015034  001412                          BEQ     2$              ;YES - SKIP
015036  012737  052734  001450          MOV     #EM11,EM1ON     ;SET MESSAGE
015044  012737  176002  001202          MOV     #176002,$REG10  ;SET VALUES FOR REPORT
015052  013737  001544  001204          MOV     T.BA,$REG11
015060  104010                          ERROR   10
```

```
5448
5449  015062  005737  001542          2$:     TST     T.WC            ;TEST IF WORD COUNT CORRECT
5450  015066  001411                          BEQ     3$              ;YES - SKIP
5451  015070  012737  052707  001450          MOV     #EM10,EM10N     ;SET MESSAGE
5452  015076  005037  001202                  CLR     $REG10          ;SET VALUES
5453  015102  013737  001542  001204          MOV     T.WC,$REG11
5454  015110  104010                          ERROR   10
5455
5456  015112                          3$:
5457
5458                                  ;;*****************************************************************
5459                                  ;*TEST 44       UNIBUS PARITY ERROR
5460                                  ;*
5461                                  ;*      INITIALIZE A MEMORY LOCATION WITH BAD
5462                                  ;*      PARITY. ISSUE A WRITE DATA OF 400 WORDS
5463                                  ;*      STARTING AT A LOCATION 110 WORDS BEFORE
5464                                  ;*      THE LOCATION WITH BAD PARITY.  MAKE SURE
5465                                  ;*      THAT UNIBUS PARITY ERROR SETS.
5466                                  ;*
5467                                  ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF
5468                                  ;*              MEMORY PARITY OPTION EXISTS FOR
5469                                  ;*              BUFFER.
5470                                  ;*
5471                                  ;;*****************************************************************
5472  015112  000004                  TST44:  SCOPE
5473  015114  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5474  015122  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5475  015124  104003                          ERROR   3               ;BAD INIT ERROR
5476
5477  015126  032737  000200  001656          BIT     #PARBK0,OPTFLG  ;TEST IF PARITY OPTION PRESENT
5478  015134  001013                          BNE     1$              ;YES-SKIP
5479  015136  032737  000004  001656          BIT     #MEMPYB,OPTFLG  ;TEST IF PARITY OPTION REPORTED
5480  015144  001005                          BNE     25$             ;NO-SKIP TO EXIT
5481  015146  104401  050257                  TYPE    ,OPR010         ;PRINT BYPASS MESSAGE
5482  015152  052737  000004  001656          BIS     #MEMPYB,OPTFLG  ;SET OPTION REPORTED BIT
5483  015160  000137  015572          25$:    JMP     2$              ;SKIP TO EXIT
5484
5485  015164                          1$:
5486  015164  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5487  015170  000123                          WRDATA                  ;WRDATA
5488  015172  177400                          -400                    ;-400 WORDS
5489  015174  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5490  015176     010                          .BYTE   10              ;SECTOR 10
5491  015177     000                          .BYTE   0               ;TRACK 0
5492  015200  000312                          312                     ;CYLINDER 312
5493
5494  015202  005077  164474                  CLR     @MMCSR1 ;CLEAR PARITY IE
5495  015206  032737  000100  001656          BIT     #PARBK1,OPTFLG  ;IS BANK 1 AVAIL
5496  015214  001402                          BEQ     3$              ;NO - SKIP
5497  015216  005077  164462                  CLR     @MMCSR2 ;CLEAR IE FOR BANK1
5498
5499  015222  004437  040776          3$:     JSR     R4,GENCOM       ;GENERATE DATA
5500  015226  000005                          5                       ;PATTERN 5
5501  015230  000400                          400                     ;400 WORDS
5502
5503  015232  012746  000340                  MOV     #PR7,-(SP)      ;PUT PRIORITY 7 ON STACK
```

```
5504  015236  012746  015244          MOV     #10$,-(SP)      ;PUT ADDRESS ON STACK
5505  015242  000002                  RTI                     ;SET PRI
5506  015244                  10$:
5507
5508  015244  032737  000100  001656  BIT     #PARBK1,OPTFLG  ;TEST IF PARITY BANK 1 AVAIL
5509  015252  001403                  BEQ     4$              ;NO - SKIP
5510  015254  012777  000004  164422  MOV     #BIT2,@MMCSR2   ;SET WRITE WRONG PARITY ENABLE
5511  015262  012777  000004  164412  4$:     MOV     #BIT2,@MMCSR1   ;SET WRITE WRONG PARITY BIT
5512  015270  012737  063032  063030          MOV     #OBUFF+226,OBUFF+224
5513  015276  012737  063030  063026          MOV     #OBUFF+224,OBUFF+222    ;WRITE ALL ONES INTO BUFFER W/BAD PARITY
5514  015304  012777  000001  164370          MOV     #BIT0,@MMCSR1   ;CLEAR CONTROL BIT, SET IE BIT
5515  015312  032737  000100  001656          BIT     #PARBK1,OPTFLG  ;TEST IF BANK 1 AVAIL
5516  015320  001403                  BEQ     5$              ;NO SKIP
5517  015322  012777  000001  164354          MOV     #BIT0,@MMCSR2   ;CLEAR CONTROL BIT, SET IE
5518
5519  015330  013746  001622          5$:     MOV     RKPRI,-(SP)     ;SET OLD PRIORITY
5520  015334  012746  015342                  MOV     #11$,-(SP)      ;SET ADDRESS
5521  015340  000002                  RTI                     ;RESTORE PRI
5522  015342                  11$:
5523
5524  015342  104417                  TLOADRK                 ;LOAD RK REGS
5525  015344  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5526  015346  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5527  015350  005077  164326          CLR     @MMCSR1 ;TURN OFF CS1
5528  015354  032737  000100  001656  BIT     #PARBK1,OPTFLG  ;TEST IF BANK 1 PARITY PRESENT
5529  015362  001402                  BEQ     6$              ;NO SKIP
5530  015364  005077  164314          CLR     @MMCSR2 ;TURN OFF BANK 1
5531  015370  005037  063030          6$:     CLR     OBUFF+224
5532  015374  005037  063026          CLR     OBUFF+222       ;CLEAR BAD PARITY ERROR
5533
5534  015400  004737  033704          JSR     PC,OPTTST       ;RESET OPTIONS
5535
5536  015404  104422                  TCHKWE                  ;CHECK OPERATION WITH ERROR
5537  015406  000000                  0
5538  015410  000000                  0
5539  015412  000100                  UPERR                   ;UNIBUS PARITY ERROR
5540  015414  104004                  ERROR   4; OR 5.6,7     ;REPORT ALL DISCREPANCIES
5541
5542  015416  012737  047425  057662          MOV     #OPER44,DFO10A  ;SET MESSAGES FOR REPORT
5543  015424  012737  052734  001450          MOV     #EM11,EM10N
5544  015432  023727  001544  063030          CMP     T.BA,#OBUFF+224 ;CHECK IF BA IN RANGE
5545  015440  103010                  BHIS    14$             ;NOT TO LOW - SKIP
5546  015442  012737  063032  001202          MOV     #OBUFF+226,$REG10 ;SET VALUES FOR REPORT
5547  015450  013737  001544  001204          MOV     T.BA,$REG11
5548  015456  104010                  ERROR   10
5549  015460  000413                  BR      16$
5550
5551  015462  023727  001544  063034  14$:    CMP     T.BA,#OBUFF+230 ;CHECK IF BA IN RANGE
5552  015470  101407                  BLOS    16$             ;YES - SKIP
5553  015472  012737  063034  001202          MOV     #OBUFF+230,$REG10 ;SET VALUES
5554  015500  012737  001544  001204          MOV     #T.BA,$REG11
5555  015506  104010                  ERROR   10
5556
5557  015510  012737  052707  001450  16$:    MOV     #EM10,EM10N     ;SET MESSAGE
5558  015516  023727  001542  177512          CMP     T.WC,#-266      ;CHECK IF WORD COUNT WITHIN RANGE
5559  015524  103007                  BHIS    20$             ;YES - SKIP
```

```
015526  012737  177512  001202          MOV     #-266,$REG10    ;SET VALUES
015534  013737  001542  001204          MOV     T.WC,$REG11
015542  104010                          ERROR   10

015544  023727  001542  177513  20$:    CMP     T.WC,#-265      ;STILL CHECKING IF WC IN RANGE
015552  101407                          BLOS    2$              ;YES - SKIP
015554  012737  177513  001202          MOV     #-265,$REG10    ;SET VALUES
015562  013737  001542  001204          MOV     T.WC,$REG11
015570  104010                          ERROR   10

015572                          2$:

                                .SBTTL  **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.
                                ;*
                                ;******************************************************************
015572  000004          TST45:  SCOPE
015574  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
015602  104416                          TSXINIT                 ;CLEAR SUBSYSTEM
015604  104003                          ERROR   3               ;BAD INIT ERROR

015606  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
015612  000123                          WRDATA                  ;WRDATA
015614  177000                          -1000                   ;-1000 WORDS
015616  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
015620     000                          .BYTE   0               ;SECTOR 0
015621     000                          .BYTE   0               ;TRACK 0
015622  000312                          312                     ;CYLINDER 312

015624  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
015630  000015                          15                      ;PATTERN 15
015632  001000                          1000                    ;1000 WORDS

015634  104417                          TLOADRK                 ;LOAD Rk REGS
015636  104430                          TWAT96                  ;WAIT FOR INTERRUPT
015640  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

015642  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
015644  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

015646  004437  040776                  JSR     R4,GENCOM       ;CLEAR IBUFF
015652  002007                          2007                    ;TO ALL 1'S
015654  001000                          1000

015656  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
015662  000121                          RDDATA                  ;RDDATA
015664  177400                          -400                    ;-400 WORDS
015666  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
015670     000                          .BYTE   0               ;SECTOR 0
015671     000                          .BYTE   0               ;TRACK 0
```

```
5616  015672  000312                        312                     ;CYLINDER 312
5617
5618  015674  104417                        TLOADRK                 ;LOAD RK REGS
5619  015676  104424                        TWAT32                  ;WAIT FOR INTERRUPT
5620  015700  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5621
5622  015702  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5623  015704  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5624
5625  015706  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
5626  015712  000121                        RDDATA                  ;RDDATA
5627  015714  177400                        -400                    ;-400 WORDS
5628  015716  061604                        IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5629  015720     001                        .BYTE   1               ;SECTOR 1
5630  015721     000                        .BYTE   0               ;TRACK 0
5631  015722  000312                        312                     ;CYLINDER 312
5632
5633  015724  104417                        TLOADRK                 ;LOAD RK REGS
5634  015726  104424                        TWAT32                  ;WAIT FOR INTERRUPT
5635  015730  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5636
5637  015732  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5638  015734  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5639
5640  015736  004437  040776                JSR     R4,GENCOM       ;COMPARE DATA
5641  015742  100000                        100000                  ;
5642  015744  001000                        1000                    ;1000 WORDS
5643  015746  000413                        BR      2$              ;NO MISCOMPARES-EXIT
5644  015750  104015                        ERROR   15              ;REPORT FIRST ERROR
5645
5646  015752  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
5647  015756  005300                6$:     DEC     R0              ;DECREMENT COUNT
5648  015760  001406                        BEQ     65$             ;IF ZERO - EXIT
5649  015762  004437  040776                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5650  015766  040000                        40000
5651  015770  000402                        BR      65$             ;NO MORE ERRORS - EXIT
5652  015772  104016                        ERROR   16              ;REPORT NEXT ERROR
5653  015774  000770                        BR      6$              ;LOOP
5654  015776                        65$:
5655
5656  015776                        2$:
5657                                ;;*****************************************************************
5658                                ;*TEST 46       TWO SECTOR WRITE DATA (PART 2)
5659                                ;*
5660                                ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5661                                ;*      TRACK 0, SECTOR 23.  READ DATA BACK ONE SECTOR
5662                                ;*      AT A TIME AND MAKE SURE A MID-TRANSFER
5663                                ;*      SEEK DID NOT TAKE PLACE.
5664                                ;*
5665                                ;;*****************************************************************
5666  015776  000004                TST46:  SCOPE
5667  016000  012737  000062  001262        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5668  016006  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
5669  016010  104003                        ERROR   3               ;BAD INIT ERROR
5670
5671  016012  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
5672  016016  000123                  WRDATA                  ;WRDATA
5673  016020  177000                  -1000                   ;-1000 WORDS
5674  016022  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5675  016024     023                  .BYTE   23              ;SECTOR 23
5676  016025     000                  .BYTE   0               ;*RACK 0
5677  016026  000312                  312                     ;CYLINDER 312
5678
5679  016030  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA
5680  016034  000016                  16                      ;PATTERN 16
5681  016036  001000                  1000                    ;1000 WORDS
5682
5683  016040  104417                  TLOADRK                 ;LOAD RK REGS
5684  016042  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5685  016044  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5686
5687  016046  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5688  016050  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5689
5690                          ;       IF THE TRACK ADDRESS AT THE END OF OPERATION IS IN ERROR
5691                          ;       THE CONTROLLER DID A MID-TRANSFER SEEK AS THOUGH IT
5692                          ;       WERE IN 24(8) SECTORS PER TRACK MODE.
5693
5694  016052  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5695  016056  000121                  RDDATA                  ;RDDATA
5696  016060  177400                  -400                    ;-400 WORDS
5697  016062  060604                  IBUFF                   ;IBUFF IS BUFF ADDRESS
5698  016064     023                  .BYTE   23              ;SECTOR 23
5699  016065     000                  .BYTE   0               ;TRACK 0
5700  016066  000312                  312                     ;CYLINDER 312
5701
5702  016070  004437  040776          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
5703  016074  002007                  2007
5704  016076  001000                  1000
5705
5706  016100  104417                  TLOADRK                 ;LOAD RK REGS
5707  016102  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5708  016104  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5709
5710  016106  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5711  016110  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5712
5713  016112  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5714  016116  000121                  RDDATA                  ;RDDATA
5715  016120  177400                  -400                    ;-400 WORDS
5716  016122  061604                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5717  016124     024                  .BYTE   24              ;SECTOR 24
5718  016125     000                  .BYTE   0               ;TRACK 0
5719  016126  000312                  312                     ;CYLINDER 312
5720
5721  016130  104417                  TLOADRK                 ;LOAD RK REGS
5722  016132  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5723  016134  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5724
5725  016136  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5726  016140  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5727
```

```
5728  016142  004437  040776          JSR     R4,GENCOM       ;COMPARE DATA
5729  016146  100000                  100000
5730  016150  001000                  1000                    ;1000 WORDS
5731  016152  000413                  BR      1$              ;NO ERRORS-SKIP
5732  016154  104015                  ERROR   15              ;REPORT FIRST ERROR
5733
5734  016156  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5735  016162  005300          64$:    DEC     R0              ;DECREMENT COUNT
5736  016164  001406                  BEQ     65$             ;IF ZERO - EXIT
5737  016166  004437  040776          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5738  016172  040000                  40000
5739  016174  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5740  016176  104016                  ERROR   16              ;REPORT NEXT ERROR
5741  016200  000770                  BR      64$             ;LOOP
5742  016202                  65$:
5743  016202                  1$:
5744                          ;;*****************************************************************
5745                          ;*TEST 47        TWO SECTOR WRITE DATA (PART 3)
5746                          ;*
5747                          ;*      ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312,
5748                          ;*      TRACK 0, SECTOR 10.  READ DATA BACK ONE SECTOR AT
5749                          ;*      A TIME AND CHECK ZERO FILL OF SECOND SECTOR.
5750                          ;*
5751                          ;;*****************************************************************
5752  016202  000004          T$T47:  SCOPE
5753  016204  012737  000062  001262  MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5754  016212  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
5755  016214  104003                  ERROR   3               ;BAD INIT ERROR
5756
5757  016216  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5758  016222  000123                  WRDATA                  ;WRDATA
5759  016224  177377                  -401                    ;-401 WORDS
5760  016226  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5761  016230     010                  .BYTE   10              ;SECTOR 10
5762  016231     000                  .BYTE   0               ;TRACK 0
5763  016232  000312                  312                     ;CYLINDER 312
5764
5765  016234  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA
5766  016240  000002                  2                       ;PATTERN 2
5767  016242  001000                  1000                    ;1000 WORDS
5768
5769  016244  104417                  TLCADRK                 ;LOAD RK REGS
5770  016246  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5771  016250  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5772
5773  016252  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5774  016254  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5775                          ;       CLEAR LAST 377 WORDS OF OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
5776  016256  012700  063606          MOV     #OBUFF+1002,R0  ;GET STARTING ADDRESS TO BE CLEARED
5777  016262  012701  000377          MOV     #377,R1         ;NUMBER OF WORDS
5778  016266  005020          1$:     CLR     (R0)+           ;CLEAR WORD
5779  016270  005301                  DEC     R1              ;DEC COUNT
5780  016272  001375                  BNE     1$              ;LOOP UNTIL COUNT ZERO
5781  016274  004437  040776          JSR     R4,GENCOM       ;CLEAR IBUFF TO ONES
5782  016300  002007                  2007
5783  016302  001000                  1000
```

```
5784
5785  016304  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5786  016310  000121                  RDDATA                  ;RDDATA
5787  016312  177400                  -400                    ;-400 WORDS
5788  016314  060604                  IBUFF                   ;IBUFF IS BUFF ADDRESS
5789  016316     010                  .BYTE   10              ;SECTOR 10
5790  016317     000                  .BYTE   0               ;TRACK 0
5791  016320  000312                  312                     ;CYLINDER 312
5792
5793  016322  104417                  TLOADRK                 ;LOAD RK REGS
5794  016324  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5795  016326  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5796
5797  016330  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5798  016332  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5799  016334  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5800  016340  000121                  RDDATA                  ;RDDATA
5801  016342  177400                  -400                    ;-400 WORDS
5802  016344  061604                  IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5803  016346     011                  .BYTE   11              ;SECTOR 11
5804  016347     000                  .BYTE   0               ;TRACK 0
5805  016350  000312                  312                     ;CYLINDER 312
5806
5807  016352  104417                  TLOADRK                 ;LOAD RK REGS
5808  016354  104424                  TWAT32                  ;WAIT FOR INTERRUPT
5809  016356  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5810
5811  016360  004437  040776          JSR     R4,GENCOM       ;DATA COMPARE
5812  016364  100000                  100000
5813  016366  001000                  1000                    ;1000 WORDS
5814  016370  000413                  BR      2$              ;NO ERROR-SKIP
5815  016372  104015                  ERROR   15              ;REPORT FIRST ERROR
5816
5817  016374  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5818  016400  005300          64$:    DEC     R0              ;DECREMENT COUNT
5819  016402  001406                  BEQ     65$             ;IF ZERO - EXIT
5820  016404  004437  040776          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5821  016410  040000                  40000
5822  016412  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5823  016414  104016                  ERROR   16              ;REPORT NEXT ERROR
5824  016416  000770                  BR      64$             ;LOOP
5825  016420                  65$:
5826
5827  016420                  2$:
5828                          ;;*****************************************************************
5829                          ;*TEST 50       MID-TRANSFER SEEK ON WRITE (PART 1)
5830                          ;*
5831                          ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5832                          ;*      TRACK 0, SECTOR 25.  READ DATA BACK ONE SECTOR
5833                          ;*      AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5834                          ;*      DID TAKE PLACE.
5835                          ;*
5836                          ;;*****************************************************************
5837  016420  000004          TST50:  SCOPE
5838  016422  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5839  016430  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
```

```
5840  016432  104003                        ERROR   3               ;BAD INIT ERROR
5841
5842  016434  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
5843  016440  000123                        WRDATA                  ;WRDATA
5844  016442  177000                        -1000                   ;-1000 WORDS
5845  016444  062604                        OBUFF                   ;OBUFF IS BUFF ADDRESS
5846  016446     025                        .BYTE   25              ;SECTOR 25
5847  016447     000                        .BYTE   0               ;TRACK 0
5848  016450  000312                        312                     ;CYLINDER 312
5849
5850  016452  004437  040776                JSR     R4,GENCOM       ;GENERATE DATA
5851  016456  000003                        3                       ;PATTERN 3
5852  016460  001000                        1000                    ;1000 WORDS
5853
5854  016462  104417                        TLOADRK                 ;LOAD RK REGS
5855  016464  104430                        TWAT96                  ;WAIT FOR INTERRUPT
5856  016466  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5857
5858  016470  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5859  016472  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5860                          ;             A TRACK ERROR PRINTED OUT AT THE END OF THE OPERATION INDICATES A
5861                          ;             MID-TRANSFER HEAD SWITCH DID NOT OCCUR.
5862  016474  004437  040776                JSR     R4,GENCOM
5863  016500  002007                        2007
5864  016502  001000                        1000
5865
5866  016504  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
5867  016510  000121                        RDDATA                  ;RDDATA
5868  016512  177400                        -400                    ;-400 WORDS
5869  016514  060604                        IBUFF                   ;IBUFF IS BUFF ADDRESS
5870  016516     025                        .BYTE   25              ;SECTOR 25
5871  016517     000                        .BYTE   0               ;TRACK 0
5872  016520  000312                        312                     ;CYLINDER 312
5873
5874  016522  104417                        TLOADRK                 ;LOAD RK REGS
5875  016524  104425                        TWAT48                  ;WAIT FOR INTERRUPT
5876  016526  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5877
5878  016530  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5879  016532  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5880
5881  016534  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
5882  016540  000121                        RDDATA                  ;RDDATA
5883  016542  177400                        -400                    ;-400 WORDS
5884  016544  061604                        IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5885  016546     000                        .BYTE   0               ;SECTOR 0
5886  016547     001                        .BYTE   1               ;TRACK 1
5887  016550  000312                        312                     ;CYLINDER 312
5888
5889  016552  104417                        TLOADRK                 ;LOAD RK REGS
5890  016554  104425                        TWAT48                  ;WAIT FOR INTERRUPT
5891  016556  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5892
5893  016560  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5894  016562  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5895
```

```
5896  016564  004437  040776          JSR     R4,GENCOM       ;COMPARE DATA
5897  016570  100000                  100000
5898  016572  001000                  1000                    ;1000 WORDS
5899  016574  000413                  BR      1$              ;NO ERRORS-SKIP
5900  016576  104015                  ERROR   15              ;REPORT FIRST ERROR
5901
5902  016600  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
5903  016604  005300          64$:    DEC     R0              ;DECREMENT COUNT
5904  016606  001406                  BEQ     65$             ;IF ZERO - EXIT
5905  016610  004437  040776          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5906  016614  040000                  40000
5907  016616  000402                  BR      65$             ;NO MORE ERRORS - EXIT
5908  016620  104016                  ERROR   16              ;REPORT NEXT ERROR
5909  016622  000770                  BR      64$             ;LOOP
5910  016624                  65$:
5911
5912  016624                  1$:
5913                          ;;*****************************************************************
5914                          ;*TEST 51        MID-TRANSFER SEEK ON WRITE (PART 2)
5915                          ;*
5916                          ;*       ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5917                          ;*       TRACK 2, SECTOR 25.  READ DATA BACK ONE SECTOR
5918                          ;*       AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5919                          ;*       DID TAKE PLACE.
5920                          ;*
5921                          ;;*****************************************************************
5922  016624  000004          TST51:  SCOPE
5923  016626  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
5924  016634  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
5925  016636  104003                  ERROR   3               ;BAD INIT ERROR
5926
5927  016640  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
5928  016644  000123                  WRDATA                  ;WRDATA
5929  016646  177000                  -1000                   ;-1000 WORDS
5930  016650  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
5931  016652     025                  .BYTE   25              ;SECTOR 25
5932  016653     002                  .BYTE   2               ;TRACK 2
5933  016654  000312                  312                     ;CYLINDER 312
5934
5935  016656  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA
5936  016662  000004                  4                       ;PATTERN 4
5937  016664  001000                  1000                    ;1000 WORDS
5938
5939  016666  104417                  TLOADRK                 ;LOAD RK REGS
5940  016670  104430                  TWAT96                  ;WAIT FOR INTERRUPT
5941  016672  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5942                          ;       A CYLINDER ERROR REPORTED AT THE END OF THE OPERATION INDICATES A
5943                          ;       MID-TRANSFER SEEK DID NOT OCCUR.
5944  016674  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5945  016676  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5946
5947  016700  004437  040776          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
5948  016704  002007                  2007
5949  016706  001000                  1000
5950
5951  016710  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
5952  016714  000121                    RDDATA                  ;RDDATA
5953  016716  177400                    -400                    ;-400 WORDS
5954  016720  060604                    IBUFF                   ;IBUFF IS BUFF ADDRESS
5955  016722     025                    .BYTE   25              ;SECTOR 25
5956  016723     002                    .BYTE   2               ;TRACK 2
5957  016724  000312                    312                     ;CYLINDER 312
5958
5959  016726  104417                    TLOADRK                 ;LOAD RK REGS
5960  016730  104425                    TWAT48                  ;WAIT FOR INTERRUPT
5961  016732  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5962
5963  016734  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5964  016736  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5965
5966  016740  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
5967  016744  000121                    RDDATA                  ;RDDATA
5968  016746  177400                    -400                    ;-400 WORDS
5969  016750  061604                    IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5970  016752     000                    .BYTE   0               ;SECTOR 0
5971  016753     000                    .BYTE   0               ;TRACK 0
5972  016754  000313                    313                     ;CYLINDER 313
5973
5974  016756  104417                    TLOADRK                 ;LOAD RK REGS
5975  016760  104425                    TWAT48                  ;WAIT FOR INTERRUPT
5976  016762  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5977
5978  016764  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5979  016766  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5980
5981  016770  004437  040776            JSR     R4,GENCOM       ;COMPARE DATA
5982  016774  100000                    100000
5983  016776  001000                    1000                    ;1000 WORDS
5984  017000  000413                    BR      1$              ;NO MISCOMPARES-SKIP
5985  017002  104015                    ERROR   15              ;REPORT 1ST ERROR
5986
5987  017004  013700  001634            MOV     ERRLMT,R0       ;GET ERROR LIMIT
5988  017010  005300              64$:  DEC     R0              ;DECREMENT COUNT
5989  017012  001406                    BEQ     65$             ;IF ZERO - EXIT
5990  017014  004437  040776            JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5991  017020  040000                    40000
5992  017022  000402                    BR      65$             ;NO MORE ERRORS - EXIT
5993  017024  104016                    ERROR   16              ;REPORT NEXT ERROR
5994  017026  000770                    BR      64$             ;LOOP
5995  017030                      65$:
5996
5997  017030                      1$:
5998                              ;;*****************************************************************
5999                              ;*TEST 52        TWO SECTOR READ DATA (PART 1)
6000                              ;*
6001                              ;*       ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312.
6002                              ;*       TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS
6003                              ;*       READ.
6004                              ;*
6005                              ;*       NOTE:   TWO SECTOR WRITE DATA (PART 1) MUST BE
6006                              ;*               EXECUTED BEFORE THIS TEST.
6007                              ;*
```

```
                                ;;*****************************************************************
017030  000004                  TST52:  SCOPE
017032  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
017040  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
017042  104003                          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
017044  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
017050  000015                          15                      ;PATTERN 15
017052  001000                          1000                    ;1000 WORDS

017054  004437  040776                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
017060  002007                          2007
017062  001000                          1000

017064  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
017070  000121                          RDDATA                  ;RDDATA
017072  177000                          -1000                   ;-1000 WORDS
017074  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
017076     000                          .BYTE   0               ;SECTOR 0
017077     000                          .BYTE   0               ;TRACK 0
017100  000312                          312                     ;CYLINDER 312

017102  104417                          TLOADRK                 ;LOAD RK REGS
017104  104430                          TWAT96                  ;WAIT FOR INTERRUPT
017106  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

017110  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017112  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

017114  004437  040776                  JSR     R4,GENCOM       ;COMPARE DATA
017120  100000                          100000
017122  001000                          1000                    ;1000 WORDS
017124  000413                          BR      1$              ;NO MISCOMPARES-SKIP
017126  104015                          ERROR   15

017130  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
017134  005300                  64$:    DEC     R0              ;DECREMENT COUNT
017136  001406                          BEQ     65$             ;IF ZERO - EXIT
017140  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
017144  040000                          40000
017146  000402                          BR      65$             ;NO MORE ERRORS - EXIT
017150  104016                          ERROR   16              ;REPORT NEXT ERROR
017152  000770                          BR      64$             ;LOOP
017154                          65$:

017154                          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.
                                ;*
                                ;;*****************************************************************
017154  000004                  TST53:  SCOPE
017156  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
017164  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
017166  104003                          ERROR   3               ;BAD INIT ERROR

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

017170  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
017174  000016                          16                      ;PATTERN 16
017176  001000                          1000                    ;1000 WORDS

017200  004437  040776                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
017204  002007                          2007
017206  001000                          1000
017210  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
017214  000121                          RDDATA                  ;RDDATA
017216  177000                          -1000                   ;-1000 WORDS
017220  063604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
017222     023                          .BYTE   23              ;SECTOR 23
017223     000                          .BYTE   0               ;TRACK 0
017224  000312                          312                     ;CYLINDER 312

017226  104417                          TLOADR%                 ;LOAD RK REGS
017230  104430                          TWAT96                  ;WAIT FOR INTERRUPT
017232  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

017234  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017236  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

017240  004437  040776                  JSR     R4,GENCOM       ;COMPARE DATA
017244  100000                          100000
017246  001000                          1000                    ;1000 WORDS
017250  000413                          BR      1$              ;NO MISCOMPARES-SKIP
017252  104015                          ERROR   15              ;REPORT 1ST ERROR

017254  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
017260  005300                  64$:    DEC     R0              ;DECREMENT COUNT
017262  001406                          BEQ     65$             ;IF ZERO - EXIT
017264  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
017270  040000                          40000
017272  000402                          BR      65$             ;NO MORE ERRORS - EXIT
017274  104016                          ERROR   16              ;REPORT NEXT ERROR
017276  000770                          BR      64$             ;LOOP
017300                          65$:

C17300                          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.
                                        ;*
                                        ;;*****************************************************************
017300  000004                  TST54:  SCOPE
017302  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
017310  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
017312  104003                          ERROR   3               ;BAD INIT ERROR

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

017314  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
017320  000002                          2                       ;PATTERN 2
017322  000401                          401                     ;401 WORDS

017324  004437  040776                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
017330  002007                          2007
017332  001000                          1000

017334  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
017340  000121                          RDDATA                  ;RDDATA
017342  177377                          -401                    ;-401 WORDS
017344  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
017346     010                          .BYTE   10              ;SECTOR 10
017347     000                          .BYTE   0               ;TRACK 0
017350  000312                          312                     ;CYLINDER 312

017352  104417                          TLOADRK                 ;LOAD RK REGS
017354  104430                          TWAT96                  ;WAIT FOR INTERRUPT
017356  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

017360  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017362  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

017364  004437  040776                  JSR     R4,GENCOM       ;COMPARE DATA
017370  100000                          100000
017372  000401                          401                     ;401 WORDS
017374  000413                          BR      1$              ;NO MISCOMPARES-SKIP
017376  104015                          ERROR   15              ;PRINT FIRST ERROR

017400  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
017404  005300                  64$:    DEC     R0              ;DECREMENT COUNT
017406  001406                          BEQ     65$             ;IF ZERO - EXIT
017410  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
017414  040000                          40000
017416  000402                          BR      65$             ;NO MORE ERRORS - EXIT
017420  104016                          ERROR   16              ;REPORT NEXT ERROR
017422  000770                          BR      64$             ;LOOP
017424                          65$:
017424                          1$:
                                        ;;*****************************************************************
                                        ;*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
```

```
 6176                                   ;*      READ AND A MID-TRANSFER SEEK DOES OCCUR.
 6177                                   ;*
 6178                                   ;*      NOTE:   MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE
 6179                                   ;*              EXECUTED BEFORE THIS TEST.
 6180                                   ;*
 6181                                   ;;*****************************************************************
 6182  017424  000004                   TST55:  SCOPE
 6183  017426  012737  000062  001252           MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
 6184  017434  104416                           TSSINIT                 ;CLEAR SUBSYSTEM
 6185  017436  104003                           ERROR   3               ;BAD INIT ERROR
 6186
 6187                                   ;       GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
 6188  017440  004437  040776                   JSR     R4,GENCOM       ;GENERATE DATA
 6189  017444  000003                           3                       ;PATTERN 3
 6190  017446  001000                           1000                    ;1000 WORDS
 6191
 6192  017450  004437  040776                   JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
 6193  017454  002007                           2007
 6194  017456  001000                           1000
 6195
 6196  017460  004437  034574                   JSR     R4,LRLOAD       ;LOAD "L" REGS
 6197  017464  000121                           RDDATA                  ;RDDATA
 6198  017466  177000                           -1000                   ;-1000 WORDS
 6199  017470  060604                           IBUFF                   ;IBUFF IS BUFF ADDRESS
 6200  017472     025                           .BYTE   25              ;SECTOR 25
 6201  017473     000                           .BYTE   0               ;TRACK 0
 6202  017474  000312                           312                     ;CYLINDER 312
 6203
 6204  017476  104417                           TLOADRK                 ;LOAD RK REGS
 6205  017500  104430                           TWAT96                  ;WAIT FOR INTERRUPT
 6206  017502  104002                           ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6207
 6208  017504  104421                           TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 6209  017506  104004                           ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 6210
 6211  017510  004437  040776                   JSR     R4,GENCOM       ;COMPARE DATA
 6212  017514  100000                           100000
 6213  017516  001000                           1000                    ;1000 WORDS
 6214  017520  000413                           BR      1$              ;NO MISCOMPARES-SKIP
 6215  017522  104015                           ERROR   15              ;PRINT FIRST ERROR
 6216
 6217  017524  013700  001634                   MOV     ERRLMT,R0       ;GET ERROR LIMIT
 6218  017530  005300                   64$:    DEC     R0              ;DECREMENT COUNT
 6219  017532  001406                           BEQ     65$             ;IF ZERO - EXIT
 6220  017534  004437  040776                   JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
 6221  017540  040000                           40000
 6222  017542  000402                           BR      65$             ;NO MORE ERRORS - EXIT
 6223  017544  104016                           ERROR   16              ;REPORT NEXT ERROR
 6224  017546  000770                           BR      64$             ;LOOP
 6225  017550                           65$:
 6226  017550                           1$:
 6227                                   ;;*****************************************************************
 6228                                   ;*TEST 56       MID-TRANSFER SEEK ON READ (PART 2)
 6229                                   ;*
 6230                                   ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
 6231                                   ;*      TRACK 2, SECTOR 25.  VERIFY THAT CORRECT DATA IS
```

```
6232                                    ;*      READ AND A MID-TRANSFER SEEK DOES OCCUR.
6233                                    ;*
6234                                    ;*      NOTE:   MID-TRANSFER SEEK ON WRITE (PART 2) MJST BE
6235                                    ;*              EXECUTED BEFORE THIS TEST.
6236                                    ;*
6237                                    ;;*****************************************************************
6238  017550  000004                    TST56:  SCOPE
6239  017552  012737  000062  001262            MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
6240  017560  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
6241  017562  104003                            ERROR   3               ;BAD INIT ERROR
6242
6243                                    ;       GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)
6244  017564  004437  040776                    JSR     R4,GENCOM       ;GENERATE DATA
6245  017570  000004                    4                               ;PATTERN 4
6246  017572  001000                    1000                            ;1000 WORDS
6247
6248  017574  004437  040776                    JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6249  017600  002007                    2007
6250  017602  001000                    1000
6251
6252  017604  004437  034574                    JSR     R4,LRLOAD       ;LOAD "L" REGS
6253  017610  000121                    RDDATA                          ;RDDATA
6254  017612  177000                    -1000                           ;-1000 WORDS
6255  017614  060604                    IBUFF                           ;IBUFF IS BUFF ADDRESS
6256  017616     025                    .BYTE   25                      ;SECTOR 25
6257  017617     002                    .BYTE   2                       ;TRACK 2
6258  017620  000312                    312                             ;CYLINDER 312
6259
6260  017622  104417                    TLOADRK                         ;LOAD RK REGS
6261  017624  104430                    TWAT96                          ;WAIT FOR INTERRUPT
6262  017626  104002                    ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR
6263
6264  017630  104421                    TCHKOP                          ;CHECK OPERATION FOR ANY ERRORS
6265  017632  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6266
6267  017634  004437  040776                    JSR     R4,GENCOM       ;COMPARE DATA
6268  017640  100000                    100000
6269  017642  001000                    1000                            ;1000 WORDS
6270  017644  000413                    BR      1$                      ;NO MISCOMPARES-SKIP
6271  017646  104015                    ERROR   15                      ;REPORT FIRST ERROR
6272
6273  017650  013700  001634                    MOV     ERRLMT,R0       ;GET ERROR LIMIT
6274  017654  005300                    64$:    DEC     R0              ;DECREMENT COUNT
6275  017656  001406                    BEQ     65$                     ;IF ZERO - EXIT
6276  017660  004437  040776                    JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6277  017664  040000                    40000
6278  017666  000402                    BR      65$                     ;NO MORE ERRORS - EXIT
6279  017670  104016                    ERROR   16                      ;REPORT NEXT ERROR
6280  017672  000770                    BR      64$                     ;LOOP
6281  017674                            65$:
6282  017674                            1$:
6283                                    ;;*****************************************************************
6284                                    ;*TEST 57       CYLINDER ADDRESS OVERFLOW (PART 1)
6285                                    ;*
6286                                    ;*      ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
6287                                    ;*      TRACK 2, SECTOR 25.  MAKE SURE CYLINDER ADDRESS
```

```
                                        ;*      OVERFLOW ERROR DOES NOT OCCUR.
                                        ;*
                                        ;;*****************************************************************
017674  000004                    TST57:  SCOPE
017676  012737  000062  001262            MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
017704  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
017706  104003                            ERROR   3               ;BAD INIT ERROR
                                
017710  004437  034574                    JSR     R4,LRLOAD       ;LOAD "L" REGS
017714  000121                            RDDATA                  ;RDDATA
017716  177400                            -400                    ;-400 WORDS
017720  060604                            IBUFF                   ;IBUFF IS BUFF ADDRESS
017722     025                            .BYTE   25              ;SECTOR 25
017723     002                            .BYTE   2               ;TRACK 2
017724  000632                            632                     ;CYLINDER 632
                                
017726  104417                            TLOADRK                 ;LOAD RK REGS
017730  104434                            TWAIT59                 ;WAIT FOR INTERRUPT
017732  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                
017734  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017736  104004                            ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
                                        ;;*****************************************************************
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
017740  000004                    TST60:  SCOPE
017742  012737  000062  001262            MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
017750  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
017752  104003                            ERROR   3               ;BAD INIT ERROR
                                
017754  004437  034574                    JSR     R4,LRLOAD       ;LOAD "L" REGS
017760  000121                            RDDATA                  ;RDDATA
017762  177377                            -401                    ;-401 WORDS
017764  060604                            IBUFF                   ;IBUFF IS BUFF ADDRESS
017766     025                            .BYTE   25              ;SECTOR 25
017767     002                            .BYTE   2               ;TRACK 2
017770  000632                            632                     ;CYLINDER 632
                                
017772  104417                            TLOADRK                 ;LOAD RK REGS
017774  104434                            TWAIT59                 ;WAIT FOR INTERRUPT
017776  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                
020000  104422                            TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
020002  001000                            COERR                   ;CYLINDER OVERFLOW
020004  000000                            0
020006  000000                            0
020010  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.
                                        ;*
                                        ;*****************************************************************
020012  000004                          T61:    SCOPE
020014  012737  000062  001262                  MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
020022  012737  000312  001664                  MOV     #312,REFMT      ;SET REFORMAT SWITCH
020030  012737  020040  001110                  MOV     #1$,$LPERR      ;SET LOCAL LOOP ON ERROR
020036  005001                                  CLR     R1              ;CLEAR R1 FOR TRACK COUNTER
020040                                  1$:
020040  104416                                  TSSINIT                 ;CLEAR SUBSYSTEM
020042  104003                                  ERROR   3               ;BAD INIT ERROR
020044  012737  010127  001600                  MOV     #WRHEAD!CFMT,L.CS1  ;SET UP FOR WRITE HEADER 24(8. SECTOR MODE
020052  013737  001626  001610                  MOV     DRVNUM,L.CS2    ;SET DRIVE NUMBER
020060  012737  000074  001602                  MOV     #74,L.WC        ;SET WORD COUNT
020066  110137  001607                          MOVB    R1,L.DT         ;LOAD TRACK ADDRESS
020072  012737  062604  001604                  MOV     #OBUFF,L.BA     ;       SET BUS ADDRESS
020100  012737  000312  001614                  MOV     #312,L.DCYL     ;     CYLINDER ADDRESS
                                        
020106  004437  040776                          JSR     R4,GENCOM       ;GENERATE HEADER
020112  001200                                  1200                    ;INCLUDE BAD SECTOR BITS
                                        
020114  104417                                  T_OADRK                 ;_OAD RK REGS
020116  104434                                  TWAT159                 ;WAIT FOR INTERRUPT
020120  104002                                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                        
020122  104421                                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
020124  104004                                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
                                        
020126  104415                                  SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
                                        
020130  005701                                  TST     R1              ;R1 POINTING TO TRACK 0
020132  001002                                  BNE     2$              ;NO-SKIP
020134  005201                                  INC     R1              ;BUMP TO TRACK 1
020136  000740                                  BR      1$              ;LOOP
020140  012737  020150  001110          2$:     MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
020146  005001                                  CLR     R1              ;CLEAR TRACK POINTER
020150                                  3$:
020150  104416                                  TSSINIT                 ;CLEAR SUBSYSTEM
020152  104003                                  ERROR   3               ;BAD INIT ERROR
020154  012737  010125  001600                  MOV     #RDHEAD!CFMT,L.CS1  ;LOAD READ 24(8) SECTOR FORMAT
020162  013737  001626  001610                  MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
020170  110137  001607                          MOVB    R1,L.DT         ;LOAD TRACK
020174  012737  000312  001614                  MOV     #312,L.DCYL     ;LOAD CYLINDER
                                        
020202  004437  035114                          JSR     R4,RDSTHD       ;GO READ STANDARD HEADER
020206  104421                                  TCHKOP                  ;RETURN IF CERR W/O DATA LATE SET
020210  104004                                  ERROR   4; OR 5,6,7     ;REPORT ALL OTHER ERRORS
020212  104013                                  ERROR   13              ;REPORT DATA LATE
020214  104002                                  ERROR   2               ;REPORT "OPERATION TO SLOW" OR "HEADER
                                                                        ;O NOT FOUND" MESSAGE
```

```
                                        SCOP1                   ;LOCAL LOOP TO 3$ ON ERROR
020216  104415                          JSR     R4,GENCOM       ;GENERATE & COMPARE HEADERS
020220  004437  040776                  101200                  ;INCLUDING BAD SECTOR LISTS
020224  101200                          BR      4$              ;NO MISCOMPARES-SKIP
020226  000413                          ERROR   15              ;REPORT FIRST MISCOMPARE
020230  104015
                                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
020232  013700  001634          64$:    DEC     R0              ;DECREMENT COUNT
020236  005300                          BEQ     65$             ;IF ZERO - EXIT
020240  001406                          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
020242  004437  040776                  40000
020246  040000                          BR      65$             ;NO MORE ERRORS - EXIT
020250  000402                          ERROR   16              ;REPORT NEXT ERROR
020252  104016                          BR      64$             ;LOOP
020254  000770
020256                          65$:

020256  104415          4$:     SCOP1                   ;LOCAL LOOP TO 3$
020260  005701                  TST     R1              ;POINTING TO TRACK 1
020262  001002                  BNE     5$              ;NO-EXIT
020264  005201                  INC     R1              ;BUMP TO TRACK 1
020266  000730                  BR      3$              ;LOOP

020270                  5$:
                        ;;*****************************************************************
                        ;*TEST 62       24 SECTOR FORMAT DATA TRANSFER (PART 1)
                        ;*
                        ;*      ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
                        ;*      TO CYLINDER 312, TRACK 0, SECTOR 0.  READ SECTOR BACK
                        ;*      AND MAKE SURE IT IS CORRECT.
                        ;*
                        ;*****************************************************************
020270  000004  TST62:   SCOPE
020272  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
020300  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
020306  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
020310  104003                          ERROR   3               ;BAD INIT ERROR

020312  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
020316  000013                          13                      ;PATTERN 13
020320  000400                          400                     ;400 WORDS

020322  004437  040776                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
020326  002007                          2007
020330  000400                          400

020332  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
020336  010123                          WRDATA!CFMT                     ;WRDATA!CFMT
020340  177400                          -400                    ;-400 WORDS
020342  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
020344     000                          .BYTE   0               ;SECTOR 0
020345     000                          .BYTE   0               ;TRACK 0
020346  000312                          312                     ;CYLINDER 312

020350  104417                          TLOADRK                 ;LOAD RK REGS
020352  104430                          TWAT96                  ;WAIT FOR INTERRUPT
```

```
6456  020354  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6457
6458  020356  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6459  020360  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6460
6461  020362  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6462  020366  010121                          RDDATA!CFMT                     ;RDDATA!CFMT
6463  020370  177400                          -400                    ;-400 WORDS
6464  020372  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
6465  020374      000                         .BYTE   0               ;SECTOR 0
6466  020375      000                         .BYTE   0               ;TRACK 0
6467  020376  000312                          312                     ;CYLINDER 312
6468
6469  020400  104417                          TLOADRK                 ;LOAD RK REGS
6470  020402  104424                          TWAIT32                 ;WAIT FOR INTERRUPT
6471  020404  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6472
6473  020406  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6474  020410  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6475
6476  020412  004437  040776                  JSR     R4,GENCOM       ;COMPARE DATA
6477  020416  100000                          100000
6478  020420  000400                          400                     ;400 WORDS
6479  020422  000413                          BR      1$              ;NO MISCOMPARES-SKIP
6480  020424  104015                          ERROR   15              ;REPORT 1ST ERROR
6481
6482  020426  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6483  020432  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6484  020434  001406                          BEQ     65$             ;IF ZERO - EXIT
6485  020436  004437  040776                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6486  020442  040000                          40000
6487  020444  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6488  020446  104016                          ERROR   16              ;REPORT NEXT ERROR
6489  020450  000770                          BR      64$             ;LOOP
6490  020452                          65$:
6491
6492  020452                          1$:
6493                                  ;;*****************************************************************
6494                                  ;*TEST 63       24 SECTOR FORMAT DATA TRANSFER (PART 2)
6495                                  ;*
6496                                  ;*      LOAD A LOCATION WITH BAD PARITY.  ISSUE A WRITE DATA OF
6497                                  ;*      400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0,
6498                                  ;*      SECTOR 0 WITH BUFFER BEGINNING 110 WORDS BEFORE WORD
6499                                  ;*      WITH BAD PARITY.  MAKE SURE UNIBUS PARITY ERROR DOES NOT SET.
6500                                  ;*      READ SECTOR BACK AND MAKE SURE IT IS CORRECT.
6501                                  ;*
6502                                  ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
6503                                  ;*              EXISTS FOR SPECIFIED LOCATION.
6504                                  ;*
6505                                  ;;*****************************************************************
6506  020452  000004                  TST63:  SCOPE
6507  020454  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
6508  020462  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
6509  020470  032737  000200  001656          BIT     #PARBKO,OPTFLG  ;PARITY OPTION PRESENT?
6510  020476  001504                          BEQ     4$              ;YES-SKIP
6511
```

```
  6512  020500  004437  040776          1$:     JSR     R4,GENCOM       ;GENERATE DATA
  6513  020504  000014                          14                      ;PATTERN 14
  6514  020506  000400                          400                     ;400 WORDS
  6515
  6516  020510  012746  000340                  MOV     #PR7,-(SP)      ;SET PRIORITY TO 7
  6517  020514  012746  020522                  MOV     #10$,-(SP)      ;SET ADDRESS
  6518  020520  000002                          RTI
  6519  020522                          10$:
  6520
  6521  020522  032737  000100  001656          BIT     #PARBK1,OPTFLG  ;TEST IF BANK 1 AVAIL
  6522  020530  001403                          BEQ     5$              ;NO - SKIP
  6523  020532  012777  000004  161144          MOV     #BIT2,@MMCSR2   ;SET WRONG PARITY WRITE
  6524  020540  012777  000004  161134  5$:     MOV     #BIT2,@MMCSR1   ;SET WRONG PARITY BIT
  6525  020546  012737  177777  063026          MOV     #-1,OBUFF+222   ;WRITE WITH BAD PARITY
  6526  020554  012737  177777  063024          MOV     #-1,OBUFF+220
  6527  020562  012777  000001  161112          MOV     #BIT0,@MMCSR1   ;CLEAR WRONG PARITY, SET IE
  6528  020570  032737  000100  001656          BIT     #PARBK1,OPTFLG  ;TEST IF BANK 1 AVAIL
  6529  020576  001403                          BEQ     5$              ;NO - SKIP
  6530  020600  012777  000001  161076          MOV     #BIT0,@MMCSR2
  6531
  6532  020606  013746  001622          6$:     MOV     RKPRI,-(SP)     ;RESTORE PRIORITY
  6533  020612  012746  020620                  MOV     #11$,-(SP)
  6534  020616  000002                          RTI
  6535  020620                          11$:
  6536  020620  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
  6537  020622  104003                          ERROR   3               ;BAD INIT ERROR
  6538  020624  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
  6539  020630  010123                          WRDATA!CFMT                     ;WRDATA!CFMT
  6540  020632  177400                          -400                    ;-400 WORDS
  6541  020634  062604                          OBJFF                   ;OBUFF IS BUFF ADDRESS
  6542  020636     000                          .BYTE   0               ;SECTOR 0
  6543  020637     000                          .BYTE   0               ;TRACK 0
  6544  020640  000312                          312                     ;CYLINDER 312
  6545
  6546  020642  104417                          TLOADRK                 ;LOAD RK REGS
  6547  020644  104430                          TWAT96                  ;WAIT FOR INTERRUPT
  6548  020646  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
  6549
  6550  020650  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
  6551  020652  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
  6552                                  ;       A UNIBUS PARITY ERROR REPORTED AT THIS
  6553                                  ;       TIME INDICATES CONTROLLER IS NOT
  6554                                  ;       PROPERLY TRANSFERRING DATA IN 18 BIT MODE
  6555
  6556  020654  005077  161022                  CLR     @MMCSR1 ;CLEAR IE
  6557  020660  032737  000100  001656          BIT     #PARBK1,OPTFLG
  6558  020666  001402                          BEQ     7$
  6559  020670  005077  161010                  CLR     @MMCSR2
  6560  020674  005037  063026          7$:     CLR     OBUFF+222
  6561  020700  005037  063024                  CLR     OBUFF+220
  6562  020704  004737  033704                  JSR     PC,OPTTST       ;RESET OPTIONS
  6563
  6564  020710                          4$:
  6565                                  ;;*****************************************************************
  6566                                  ;*TEST 64       24 SECTOR FORMAT DATA TRANSFER (PART 3)
  6567                                  ;*
```

```
                                        ;*      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.
                                        ;*
                                        ;;*****************************************************************
020710  000004                  TST64:  SCOPE
020712  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
020720  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
020726  004737  033704                  JSR     PC,OPTTST       ;SET UP OPTIONS
020732  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
020734  104003                          ERROR   3               ;BAD INIT ERROR

020736  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
020742  000015                          15                      ;PATTERN 15
020744  001000                          1000                    ;1000 WORDS

020746  004437  040776                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
020752  002007                          2007
020754  001000                          1000

020756  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
020762  010123                          WRDATA!CFMT                     ;WRDATA!CFMT
020764  177000                          -1000                   ;-1000 WORDS
020766  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
020770     023                          .BYTE   23              ;SECTOR 23
020771     000                          .BYTE   0               ;TRACK 0
020772  000312                          312                     ;CYLINDER 312

020774  104417                          TLOADRK                 ;LOAD RK REGS
020776  104430                          TWAT96                  ;WAIT FOR INTERRUPT
021000  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

021002  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
021004  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

021006  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
021012  010121                          RDDATA!CFMT                     ;RDDATA!CFMT
021014  177000                          -1000                   ;-1000 WORDS
021016  060604                          IBUFF                   ;IBUFF IS BUFF ADDRESS
021020     023                          .BYTE   23              ;SECTOR 23
021021     000                          .BYTE   0               ;TRACK 0
021022  000312                          312                     ;CYLINDER 312

021024  104417                          TLOADRK                 ;LOAD RK REGS
021026  104426                          TWAT64                  ;WAIT FOR INTERRUPT
021030  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

021032  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
021034  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

021036  004437  040776                  JSR     R4,GENCOM       ;COMPARE DATA
021042  100000                          100000
021044  001000                          1000                    ;1000 WORDS
021046  000413                          BR      1$              ;NO MISCOMPARES-SKIP
021050  104015                          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.
                                ;*
                                ;;*****************************************************************
021076  000004         TST65:  SCOPE
021100  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
021106  012737  000312  001664          MOV     #312,REFMT      ;SET REFORMAT SWITCH
021114  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
021116  104003                          ERROR   3               ;BAD INIT ERROR

021120  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
021124  000127                          WRHEAD                  ;WRHEAD
021126  177676                          -102                    ;-102 WORDS
021130  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
021132     000                          .BYTE   0               ;SECTOR 0
021133     000                          .BYTE   0               ;TRACK 0
021134  000312                          312                     ;CYLINDER 312

021136  004437  040776                  JSR     R4,GENCOM       ;BUILD HEADERS
021142  000600                          600

021144  042737  040000  062614          BIC     #BIT14,OBUFF+10 ;MARK SECTOR 1 BAD
021152  042737  040000  062616          BIC     #BIT14,OBUFF+12 ;CORRECT HDRC

021160  104417                          TLOADRK                 ;LOAD RK REGS
021162  104431                          TWAIT12                 ;WAIT FOR INTERRUPT
021164  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

021166  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
021170  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
```

```
6680  021172  004437  040776              JSR     R4,GENCOM       ;GENERATE DATA
6681  021176  000016                      16                      ;PATTERN 16
6682  021200  001000                      1000                    ;1000 WORDS
6683
6684  021202  004437  040776              JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6685  021206  002007                      2007
6686  021210  001000                      1000
6687
6688  021212  004437  034574              JSR     R4,LRLOAD       ;LOAD "L" REGS
6689  021216  000123                      WRDATA                  ;WRDATA
6690  021220  177000                      -1000                   ;-1000 WORDS
6691  021222  062604                      OBUFF                   ;OBUFF IS BUFF ADDRESS
6692  021224     000                      .BYTE   0               ;SECTOR 0
6693  021225     000                      .BYTE   0               ;TRACK 0
6694  021226  000312                      312                     ;CYLINDER 312
6695
6696  021230  104417                      TLOADRK                 ;LOAD RK REGS
6697  021232  104424                      TWAT32                  ;WAIT FOR INTERRJPT
6698  021234  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6699
6700  021236  104422                      TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERR
6701  021240  000000                      0
6702  021242  000100                      BSERR                   ;BAD SECTOR ERROR
6703  021244  000000                      0
6704  021246  104004                      ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
6705  021250  005037  046520              CLR     GRP4ER          ;CLEAR GROUP 4 ERRORS
6706  021254  004437  036366              JSR     R4,CHKCTS       ;CHECK CYL, TRK, SECT CORRECT AFTER ABORTED WRITE
6707  021260  032737  000020  046520      BIT     #TRKERR,GRP4ER  ;TRK IN ERROR?
6708  021266  001416                      BEQ     1$              ;NO-SKIP
6709  021270  012737  053071  001450      MOV     #EM13,EM1ON     ;"TRACK ADDRESS INCORRECT"
6710  021276  013737  046474  001202      MOV     EXPTRK,$REG10   ;EXPECTED VALUE
6711  021304  013737  046506  001204      MOV     REALTRK,$REG11  ;REAL VALUE
6712  021312  012737  047205  057662      MOV     #OPER37,DF010A  ;"AFTER WRITE DATA TERMINATED WITH BSE"
6713  021320  104010                      ERROR   10
6714  021322  000527                      BR      5$              ;EXIT
6715
6716  021324  032737  000040  046520  1$: BIT     #SECERR,GRP4ER  ;SECTOR IN ERROR?
6717  021332  001422                      BEQ     3$              ;NO-SKIP
6718  021334  012737  053121  001450      MOV     #EM14,EM1ON     ;"SECTOR ADDRESS INCORRECT"
6719  021342  012737  047205  057662      MOV     #OPER37,DF010A  ;"AFTER WRITE DATA ABORTED WITH BSE"
6720  021350  013737  046472  001202      MOV     EXPSEC,$REG10   ;EXPECTED VALUE
6721  021356  013737  046510  001204      MOV     REALSEC,$REG11  ;REAL VALUE
6722  021364  104010                      ERROR   10
6723  021366  000505                      BR      5$              ;EXIT
6724  021370  104415                      SCOP1                   ;LOCAL LOOP TO BEGINNING OF TEST
6725  021372  012737  021400  001110      MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
6726  021400                          3$:
6727  021400  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
6728  021402  104003                      ERROR   3               ;BAD INIT ERROR
6729  021404  004437  034574              JSR     R4,LRLOAD       ;LOAD "L" REGS
6730  021410  000121                      RDDATA                  ;RDDATA
6731  021412  177400                      -400                    ;-400 WORDS
6732  021414  060604                      IBUFF                   ;IBUFF IS BUFF ADDRESS
6733  021416     000                      .BYTE   0               ;SECTOR 0
6734  021417     000                      .BYTE   0               ;TRACK 0
6735  021420  000312                      312                     ;CYLINDER 312
```

```
                                                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

021434  004437  040776                        JSR     R4,GENCOM       ;COMPARE DATA
021440  100000                                100000
021442  000400                                400                     ;400 WORDS
021444  000413                                BR      4$              ;NO MISCOMPARES-EXIT
021446  104015                                ERROR   15              ;REPORT FIRST ERROR

021450  013700  001634                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
021454  005300                        64$:    DEC     R0              ;DECREMENT COUNT
021456  001406                                BEQ     65$             ;IF ZERO - EXIT
021460  004437  040776                        JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
021464  040000                                40000
021466  000402                                BR      65$             ;NO MORE ERRORS - EXIT
021470  104016                                ERROR   16              ;REPORT NEXT ERROR
021472  000770                                BR      64$             ;LOOP
021474                              65$:

021474  004437  040776                4$:     JSR     R4,GENCOM       ;CLEAR IBUFF
021500  002007                                2007
021502  001000                                1000

021504  004437  034574                        JSR     R4,LRLOAD       ;LOAD "L" REGS
021510  000121                                RDDATA                  ;RDDATA
021512  177000                                -1000                   ;-1000 WORDS
021514  060604                                IBUFF                   ;IBUFF IS BUFF ADDRESS
021516     000                                .BYTE   0               ;SECTOR 0
021517     000                                .BYTE   0               ;TRACK 0
021520  000312                                312                     ;CYLINDER 312

021522  104417                                TLOADRK                 ;LOAD RK REGS
021524  104424                                TWAT32                  ;WAIT FOR INTERRUPT
021526  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

021530  104422                                TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
021532  000000                                0
021534  000100                                BSERR                   :BAD SECTOR ERROR
021536  000000                                0
021540  104004                                ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES

021542  004437  040776                        JSR     R4,GENCOM       ;COMPARE DATA AGAIN
021546  100000                                100000
021550  000400                                400                     ;400 WORDS
021552  000413                                BR      5$              ;NO MISCOMPARES
021554  104015                                ERROR   15              ;REPORT FIRST ERROR

021556  013700  001634                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
021562  005300                        66$:    DEC     R0              ;DECREMENT COUNT
021564  001406                                BEQ     67$             ;IF ZERO - EXIT
021566  004437  040776                        JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
```

```
                                                40000
                                                BR      67$             ;NO MORE ERRORS - EXIT
                                                ERROR   16              ;REPORT NEXT ERROR
                                                BR      66$             ;LOOP
                                        67$:

                                        5$:
                                        ;;*****************************************************************
                                        ;*TEST 66       FORMAT TEST
                                        ;*
                                        ;*      FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR FORMAT.
                                        ;*      MAKE SURE NO ERRORS SET.  READ SECTORS 0-25 AND MAKE
                                        ;*      SURE DATA CHECK DOES NOT OCCUR.
                                        ;*
                                        ;;*****************************************************************
021602  000004                          TST66:  SCOPE
021604  012737  000001  001262                MOV     #1,$TIMES       ;;DO 1 ITERATION
021612  005000                          1$:     CLR     R0              ;CLEAR TRACK COUNTER
021614  012737  021622  001110                MOV     #1$,$LPERR      ;SET LOCAL LOOP

021622                                  1$:
021622  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
021624  104003                                ERROR   3               ;BAD INIT ERROR

021626  013737  001626  001610                MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
021634  012737  000127  001600                MOV     #WRHEAD,L.CS1   ;LOAD WRITE HEADER
021642  110037  001607                        MOVB    R0,L.DT         ;LOAD DESIRED TRACK FROM TRACK COUNTER
021646  012737  062604  001604                MOV     #OBUFF,L.BA     ;LOAD BUS ADDRESS
021654  012737  177676  001602                MOV     #-102,L.WC      ;       WORD COUNT
021662  012737  000312  001614                MOV     #312,L.DCYL     ;       CYLINDER

021670  004437  040776                        JSR     R4,GENCOM       ;BUILD HEADER
021674  001200                                1200                    ;WITH BSE FLAGGED

021676  104417                                TLOADRK                 ;LOAD RK REGS
021700  104431                                TWAIT12                 ;WAIT FOR INTERRUPT
021702  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

021704  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
021706  104004                                ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

021710  104415                                SCOP1                   ;LOCAL LOOP TO 1$

021712  005700                                TST     R0              ;R0 AT ZERO?
021714  001002                                BNE     2$              ;NO-EXIT
021716  005200                                INC     R0              ;BUMP COUNTER
021720  000740                                BR      1$              ;LOOP
021722  005000                          2$:     CLR     R0              ;CLEAR SECTOR COUNTER
021724  012737  021732  001110                MOV     #113$,$LPERR    ;SET LOCAL LOOP ON ERROR

021732                                  113$:
021732  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
021734  104003                                ERROR   3               ;BAD INIT ERROR
021736                                  3$:
021736  004437  034574                        JSR     R4,LRLOAD       ;LOAD "L" REGS
021742  000121                                RDDATA                  ;RDDATA
```

```
6848  021744  177400                    -400                    ;-400 WORDS
6849  021746  060604                    IBUFF                   ;IBUFF IS BUFF ADDRESS
6850  021750     000                    .BYTE   0               ;SECTOR 0
6851  021751     000                    .BYTE   0               ;TRACK 0
6852  021752  000312                    312                     ;CYLINDER 312
6853
6854  021754  110037  001606            MOVB    R0,L.DS         ;LOAD SECTOR COUNTER INTO DESIRED SECTOR
6855
6856  021760  104417                    TLOADRK                 ;LOAD RK REGS
6857  021762  104424                    TWAT32                  ;WAIT FOR INTERRUPT
6858  021764  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6859
6860  021766  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6861  021770  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6862
6863  021772  104415                    SCOP1                   ;LOCAL LOOP TO 3$ ON ERROR
6864
6865  021774  022700  000024            CMP     #24,R0          ;LAST SECTOR READ?
6866  022000  001402                    BEQ     4$              ;YES-EXIT
6867  022002  005200                    INC     R0              ;BUMP SECTOR COUNTER
6868  022004  000754                    BR      3$              ;LOOP
6869
6870  022006  005037  001664  4$:       CLR     REFMT           ;CLEAR REFORMAT SWITCH
6871
6872                        .SBTTL  **WRITE CHECK TESTS
6873
6874                        ;;*****************************************************************
6875                        ;*TEST 67       WRITE-CHECK WITH NO ERROR
6876                        ;*
6877                        ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN.
6878                        ;*      DO A WRITE-CHECK OF 400 WORDS.  MAKE SURE NO
6879                        ;*      ERROR OCCURS.
6880                        ;*
6881                        ;;*****************************************************************
6882  022012  000004  TST67:  SCOPE
6883  022014  012737  000062  001262    MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
6884  022022  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
6885  022024  104003                    ERROR   3               ;BAD INIT ERROR
6886
6887  022026  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
6888  022032  000123                    WRDATA                  ;WRDATA
6889  022034  177400                    -400                    ;-400 WORDS
6890  022036  062604                    OBUFF                   ;OBUFF IS BUFF ADDRESS
6891  022040     000                    .BYTE   0               ;SECTOR 0
6892  022041     000                    .BYTE   0               ;TRACK 0
6893  022042  000312                    312                     ;CYLINDER 312
6894
6895  022044  004437  040776            JSR     R4,GENCOM       ;GENERATE DATA
6896  022050  000002                    2                       ;PATTERN 2
6897  022052  000400                    400                     ;400 WORDS
6898
6899  022054  104417                    TLOADRK                 ;LOAD RK REGS
6900  022056  104430                    TWAT96                  ;WAIT FOR INTERRUPT
6901  022060  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6902
6903  022062  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
```

```
                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

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

                        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
;;*****************************************************************
;*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:
;*
;*      000001  000020  000400  010000
;*      000002  000040  001000  020000
;*      000004  000100  002000  040000
;*      000010  000200  004000  100000
;*
;*      MAKE SURE WRITE CHECK ERROR SET FOR EACH
;*      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
;*      AND WORD COUNT IS CORRECT.
;*
;;*****************************************************************
TST70:  SCOPE
        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
        MOV     #BIT0,R0        ;SET LO ORDER BIT IN R0 FOR
                                ;CAUSING WRITE CHECK ERROR
        TSSINIT                 ;CLEAR SUBSYSTEM
        ERROR   3               ;BAD INIT ERROR
        JSR     R4,GENCOM       ;GENERATE DATA, ALL 0'S
        1
        400

        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

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

```
6960  022172  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6961  022174  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6962
6963  022176  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6964  022202  000131                          WRTCHK                  ;WRTCHK
6965  022204  177400                          -400                    ;-400 WORDS
6966  022206  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
6967  022210     000                          .BYTE   0               ;SECTOR 0
6968  022211     000                          .BYTE   0               ;TRACK 0
6969  022212  000312                          312                     ;CYLINDER 312
6970
6971  022214  104417                          TLOADRK                 ;LOAD RK REGS
6972  022216  104424                          TWAT32                  ;WAIT FOR INTERRUPT
6973  022220  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6974
6975  022222  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6976  022224  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6977
6978  022226  104415                          SCOP1                   ;LOCAL LOOP ON WRITE CHECK
6979  022230  012737  022236  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP
6980  022236  010037  063024          1$:     MOV     R0,OBUFF+220    ;CAUSE ERROR BIT IN BUFFER
6981  022242  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6982  022244  104003                          ERROR   3               ;BAD INIT ERROR
6983  022246  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6984  022252  000131                          WRTCHK                  ;WRTCHK
6985  022254  177400                          -400                    ;-400 WORDS
6986  022256  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
6987  022260     000                          .BYTE   0               ;SECTOR 0
6988  022261     000                          .BYTE   0               ;TRACK 0
6989  022262  000312                          312                     ;CYLINDER 312
6990
6991  022264  104417                          TLOADRK                 ;LOAD RK REGS
6992  022266  104424                          TWAT32                  ;WAIT FOR INTERRUPT
6993  022270  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6994
6995  022272  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
6996  022274  000000                          0
6997  022276  000004                          WCKERR                  ;WRITE CHECK ERROR
6998  022300  000000                          0
6999  022302  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
7000
7001  022304  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
7002
7003                                  ;       NOTE:   THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF TWO
7004                                  ;       VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE
7005                                  ;       WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE
7006                                  ;       NPR CYCLE, WC AND BA WILL BE ONE PAST WHERE THE ERROR ACTUALLY
7007                                  ;       OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST
7008                                  ;       NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS REFLECT THE
7009                                  ;       ACTUAL WORD WHERE THE ERROR WAS.
7010
7011  022306  023727  001544  063026          CMP     T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
7012  022314  001416                          BEQ     2$              ;YES-SKIP
7013  022316  101037                          BHI     6$              ;IF TO HI - SKIP
7014  022320  012737  052734  001450          MOV     #EM11,EM10N     ;"INCORRECT BA"
7015  022326  012737  063026  001202          MOV     #OBUFF+222,$REG10  ;GOOD VALUE
```

```
7016  022334  013737  001544  001204          MOV     T.BA,$REG11     ;BAC VALUE
7017  022342  012737  047241  057662          MOV     #OPER41,DFO10A  ;"WRITE CHECK ABORTED WITH WCE"
7018  022350  104010                          ERROR   10
7019
7020  022352  023727  001542  177511  2$:     CMP     T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7021  022360  001460                          BEQ     3$              ;YES-SKIP
7022  022362  012737  052707  001450          MOV     #EM10,EM10N     ;"INCORRECT WC"
7023  022370  012737  047241  057662          MOV     #OPER41,DFO10A  ;"WRITE CHECK ABORTED WITH WCE"
7024  022376  012737  177511  001202          MOV     #-267,$REG10    ;GOOD VALUE
7025  022404  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7026  022412  104010                          ERROR   10
7027  022414  000442                          BR      3$              ;EXIT
7028
7029  022416  023727  001544  063030  6$:     CMP     T.BA,#OBUFF+224 ;TEST IF BA AT HI SIDE
7030  022424  001415                          BEQ     7$              ;YES - SKIP
7031  022426  012737  052734  001450          MOV     #EM11,EM10N     ;SET MESSAGE
7032  022434  012737  063030  001202          MOV     #OBUFF+224,$REG10 ;GOOD VALUE
7033  022442  013737  001544  001204          MOV     T.BA,$REG11     ;ERROR VALUE
7034  022450  012737  047241  057662          MOV     #OPER41,DFO10A  ;"WRITE CHECK ABORTED WITH WCE"
7035  022456  104010                          ERROR   10
7036
7037  022460  023727  001542  177512  7$:     CMP     T.WC,#-266      ;TEST IF WORD COUNT AT HI SIDE
7038  022466  001415                          BEQ     3$              ;YES - SKIP
7039  022470  012737  052707  001450          MOV     #EM10,EM10N     ;SET MESSAGE
7040  022476  012737  047241  057662          MOV     #OPER41,DFO10A  ;"WC ABORTED WITH WCE"
7041  022504  012737  177512  001202          MOV     #-266,$REG10    ;GOOD VALUE
7042  022512  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7043  022520  104010                          ERROR   10
7044
7045  022522  104415              3$:         SCOPE                   ;LOCAL LOOP ON ERROR TO 1$
7046
7047  022524  032700  100000                  BIT     #BIT15,R0       ;BIT 15 SET?
7048  022530  001002                          BNE     4$              ;YES-EXIT
7049  022532  006300                          ASL     R0              ;SHIFT ERROR BIT
7050  022534  000640                          BR      1$              ;LOOP
7051
7052  022536              4$:
7053                      ;;*****************************************************************
7054                      ;*TEST 71        WRITE CHECK ERROR (PART 2)
7055                      ;*
7056                      ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
7057                      ;*      IN ALL WORDS.  WRITE CHECK CYLINDER 312, TRACK 0,
7058                      ;*      SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
7059                      ;*      ONE OF THE FOLLWOING CONFIGURATIONS:
7060                      ;*
7061                      ;*      177776  177757  177377  167777
7062                      ;*      177775  177737  176777  157777
7063                      ;*      177773  177677  175777  137777
7064                      ;*      177767  177577  173777  077777
7065                      ;*
7066                      ;*      MAKE SURE WRITE CHECK ERROR SET FOR EACH
7067                      ;*      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
7068                      ;*      AND WORD COUNT IS CORRECT.
7069                      ;*
7070                      ;*****************************************************************
7071   022536  000004     TST71:  SCOPE
```

```
7072  022540  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
7073  022546  012700  177776                  MOV     #177776,R0      ;LOAD R0 FOR CAUSING WRITE CHECK ERROR
7074
7075  022552  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7076  022554  104003                          ERROR   3               ;BAD INIT ERROR
7077  022556  004437  040776                  JSR     R4,GENCOM       ;GENERATE DATA
7078  022562  000007                          7                       ;ALL 1'S
7079  022564  000400                          400                     ;400 WORDS
7080
7081  022566  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7082  022572  000123                          WRDATA                  ;WRDATA
7083  022574  177400                          -400                    ;-400 WORDS
7084  022576  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7085  022600     000                          .BYTE   0               ;SECTOR 0
7086  022601     000                          .BYTE   0               ;TRACK 0
7087  022602  000312                          312                     ;CYLINDER 312
7088
7089  022604  104417                          TLOADRK                 ;LOAD RK REGS
7090  022606  104430                          TWAT96                  ;WAIT FOR INTERRUPT
7091  022610  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7092
7093  022612  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7094  022614  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7095
7096  022616  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7097  022622  000131                          WRTCHK                  ;WRTCHK
7098  022624  177400                          -400                    ;-400 WORDS
7099  022626  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7100  022630     000                          .BYTE   0               ;SECTOR 0
7101  022631     000                          .BYTE   0               ;TRACK 0
7102  022632  000312                          312                     ;CYLINDER 312
7103
7104  022634  104417                          TLOADRK                 ;LOAD RK REGS
7105  022636  104424                          TWAT32                  ;WAIT FOR INTERRUPT
7106  022640  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7107
7108  022642  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7109  022644  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7110
7111  022646  104415                          SCOP1           ;LOCAL LOOP TO START OF TEST
7112  022650  012737  022656  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP
7113
7114  022656  010037  063024          1$:     MOV     R0,OBUFF+220    ;PUT WORD IN OBUFF TO CAUSE WCE
7115  022662  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7116  022664  104003                          ERROR   3               ;BAD INIT ERROR
7117
7118  022666  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7119  022672  000131                          WRTCHK                  ;WRTCHK
7120  022674  177400                          -400                    ;-400 WORDS
7121  022676  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7122  022700     000                          .BYTE   0               ;SECTOR 0
7123  022701     000                          .BYTE   0               ;TRACK 0
7124  022702  000312                          312                     ;CYLINDER 312
7125
7126  022704  104417                          TLOADRK                 ;LOAD RK REGS
7127  022706  104424                          TWAT32                  ;WAIT FOR INTERRUPT
```

```
7128  022710  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7129
7130  022712  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7131  022714  000000                          0
7132  022716  000004                          WCKERR                  ;WRITE CHECK ERROR
7133  022720  000000                          0
7134  022722  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
7135
7136  022724  104415                          SCOP1                   ;LOCAL LOOP TO 1$
7137
7138                                ;         NOTE:   THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF TWO
7139                                ;         VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE
7140                                ;         WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE
7141                                ;         NPR CYCLE, WC AND BA WILL BE ONE PAST WHERE THE ERROR ACTUALLY
7142                                ;         OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST
7143                                ;         NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS REFLECT THE
7144                                ;         ACTUAL WORD WHERE THE ERROR WAS.
7145
7146  022726  023727  001544  063026          CMP     T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
7147  022734  001416                          BEQ     2$              ;YES-SKIP
7148  022736  101037                          BHI     6$              ;IF TO HI - SKIP
7149  022740  012737  052734  001450          MOV     #EM11,EM1ON     ;"INCORRECT BA"
7150  022746  012737  063026  001202          MOV     #OBUFF+222,$REG10  ;GOOD VALUE
7151  022754  013737  001544  001204          MOV     T.BA,$REG11     ;BAD VALUE
7152  022762  012737  047241  057662          MOV     #OPER41,DF01OA  ;"WRITE CHECK ABORTED WITH WCE"
7153  022770  104010                          ERROR   10
7154
7155  022772  023727  001542  177511  2$:     CMP     T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7156  023000  001460                          BEQ     3$              ;YES-SKIP
7157  023002  012737  052707  001450          MOV     #EM10,EM1ON     ;"INCORRECT WC"
7158  023010  012737  047241  057662          MOV     #OPER41,DF01OA  ;"WRITE CHECK ABORTED WITH WCE"
7159  023016  012737  177511  001202          MOV     #-267,$REG10    ;GOOD VALUE
7160  023024  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7161  023032  104010                          ERROR   10
7162  023034  000442                          BR      3$              ;EXIT
7163
7164  023036  023727  001544  063030  6$:     CMP     T.BA,#OBUFF+224 ;TEST IF BA AT HI SIDE
7165  023044  001415                          BEQ     7$              ;YES - SKIP
7166  023046  012737  052734  001450          MOV     #EM11,EM1ON     ;SET MESSAGE
7167  023054  012737  063030  001202          MOV     #OBUFF+224,$REG10 ;GOOD VALUE
7168  023062  013737  001544  001204          MOV     T.BA,$REG11     ;ERROR VALUE
7169  023070  012737  047241  057662          MOV     #OPER41,DF01OA  ;"WRITE CHECK ABORTED WITH WCE"
7170  023076  104010                          ERROR   10
7171
7172  023100  023727  001542  177512  7$:     CMP     T.WC,#-266      ;TEST IF WORD COUNT AT HI SIDE
7173  023106  001415                          BEQ     3$              ;YES - SKIP
7174  023110  012737  052707  001450          MOV     #EM10,EM1ON     ;SET MESSAGE
7175  023116  012737  047241  057662          MOV     #OPER41,DF01OA  ;"WC ABORTED WITH WCE"
7176  023124  012737  177512  001202          MOV     #-266,$REG10    ;GOOD VALUE
7177  023132  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7178  023140  104010                          ERROR   10
7179
7180  023142  104415                  3$:     SCOP1
7181
7182  023144  032700  100000                  BIT     #BIT15,R0       ;BIT 15 SET? (ALL PATTERNS TESTED)
7183  023150  001002                          BNE     4$              ;YES-EXIT
```

```
7184  023152  006300                          ASL     R0              ;SHIFT FOR NEXT TEST
7185  023154  000640                          BR      1$              ;LOOP
7186
7187  023156                              4$:
7188                                      ;;*****************************************************************
7189                                      ;*TEST 72       WRITE CHECK OF PARTIAL SECTOR
7190                                      ;*
7191                                      ;*      WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN
7192                                      ;*      CONFIGURATIONS.  ISSUE A WRITE CHECK COMMAND OF
7193                                      ;*      110 WORDS MAKING SURE THE 111TH WORD IS
7194                                      ;*      DIFFERENT THAN DATA ON DISK.  MAKE SURE
7195                                      ;*      WFITE CHECK ERROR DOES NOT SET.
7196                                      ;*
7197                                      ;;*****************************************************************
7198  023156  000004                      1$72:   SCOPE
7199  023160  012737  000062  001262              MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
7200  023166  104416                              TSSINIT                 ;CLEAR SUBSYSTEM
7201  023170  104003                              ERROR   3               ;BAD INIT ERROR
7202
7203  023172  004437  040776                      JSR     R4,GENCOM       ;GENERATE DATA
7204  023176  000007                              7                       ;ALL 1'S
7205  023200  000400                              400                     ;400 WORDS
7206
7207  023202  004437  034574                      JSR     R4,LRLOAD       ;LOAD "L" REGS
7208  023206  000123                              WRDATA                  ;WRDATA
7209  023210  177400                              -400                    ;-400 WORDS
7210  023212  062604                              OBUFF                   ;OBUFF IS BUFF ADDRESS
7211  023214     000                              .BYTE   0               ;SECTOR 0
7212  023215     000                              .BYTE   0               ;TRACK 0
7213  023216  000312                              312                     ;CYLINDER 312
7214
7215  023220  104417                              TLOADRK                 ;LOAD RK REGS
7216  023222  104430                              TWAT96                  ;WAIT FOR INTERRUPT
7217  023224  104002                              ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7218
7219  023226  104421                              TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7220  023230  104004                              ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7221
7222  023232  005037  063026                      CLR     OBUFF+222
7223
7224  023236  004437  034574                      JSR     R4,LRLOAD       ;LOAD "L" REGS
7225  023242  000131                              WRTCHK                  ;WRTCHK
7226  023244  177670                              -110                    ;-110 WORDS
7227  023246  062604                              OBUFF                   ;OBUFF IS BUFF ADDRESS
7228  023250     000                              .BYTE   0               ;SECTOR 0
7229  023251     000                              .BYTE   0               ;TRACK 0
7230  023252  000312                              312                     ;CYLINDER 312
7231
7232  023254  104417                              TLOADRK                 ;LOAD RK REGS
7233  023256  104424                              TWAT32                  ;WAIT FOR INTERRUPT
7234  023260  104002                              ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7235
7236  023262  104421                              TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7237  023264  104004                              ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7238
7239                                  .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.
                                ;*
                                ;;*****************************************************************
023266  000004                  TST73:  SCOPE
023270  012737  000005  001262          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
023276  032737  000400  001656          BIT     #FPFMT,OPTFLG   ;TEST IF FIRST PASS SWITCH SET
023304  001043                          BNE     24$             ;YES - SKIP FORMAT

023306  105037  023331                  CLRB    21$             ;CLEAR ADDRESS POINTERS
023312  005037  023332                  CLR     22$

023316  004437  034574          20$:    JSR     R4,LRLOAD       ;LOAD "L" REGISTERS
023322  000127                          WRHEAD                  ;WRITE HEADER
023324  177676                          -102                    ;102 WORDS
023326  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
023330     000                          .BYTE   0               ;SECTOR 0
023331     000          21$:            .BYTE   0               ;TRACK ADDRESS (VARIABLE)
023332  000000          22$:            0                       ;CYLINDER 0 (VARIABLE)

023334  004437  040776                  JSR     R4,GENCOM       ;GO GENERATE HEADERS
023340  001200                          1200                    ;WITH BAD SECTOR ERRORS

023342  104417                          TLOADRK                 ;LOAD RK REGS
023344  104431                          TWAT112                 ;WAIT FOR INTERRUPT
023346  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

023350  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
023352  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

023354  123727  023331  000002          CMPB    21$,#2          ;TEST IF LAST TRACK
023362  001403                          BEQ     23$             ;YES - SKIP
023364  105237  023331                  INCB    21$             ;ELSE BUMP TRACK
023370  000752                          BR      20$             ;LOOP

023372  105037  023331          23$:    CLRB    21$             ;CLEAR TRACK POINTER
023376  023727  023332  000003          CMP     22$,#3          ;TEST IF LAST CYLINDER WRITTEN
023404  001403                          BEQ     24$             ;YES - SKIP
023406  005237  023332                  INC     22$             ;ELSE BUMP CYLINDER
023412  000741                          BR      20$             ;LOOP
```

```
7296
7297  023414  013737  023426  001110  24$:    MOV     1$,$LPERR       ;SET LOCAL LOOP ON ERROR
7298  023422  012703  000400                  MOV     #400,R3         ;SET COUNT FOR SECTOR CLEARING
7299  023426                          1$:
7300  023426  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7301  023430  104003                          ERROR   3               ;BAD INIT ERROR
7302  023432  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7303  023436  000123                          WRDATA                  ;WRDATA
7304  023440  177400                          -400                    ;-400 WORDS
7305  023442  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7306  023444     000                          .BYTE   0               ;SECTOR 0
7307  023445     000                          .BYTE   0               ;TRACK 0
7308  023446  000000                          0                       ;CYLINDER 0
7309
7310  023450  104417                          TLOADRK                 ;LOAD RK REGS
7311  023452  104434                          TWAT159                 ;WAIT FOR INTERRUPT
7312  023454  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7313
7314  023456  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7315  023460  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7316
7317  023462  104415                          SCOPI                   ;LOCAL LOOP ON ERROR TO 1$
7318  023464  005303                          DEC     R3              ;DECREMENT COUNT
7319  023466  012762  062604  000004  2$:     MOV     #OBUFF,RKBA(R2) ;SET BA
7320  023474  012762  177400  000002          MOV     #-400,RKWC(R2)  ;AND WC AGAIN
7321  023502  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
7322  023506  013762  001626  000010          MOV     DRVNUM,RKCS2(R2)        ;SET DRIVE NUMBER
7323  023514  012762  000123  000000          MOV     #WRDATA,RKCS1(R2) ;DO WRITE DATA
7324
7325  023522  104425                          TWAT48                  ;WAIT FOR INTERRUPT
7326  023524  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7327
7328  023526  032762  000200  000014          BIT     #BSE,RKER(R2)   ;BAD SECTOR ERROR?
7329  023534  001415                          BEQ     3$              ;NO-SKIP
7330  023536  032737  000200  001656          BIT     #BSERPT,OPTFLG  ;TEST IF BSE TO MANY HAS BEEN REPORTED
7331  023544  001007                          BNE     5$              ;YES - SKIP
7332  023546  052737  000200  001656          BIS     #BSERPT,OPTFLG  ;SET FLAG
7333  023554  012737  051362  001360          MOV     #OPR017,EMIN    ;SET MESSAGE
7334  023562  104001                          ERROR   1               ;"FIRST 256 SECTOR NOT BSE FREE"
7335  023564  000137  024120          5$:     JMP     14$             ;GO TO EXIT
7336                                  ;
                                      CHECK FOR ANY OTHER ERRORS
7337  023570                          3$:
7338  023570  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7339  023572  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7340  023574  104415                          SCOPI                   ;LOCAL LOOP TO 1$ (RESTART SECTOR CLEAR)
7341
7342  023576  005303                          DEC     R3              ;DECREMENT COUNT
7343  023600  001332                          BNE     2$              ;LOOP IF NOT ZERO
7344
7345  023602  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7346  023606  000117                          SEEK                    ;SEEK
7347  023610  000000                          0                       ;0 WORDS
7348  023612  000000                          0                       ;0 IS BUFF ADDRESS
7349  023614     000                          .BYTE   0               ;SECTOR 0
7350  023615     000                          .BYTE   0               ;TRACK 0
7351  023616  000000                          0                       ;CYLINDER 0
```

```
                                                ;LOAD RK REGS
                                                ;WAIT FOR INTERRUPT
                                                ;TO SLOW/NOT COMPLETE ERROR
                                                ;CLEAR FIRST INTERRUPT
                                                ;CHECK OPERATION FOR ANY ERRORS
                                                ;REPORT ALL ERRORS
                                                ;WAIT FOR SECOND INTERRUPT
                                                ;TO SLOW/NOT COMPLETE ERROR
                                                ;GENERATE DATA
                                                ;PATTERN 6, 1ST WORD REPEATED
                                                ;400 WORDS

                                                ;CLEAR SUBSYSTEM
                                                ;BAD INIT ERROR
                                                ;LOAD "L" REGS
                                                ;WRDATA
                                                ;0 WORDS
                                                ;OBUFF IS BUFF ADDRESS
                                                ;SECTOR 0
                                                ;TRACK 0
                                                ;CYLINDER 0

                                                ;LOAD RK REGS
                                                ;WAIT FOR SECOND INTERRUPT
                                                ;ELSE REPORT TO SLOW/NOT COMPLETE ERROR

                                                ;CHECK OPERATION FOR ANY ERRORS
                                                ;REPORT ALL ERRORS

                                                ;INTERNAL LOOP ON ERROR TO 4$
                                                ;SET COUNTER FOR READ-COMPARE LOOP
                                                ;CLEAR SECTOR AND TRACK VALUES
                                                ;CLEAR CYL VALUE
                                                ;SET LOCAL LOOP ON ERROR
                                                ;CLEAR BAI

                                                ;CLEAR SUBSYSTEM
                                                ;BAD INIT ERROR
                                                ;LOAD RK REGS
                                                ;READ DATA
                                                ;400 WORDS
                                                ;INTO IBUFF
                                                ;SECTOR (VARIABLE)
                                                ;TRACK  (VARIABLE)
                                                ;CYL    (VARIABLE)

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

```
7408  023776  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7409  024000  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7410
7411  024002  104415                  SCOP1                   ;LOCAL LOOP ON ERROR TO 8$
7412
7413  024004  004437  040776          JSR     R4,GENCOM       ;COMPARE DATA
7414  024010  100000                  100000
7415  024012  000400                  400                     ;400 WORDS
7416  024014  000413                  BR      13$             ;NO MISCOMPARE-EXIT LOOP
7417  024016  104015                  ERROR   15              ;REPORT FIRST ERROR
7418
7419  024020  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
7420  024024  005300          64$:    DEC     R0              ;DECREMENT COUNT
7421  024026  001406                  BEQ     65$             ;IF ZERO - EXIT
7422  024030  004437  040776          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
7423  024034  040000                  40000
7424  024036  000402                  BR      65$             ;NO MORE ERRORS - EXIT
7425  024040  104016                  ERROR   16              ;REPORT NEXT ERROR
7426  024042  000770                  BR      64$             ;LOOP
7427  024044          65$:
7428
7429  024044  104415          13$:    SCOP1                   ;LOCAL LOOP TO 8$
7430
7431  024046  005303                  DEC     R3              ;DEC READ LOOP COUNT
7432  024050  001423                  BEQ     14$             ;IF ZERO-EXIT
7433
7434  024052  105237  023764          INCB    10$             ;BUMP SECTOR
7435  024056  123727  023764  000026  CMPB    10$,#26         ;FINISHED WITH TRACK?
7436  024064  001332                  BNE     9$              ;NO-LOOP
7437  024066  105037  023764          CLRB    10$             ;CLEAR SECTOR
7438  024072  105237  023765          INCB    11$             ;BUMP TRACK
7439  024076  123727  023765  000003  CMPB    11$,#3          ;FINISHED WITH CYLINDER?
7440  024104  001322                  BNE     9$              ;NO-LOOP
7441  024106  105037  023765          CLRB    11$             ;CLEAR TRACK
7442  024112  005237  023766          INC     12$             ;BUMP CYL.
7443  024116  000715                  BR      9$              ;LOOP
7444
7445  024120                  14$:
7446                          ;;*****************************************************************
7447                          ;*TEST 74       MAXIMUM DATA TRANSFER (PART 2)
7448                          ;*
7449                          ;*      ZERO OUT FIRST 256 SECTORS OF THE DISK WITH
7450                          ;*      200000 WORD WRITE.  SEEK TO CYLINDER 632.
7451                          ;*      ISSJE A WRITE OF MAXIMUM DATA TRANSFER
7452                          ;*      200000 WORD WRITE.  MAKE SURE CONTROLLER TIME
7453                          ;*      OUT IS NOT SET.  CHECK CYLINDER ADDRESS
7454                          ;*      DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
7455                          ;*      SEEK TO CYLINDER 632.  ISSUE A WRITE CHECK
7456                          ;*      OF 200000 WORDS.  MAKE SURE NO ERROR SETS.
7457                          ;*
7458                          ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
7459                          ;*              IN THE FIRST 256 SECTORS ON THE PACK.
7460                          ;*
7461                          ;;*****************************************************************
7462  024120  000004          TST74:  SCOPE
7463  024122  012737  000005  001262          MOV     #5,%TIMES       ;;DO 5 ITERATIONS
```

```
7464  024130  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
7465  024132  104003                        ERROR   3               ;BAD INIT ERROR
7466  024134  012700  000620                MOV     #400.,R0        ;SET COUNT FOR INTERRUPT WAIT
7467  024140  005037  062604                CLR     OBUFF
7468
7469  024144  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
7470  024150  000123                        WRDATA                  ;WRDATA
7471  024152  000000                        0                       ;0 WORDS
7472  024154  062604                        OBUFF                   ;OBUFF IS BUFF ADDRESS
7473  024156     000                        .BYTE   0               ;SECTOR 0
7474  024157     000                        .BYTE   0               ;TRACK 0
7475  024160  000000                        0                       ;CYLINDER 0
7476  024162  052737  000020  001610        BIS     #BAI,L.CS2      ;SET BAI
7477  024170  104417                        TLOADRK                 ;LOAD RK REGS
7478  024172  104434                1$:     TWAIT59                 ;WAIT FOR INTERRUPT
7479  024174  000401                        BR      2$              ;NO INTERRUPT-SKIP
7480  024176  000403                        BR      3$              ;INTERRUPT-SKIP
7481
7482  024200  005300                2$:     DEC     R0              ;DEC WAIT COUNTER
7483  024202  001373                        BNE     1$              ;NO ZERO-LOOP
7484  024204  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7485
7486  024206  032762  000200  000014  3$:   BIT     #BSE,RKER(R2)   ;DID BSE SET
7487  024214  001415                        BEQ     4$              ;NO-SKIP
7488
7489  024216  032737  000200  001656        BIT     #BSERPT,OPTFLG  ;TEST IF TO MANY BAD SECTORS REPORTED
7490  024224  001007                        BNE     12$             ;YES - SKIP
7491  024226  052737  000200  001656        BIS     #BSERPT,OPTFLG  ;SET FLAG
7492  024234  012737  051362  001360        MOV     #OPR017,EMIN    ;SET MESSAGE
7493  024242  104001                        ERROR   1               ;"FIRST 256 SECTORS NOT BSE FREE"
7494  024244  000137  024576          12$:  JMP     11$             ;EXIT
7495
7496  024250                          4$:
7497  024250  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7498  024252  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7499
7500  024254  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
7501  024260  000117                        SEEK                    ;SEEK
7502  024262  000000                        0                       ;0 WORDS
7503  024264  000000                        0                       ;0 IS BUFF ADDRESS
7504  024266     000                        .BYTE   0               ;SECTOR 0
7505  024267     000                        .BYTE   0               ;TRACK 0
7506  024270  000632                        632                     ;CYLINDER 632
7507
7508  024272  104417                        TLOADRK                 ;LOAD RK REGS
7509  024274  104423                        TWAIT16                 ;WAIT FOR INTERRUPT
7510  024276  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7511  024300  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG
7512
7513  024304  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7514  024306  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7515
7516  024310  104427                        TWAIT80                 ;WAIT FOR 2ND INTERRUPT
7517  024312  104002                        ERROR   2
7518
7519  024314  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
```

```
7520  024316  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7521
7522  024320  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7523  024324  000105                          CLEAR                   ;CLEAR
7524  024326  000000                          0                       ;0 WORDS
7525  024330  000000                          0                       ;0 IS BUFF ADDRESS
7526  024332     000                          .BYTE   0               ;SECTOR 0
7527  024333     000                          .BYTE   0               ;TRACK 0
7528  024334  000000                          0                       ;CYLINDER 0
7529
7530  024336  104417                          TLOADRK                 ;LOAD RK REGS
7531  024340  104423                          TWAT16                  ;WAIT FOR INTERRUPT
7532  024342  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7533
7534  024344  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7535  024346  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7536
7537  024350  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7538  024354  000123                          WRDATA                  ;WRDATA
7539  024356  000000                          0                       ;0 WORDS
7540  024360  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7541  024362     000                          .BYTE   0               ;SECTOR 0
7542  024363     000                          .BYTE   0               ;TRACK 0
7543  024364  000000                          0                       ;CYLINDER 0
7544
7545  024366  012737  135143  062604          MOV     #135143,OBUFF   ;SET WORD FOR OUTPUT
7546  024374  012700  000621                  MOV     #401.,R0        ;SET COUNT FOR INTERRUPT WAIT
7547  024400  052737  000020  001610          BIS     #BAI.L.CS2      ;SET BUS ADDRESS INC INHIBIT
7548
7549  024406  104417                          TLOADRK                 ;LOAD RK REGS
7550  024410  104434                  5$:     TWAT159                 ;WAIT FOR INTERRUPT
7551  024412  000401                          BR      6$              ;NO INTERRUPT-BRANCH
7552  024414  000403                          BR      7$              ;INTERRUPT-BRANCH
7553
7554  024416  005300                  6$:     DEC     R0              ;DEC WAIT COUNT
7555  024420  001373                          BNE     5$              ;LOOP IF NOT ZERO
7556  024422  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7557
7558  024424                          7$:
7559  024424  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7560  024426  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7561  024430  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7562  024434  000117                          SEEK                    ;SEEK
7563  024436  000000                          0                       ;0 WORDS
7564  024440  000000                          0                       ;0 IS BUFF ADDRESS
7565  024442     000                          .BYTE   0               ;SECTOR 0
7566  024443     000                          .BYTE   0               ;TRACK 0
7567  024444  000632                          632                     ;CYLINDER 632
7568  024446  104417                          TLOADRK                 ;LOAD RK REGS
7569  024450  104423                          TWAT16                  ;WAIT FOR INTERRUPT
7570  024452  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7571  024454  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
7572
7573  024460  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7574  024462  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7575
```

```
7576  024464  104427                          TWAT80                  ;WAIT FOR SECOND INIT
7577  024466  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7578  024470  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7579  024472  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7580
7581  024474  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7582  024500  000105                          CLEAR                   ;CLEAR
7583  024502  000000                          0                       ;0 WORDS
7584  024504  000000                          0                       ;0 IS BUFF ADDRESS
7585  024506     000                          .BYTE   0               ;SECTOR 0
7586  024507     000                          .BYTE   0               ;TRACK 0
7587  024510  000000                          0                       ;CYLINDER 0
7588
7589  024512  104417                          TLOADRK                 ;LOAD RK REGS
7590  024514  104423                          TWAT16                  ;WAIT FOR INTERRUPT
7591  024516  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7592
7593  024520  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7594  024522  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7595
7596  024524  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7597  024530  000131                          WRTCHK                  ;WRTCHK
7598  024532  000000                          0                       ;0 WORDS
7599  024534  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7600  024536     000                          .BYTE   0               ;SECTOR 0
7601  024537     000                          .BYTE   0               ;TRACK 0
7602  024540  000000                          0                       ;CYLINDER 0
7603  024542  052737  000020  001610          BIS     #BAI.L.CS2      ;SET BAI FLAG
7604  024550  012700  000621                  MOV     #401.,R0        ;SET WAIT COUNT
7605
7606  024554  104417                          TLOADRK                 ;LOAD RK REGS
7607  024556  104434                  8$:     TWAT159                 ;WAIT FOR INTERRUPT
7608  024560  000401                          BR      9$              ;NO INTERRUPT-SKIP
7609  024562  000403                          BR      10$             ;INTERRUPT-SKIP
7610
7611  024564  005300                  9$:     DEC     R0              ;DEC WAIT COUNT
7612  024566  001373                          BNE     8$              ;NOT ZERO-LOOP
7613  024570  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7614
7615  024572                          10$:
7616  024572  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7617  024574  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7618
7619  024576                          11$:
7620                                  ;;*****************************************************************
7621                                  ;*TEST 75       CONTROLLER TIME OUT
7622                                  ;*
7623                                  ;*      SEEK TO CYLINDER 632.  ISSUE A RECALIBRATE AND DO NOT
7624                                  ;*      WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER
7625                                  ;*      OF CYLINDER 0, TRACK 0.  MAKE SURE CONTROLLER TIME
7626                                  ;*      OUT SETS.
7627                                  ;*
7628                                  ;;*****************************************************************
7629  024576  000004                  TST75:  SCOPE
7630  024600  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
7631  024606  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
```

```
7632  024610  104003                    ERROR   3               ;BAD INIT ERROR
7633
7634  024612  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
7635  024616  000117                    SEEK                    ;SEEK
7636  024620  000000                    0                       ;0 WORDS
7637  024622  000000                    0                       ;0 IS BUFF ADDRESS
7638  024624     000                    .BYTE   0               ;SECTOR 0
7639  024625     000                    .BYTE   0               ;TRACK 0
7640  024626  000632                    632                     ;CYLINDER 632
7641
7642  024630  104417                    TLOADRK                 ;LOAD RK REGS
7643  024632  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7644  024634  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7645
7646  024636  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7647  024640  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7648
7649  024642  005037  001662            CLR     INTSET          ;CLEAR INTERRUPT FLAG
7650  024646  104427                    TWAT80                  ;WAIT FOR SECOND INTERRUPT
7651  024650  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7652  024652  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7653  024654  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7654
7655  024656  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
7656  024662  000105                    CLEAR                   ;CLEAR
7657  024664  000000                    0                       ;0 WORDS
7658  024666  000000                    0                       ;0 IS BUFF ADDRESS
7659  024670     000                    .BYTE   0               ;SECTOR 0
7660  024671     000                    .BYTE   0               ;TRACK 0
7661  024672  000000                    0                       ;CYLINDER 0
7662
7663  024674  104417                    TLOADRK                 ;LOAD RK REGS
7664  024676  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7665  024700  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7666
7667  024702  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7668  024704  104004                    ERROR   4  ;OR 5. 6. 7  ;REPORT ALL ERRORS
7669
7670  024706  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
7671  024712  000113                    RECAL                   ;RECAL
7672  024714  000000                    0                       ;0 WORDS
7673  024716  000000                    0                       ;0 IS BUFF ADDRESS
7674  024720     000                    .BYTE   0               ;SECTOR 0
7675  024721     000                    .BYTE   0               ;TRACK 0
7676  024722  000000                    0                       ;CYLINDER 0
7677
7678  024724  104417                    TLOADRK                 ;LOAD RK REGS
7679  024726  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7680  024730  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7681
7682  024732  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
7683  024736  000125                    RDHEAD                  ;RDHEAD
7684  024740  000000                    0                       ;0 WORDS
7685  024742  000000                    0                       ;0 IS BUFF ADDRESS
7686  024744     000                    .BYTE   0               ;SECTOR 0
7687  024745     000                    .BYTE   0               ;TRACK 0
```

```
7688  024746  000000                        0                       ;CYLINDER 0
7689
7690  024750  104417                        TLOADRK                 ;LOAD RK REGS
7691  024752  104436                        TWAT2S                  ;WAIT FOR INTERRUPT
7692  024754  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7693
7694  024756  104422                        TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7695  024760  000000                        0
7696  024762  000000                        0
7697  024764  000002                        CTOERR                  ;CONTROLLER TIME OUT
7698  024766  104004                        ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
7699  024770  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
7700  024772  104003                        ERROR   3               ;BAD INIT ERROR
7701  024774  005037  001662                CLR     INTSET          ;CLEAR INT FLAG
7702  025000  012762  000100  000000        MOV     #IE,RKCS1(R2)   ;SET INT ENABLE
7703  025006  104437                        TWAT8S                  ;WAIT FOR SECOND INT
7704  025010  104002                        ERROR   2
7705
7706
7707
7708                                  .SBTTL  **ERRORS DURING DATA TRANSFER
7709
7710
7711                                  ;;*****************************************************************
7712                                  ;*TEST 76       LIMIT DETECT ON DATA TRANSFER
7713                                  ;*
7714                                  ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.  ISSUE
7715                                  ;*      A SEEK TO CYLINDER 2 WITH BAD PARITY.  ISSUE A DRIVE
7716                                  ;*      CLEAR.  ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
7717                                  ;*      TRACK 0, HEAD 0.  SEEK INCOMPLETE BECAUSE OF OUTER
7718                                  ;;*     LIMIT SHOULD BE THE ONLY ERROR SET.
7719                                  ;;*****************************************************************
7720  025012  000004                  TST76:  SCOPE
7721  025014  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
7722
7723  025022  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
7724  025024  104003                        ERROR   3               ;BAD INIT ERROR
7725
7726  025026  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
7727  025032  000113                        RECAL                   ;RECAL
7728  025034  000000                        0                       ;0 WORDS
7729  025036  000000                        0                       ;0 IS BUFF ADDRESS
7730  025040     000                        .BYTE   0               ;SECTOR 0
7731  025041     000                        .BYTE   0               ;TRACK 0
7732  025042  000000                        0                       ;CYLINDER 0
7733
7734  025044  104417                        TLOADRK                 ;LOAD RK REGS
7735  025046  104423                        TWAT16                  ;WAIT FOR INTERRUPT
7736  025050  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7737
7738  025052  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG
7739  025056  104437                        TWAT8S                  ;WAIT FOR SECOND INTERRUPT
7740  025060  104002                        ERROR   2
7741
7742  025062  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7743  025064  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
```

```
      025066  004437  034574                    JSR     R4,LRLOAD          ;LOAD "L" REGS
      025072  000117                            SEEK                       ;SEEK
      025074  000000                            0                          ;0 WORDS
      025076  000000                            0                          ;0 IS BUFF ADDRESS
      025100     000                            .BYTE   0                  ;SECTOR 0
      025101     000                            .BYTE   0                  ;TRACK 0
      025102  000002                            2                          ;CYLINDER 2
      025104  012737  000020  001616            MOV     #PAT,L.MR1         ;SET EVEN PARITY BIT
      025112  104417                            TLOADRK                    ;LOAD RK REGS
      025114  104423                            TWAT16                     ;WAIT FOR INTERRUPT
      025116  104002                            ERROR   2                  ;TO SLOW/NOT COMPLETE ERROR

      025120  104416                            TSSINIT                    ;CLEAR SUBSYSTEM
      025122  104003                            ERROR   3                  ;BAD INIT ERROR

      025124  004437  034574                    JSR     R4,LRLOAD          ;LOAD "L" REGS
      025130  000123                            WRDATA                     ;WRDATA
      025132  177400                            -400                       ;-400 WORDS
      025134  062604                            OBUFF                      ;OBUFF IS BUFF ADDRESS
      025136     000                            .BYTE   0                  ;SECTOR 0
      025137     000                            .BYTE   0                  ;TRACK 0
      025140  000001                            1                          ;CYLINDER 1

      025142  104417                            TLOADRK                    ;LOAD RK REGS
      025144  104423                            TWAT16                     ;WAIT FOR INTERRUPT
      025146  104002                            ERROR   2                  ;TO SLOW/NOT COMPLETE ERROR

      025150  104422                            TCHKWE                     ;CHECK OPERATION WITH ERROR
      025152  000002                            SKIERR                     ;SEEK INCOMPLETE
      025154  000000                            0
      025156  000000                            0
      025160  104004                            ERROR   4  ;OR 5,6,OR7     ;REPORT ALL DISCREPANCIES

                                        ;       THIS DELAY IS NEEDED TO ALLOW SEEK AND NO MOTION TO
                                        ;       SET IN THE DRIVE BEFORE THE SUBSYSTEM CLEAR IS DONE.

      025162  012701  025000                    MOV     #25000,R1          ;SET A COUNT
      025166  005301                    3$:     DEC     R1                 ;DEC COUNT
      025170  001376                            BNE     3$                 ;LOOP UNTIL ZERO

      025172  104416                            TSSINIT                    ;CLEAR SUBSYSTEM
      025174  104003                            ERROR   3                  ;BAD INIT ERROR

      025176  012762  000100  000000            MOV     #IE,RKCS1(R2)      ;SET IE

      025204  005037  001662                    CLR     INTSET             ;CLEAR INT FLAG
      025210  104437                            TWAT85                     ;WAIT FOR SECOND INTERRUPT
      025212  000401                            BR      1$
      025214  000404                            BR      2$

      025216  012737  057175  001372    1$:     MOV     #DH016,DH2N        ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
                                                                           ;ALLOWING HEADS TO RELOAD"
      025224  104002                            ERROR   2

      025226                            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.
                                ;*****************************************************************
025226  000004                  ST77:   SCOPE
025230  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
025236  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
025240  104003                        ERROR   3               ;BAD INIT ERROR

025242  004437  034574                JSR     R4,_RLOAD       ;LOAD "L" REGS
025246  000121                        RDDATA                  ;RDDATA
025250  177400                        -400                    ;-400 WORDS
025252  060604                        IBUFF                   ;IBUFF IS BUFF ADDRESS
025254     000                        .BYTE   0               ;SECTOR 0
025255     000                        .BYTE   0               ;TRACK 0
025256  000000                        0                       ;CYLINDER 0

025260  104417                        TLOADRK         ;LOAD RK REGS

025262  012762  000001  000022        MOV     #1,RKSPAR(R2)   ;WRITE SPARE REGISTER

025270  104423                        TWAIT6          ;WAIT FOR INTERRUPT
025272  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

025274  104422                        TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
025276  000000                        0
025300  000000                        0
025302  000020                        PGERR                   ;PROG ERROR
025304  104004                        ERROR   4  ;OR 5,6,7    ;REPORT ALL DISCREPANCIES


                                ;;*****************************************************************
                                ;*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.
                                ;*****************************************************************
025306  000004                  ST100:  SCOPE
025310  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
025316  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
025320  104003                        ERROR   3               ;BAD INIT ERROR

025322  004437  040776                JSR     R4,GENCOM       ;GENERATE DATA OF ALL ONES
025326  000007                        7
025330  000400                        400
```

```
 7856  025332  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 7857  025336  000123                  WRDATA                  ;WRDATA
 7858  025340  177400                  -400                    ;-400 WORDS
 7859  025342  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
 7860  025344     000                  .BYTE   0               ;SECTOR 0
 7861  025345     000                  .BYTE   0               ;TRACK 0
 7862  025346  000000                  0                       ;CYLINDER 0
 7863
 7864  025350  104417                  TLOADRK                 ;LOAD RK REGS
 7865  025352  104430                  TWAT96                  ;WAIT FOR INTERRUPT
 7866  025354  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7867
 7868  025356  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7869  025360  104004                  ERROR   4  ;OR 5, 6, 7. 10 ;REPORT ALL ERRORS
 7870
 7871  025362  004437  040776          JSR     R4,GENCOM       ;GENERATE DATA OF ZEROS
 7872  025366  000001                  1
 7873  025370  000400                  400
 7874
 7875  025372  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 7876  025376  000123                  WRDATA                  ;WRDATA
 7877  025400  177630                  -150                    ;-150 WORDS
 7878  025402  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
 7879  025404     000                  .BYTE   0               ;SECTOR 0
 7880  025405     000                  .BYTE   0               ;TRACK 0
 7881  025406  000000                  0                       ;CYLINDER 0
 7882
 7883  025410  104417                  TLOADRK                 ;START OPERATION
 7884
 7885  025412  005737  001662  1$:     TST     INTSET          ;CHECK IF INTERRUPT HAS OCCURRED
 7886  025416  001026                  BNE     2$              ;YES - MUCH TO SOON. REPORT ERROR .
 7887  025420  005762  000002          TST     RKWC,R2)        ;TEST IF NPR'S DONE
 7888  025424  001372                  BNE     1$              ;NO - LOOP
 7889
 7890  025426  052762  100000  000000  BIS     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER (CROWBAR WRITE)
 7891
 7892  025434  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 7893  025440  000121                  RDDATA                  ;RDDATA
 7894  025442  177400                  -400                    ;-400 WORDS
 7895  025444  060604                  IBUFF                   ;IBUFF IS BUFF ADDRESS
 7896  025446     000                  .BYTE   0               ;SECTOR 0
 7897  025447     000                  .BYTE   0               ;TRACK 0
 7898  025450  000000                  0                       ;CYLINDER 0
 7899
 7900  025452  104417                  TLOADRK                 ;LOAD RK REGS
 7901  025454  104425                  TWAT48                  ;WAIT FOR INTERRUPT
 7902  025456  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7903
 7904  025460  104422                  TCHKWE                  ;CHECK OPERATION WITH ERROR
 7905  025462  000000                  0
 7906  025464  000003                  DCKERR!ECHERR           ;DATA CHECK AND ECC HARD
 7907  025466  000000                  0
 7908  025470  104004                  ERROR   4 ;OR 5,6,7     ;REPORT ALL DISCREPANCIES
 7909
 791C  025472  000402                  BR      3$              ;SKIP TO EXIT
 791:  025474                  2$:
```

```
                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
                3$:


                ;;*****************************************************************
                ;*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.
                ;;*****************************************************************
025500 000004   TST101: SCOPE
025502 012737 000012 001262     MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

025510 104416                   TSSINIT                 ;CLEAR SUBSYSTEM
025512 104003                   ERROR   3               ;BAD INIT ERROR

025514 004437 034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
025520 000117                   SEEK                    ;SEEK
025522 000000                   0                       ;0 WORDS
025524 000000                   0                       ;0 IS BUFF ADDRESS
025526    000                   .BYTE   0               ;SECTOR 0
025527    000                   .BYTE   0               ;TRACK 0
025530 000632                   632                     ;CYLINDER 632

025532 104417                   TLOADRK                 ;LOAD RK REGS
025534 104423                   TWAT16                  ;WAIT FOR INTERRUPT
025536 104002                   ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
025540 005037 001662            CLR     INTSET          ;CLEAR INT FLAG
025544 104430                   TWAT96                  ;WAIT FOR SECOND INTERRUPT
025546 104002                   ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

025550 004437 034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
025554 000113                   RECAL                   ;RECAL
025556 000000                   0                       ;0 WORDS
025560 000000                   0                       ;0 IS BUFF ADDRESS
025562    000                   .BYTE   0               ;SECTOR 0
025563    000                   .BYTE   0               ;TRACK 0
025564 000000                   0                       ;CYLINDER 0

025566 104417                   TLOADRK                 ;LOAD RK REGS
025570 104423                   TWAT16                  ;WAIT FOR INTERRUPT
025572 104002                   ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

025574 004437 034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
025600 000125                   RDHEAD                  ;RDHEAD
025602 000000                   0                       ;0 WORDS
025604 000000                   0                       ;0 IS BUFF ADDRESS
025606    000                   .BYTE   0               ;SECTOR 0
025607    000                   .BYTE   0               ;TRACK 0
025610 000000                   0                       ;CYLINDER 0
025612 012737 000040 001616     MOV     #DMD,L.MR1      ;SET DIAG MODE
```

```
7968  025620  104417                          TLOADRK                 ;LOAD RK REGS
7969
7970  025622  004437  035050                  JSR     R4,MCLOCK       ;CLOCK CONTROLLER THROUGH SEEK
7971  025626  001062                          1062                    ;AND CLEAR TO READ
7972
7973  025630  005062  000026                  CLR     RKMR1(R2)       ;RESET DIAG MODE, LET RD HDRS COMPLETE
7974
7975  025634  104424                          TWAT32                  ;WAIT FOR INTERRUPT
7976  025636  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7977  025640  104422                          TCHKWE                  ;CHECK OPERATION WITH EXP ERROR
7978  025642  010000                          DTERR                   ;DRIVE TIMING ERROR
7979  025644  000000                          0
7980  025646  000000                          0
7981  025650  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL DISCREPANCIES
7982
7983  025652                          1$:
7984  025652  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7985  025654  104003                          ERROR   3               ;BAD INIT ERROR
7986  025656  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
7987  025664  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
7988
7989  025670  104437                          TWATBS                  ;WAIT FOR INTERRUPT FOR END OF RECAL
7990  025672  104002                          ERROR   2
7991
7992  025674  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7993  025700  000105                          CLEAR                   ;CLEAR
7994  025702  000000                          0                       ;0 WORDS
7995  025704  000000                          0                       ;0 IS BUFF ADDRESS
7996  025706     000                          .BYTE   0               ;SECTOR 0
7997  025707     000                          .BYTE   0               ;TRACK 0
7998  025710  000000                          0                       ;CYLINDER 0
7999
8000  025712  104417                          TLOADRK                 ;LOAD RK REGS
8001  025714  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8002  025716  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8003  025720  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8004  025722  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8005
8006                                  .SBTTL  **ERROR FORCING IN DRIVE
8007
8008                                  ;;*****************************************************************
8009                                  ;*TEST 102      INITIALIZE CLEARING SACK
8010                                  ;*
8011                                  ;*      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
8012                                  ;*      DRIVE.  ISSUE A SUBSYSTEM CLEAR.  PUT CONTROLLER IN
8013                                  ;*      DIAGNOSTIC MODE.  ISSUE A SELECT COMMAND WITH
8014                                  ;*      MESSAGE ID = 3 AND DRIVE SELECTED = 0.  CLOCK THROUGH
8015                                  ;*      PHASE ADDRESS 6.  TURN OFF DIAGNOSTIC MODE.  MAKE
8016                                  ;*      SURE UNIT FIELD ERROR DOES NOT SET.
8017                                  ;*
8018                                  ;;*****************************************************************
8019  025724  000004                  TST102: SCOPE
8020  025726  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
8021  025734  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8022  025736  104003                          ERROR   3               ;BAD INIT ERROR
8023
```

```
8024  025740  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
8025  025744  000101                  SELDRV                  ;SELDRV
8026  025746  000000                  0                       ;0 WORDS
8027  025750  000000                  0                       ;0 IS BUFF ADDRESS
8028  025752     000                  .BYTE   0               ;SECTOR 0
8029  025753     000                  .BYTE   0               ;TRACK 0
8030  025754  000000                  0                       ;CYLINDER 0
8031
8032  025756  104417                  TLOADRK                 ;LOAD RK REGS
8033  025760  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8034  025762  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8035
8036  025764  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8037  025766  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8038
8039  025770  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
8040  025772  104003                  ERROR   3               ;BAD INIT ERROR
8041
8042  025774  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
8043  026000  000101                  SELDRV                  ;SELDRV
8044  026002  000000                  0                       ;0 WORDS
8045  026004  000000                  0                       ;0 IS BUFF ADDRESS
8046  026006     000                  .BYTE   0               ;SECTOR 0
8047  026007     000                  .BYTE   0               ;TRACK 0
8048  026010  000000                  0                       ;CYLINDER 0
8049  026012  012737  000043  001616  MOV     #3!DMD,L.MR1    ;SET DIAG MODE AND MESSAGE PAIR 3
8050  026020  005037  001610          CLR     L.CS2           ;SELECT DRIVE 0
8051
8052  026024  104417                  TLOADRK                 ;LOAD RK REGS
8053
8054  026026  004437  035050          JSR     R4,MCLOCK       ;CLOCK THROUGH PHASE ADDRESS 6
8055  026032  001027                  1027
8056
8057  026034  042762  000040  000026  BIC     #DMD,RKMR1(R2)  ;CLEAR MAINTENANCE MODE
8058
8059  026042  104424                  TWAT32                  ;WAIT FOR INTERRUPT
8060  026044  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE
8061
8062  026046  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8063  026050  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8064                          ;;*****************************************************************
8065                          ;*TEST 103      DRIVE OFF TRACK
8066                          ;*
8067                          ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.  ISSUE
8068                          ;*      OFFSET OF +1200 MICRO-INCHES.  PUT CONTROLLER IN DIAGNOSTIC
8069                          ;*      MODE.  ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
8070                          ;*      TRACK 0, SECTOR 0.  CLOCK THROUGH SEEK AND DRIVE CLEAR
8071                          ;*      MESSAGES.  TURN OFF DIAGNOSTIC MODE.  DRIVE OFF TRACK
8072                          ;*      SHOULD SET IN DRIVE.  REPEAT FOR ALL AVAILIABLE DRIVES.
8073                          ;*
8074                          ;;*****************************************************************
8075  026052  000004          TST103: SCOPE
8076  026054  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
8077  026062  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
8078  026064  104003                  ERROR   3               ;BAD INIT ERROR
8079
```

```
 8080  026066  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 8081  026072  000113                  RECAL                   ;RECAL
 8082  026074  000000                  0                       ;0 WORDS
 8083  026076  000000                  0                       ;0 IS BUFF ADDRESS
 8084  026100     000                  .BYTE   0               ;SECTOR 0
 8085  026101     000                  .BYTE   0               ;TRACK 0
 8086  026102  000000                  0                       ;CYLINDER 0
 8087
 8088  026104  104417                  TLOADRK                 ;LOAD RK REGS
 8089  026106  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8090  026110  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8091  026112  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
 8092
 8093  026116  104437                  TWAT3S                  ;WAIT FOR INTERRUPT #2
 8094  026120  104002                  ERROR   2
 8095
 8096  026122  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 8097  026126  000105                  CLEAR                   ;CLEAR
 8098  026130  000000                  0                       ;0 WORDS
 8099  026132  000000                  0                       ;0 IS BUFF ADDRESS
 8100  026134     000                  .BYTE   0               ;SECTOR 0
 8101  026135     000                  .BYTE   0               ;TRACK 0
 8102  026136  000000                  0                       ;CYLINDER 0
 8103
 8104  026140  104417                  TLOADRK                 ;LOAD RK REGS
 8105  026142  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8106  026144  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8107
 8108  026146  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8109  026150  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8110
 8111  026152  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
 8112  026156  000115                  OFFSET                  ;OFFSET
 8113  026160  000000                  0                       ;0 WORDS
 8114  026162  000000                  0                       ;0 IS BUFF ADDRESS
 8115  026164     000                  .BYTE   0               ;SECTOR 0
 8116  026165     000                  .BYTE   0               ;TRACK 0
 8117  026166  000000                  0                       ;CYLINDER 0
 8118  026170  112737  000060  001612  MOVB    #60,L.ASOF      ;SET OFFSET AT +1200
 8119
 8120  026176  104417                  TLOADRK                 ;LOAD RK REGS
 8121  026200  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8122  026202  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8123
 8124  026204  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8125  026206  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8126
 8127  026210  005037  001662          CLR     INTSET          ;CLEAR INT FLAG
 8128
 8129  026214  104424                  TWAT32                  ;WAIT FOR INT #2
 8130  026216  104002                  ERROR   2
 8131
 8132  026220  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8133  026222  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8134
 8135  026224  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
 8136  026230  000105                        CLEAR                   ;CLEAR
 8137  026232  000000                        0                       ;0 WORDS
 8138  026234  000000                        0                       ;0 IS BUFF ADDRESS
 8139  026236     000                        .BYTE   0               ;SECTOR 0
 8140  026237     000                        .BYTE   0               ;TRACK 0
 8141  026240  000000                        0                       ;CYLINDER 0
 8142
 8143  026242  104417                        TLOADRK                 ;LOAD RK REGS
 8144  026244  104423                        TWAT16                  ;WAIT FOR INTERRUPT
 8145  026246  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8146
 8147  026250  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8148  026252  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8149
 8150  026254  004437  034574                JSR     R4,LRLOAD       ;LOAD "L" REGS
 8151  026260  000123                        WRDATA                  ;WRDATA
 8152  026262  177777                        -1                      ;-1 WORDS
 8153  026264  062604                        OBUFF                   ;OBUFF IS BUFF ADDRESS
 8154  026266     000                        .BYTE   0               ;SECTOR 0
 8155  026267     000                        .BYTE   0               ;TRACK 0
 8156  026270  000000                        0                       ;CYLINDER 0
 8157  026272  012737  000040  001616        MOV     #DMD,L.MR1      ;SET DIAGNOSTIC MODE
 8158
 8159  026300  104417                        TLOADRK
 8160
 8161  026302  004437  035050                JSR     R4,MCLOCK       ;CLOCK THROUGH SEEK & DRIVE CLEAR
 8162  026306  001064                        1064
 8163
 8164  026310  005062  000026                CLR     RKMR1(R2)       ;CLEAR DIAGNOSTIC MODE
 8165  026314  104424                        TWAT32                  ;WAIT FOR INTERRUPT
 8166  026316  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE
 8167
 8168                                ;;**************************************************************
 8169                                ;;**************************************************************
 8170
 8171                                ;       N O T E:
 8172                                ;
 8173                                ;       THE DRIVE LOGIC DOES NOT RAISE ATTENTION AT THEOCCURRENCE
 8174                                ;       OF THE ERROR. IT FIRST RETURNS THE HEADS TOCENTERLINE
 8175                                ;       AND FIRES A HEADS SETTLING ONE-SHOT. ATTENTION IS NOT SET
 8176                                ;       UNTIL THE ONE-SHOT TIMES OUT. CONSEQUENTLY THE CONTROLLER
 8177                                ;       WILL FINISH THE WRITE, RAISE INTERRUPT AS THOUGH NO ERROR
 8178                                ;       OCCURRED, THE DRIVE ATTENTION WILL HAPPEN A LITTLE LATER
 8179                                ;       (ABOUT 3 MILISECONDS) AT WHICH TIME THE DRIVE MUST BE SELECTED
 8180                                ;       TO CHECK THAT DRIVE OFF TRACK SET.
 8181
 8182                                ;;**************************************************************
 8183                                ;;**************************************************************
 8184
 8185                                ;       THIS DRIVE LOGIC MAY CHANGE!!! IF IT DOES THIS
 8186                                ;       TEST MUST BE ALTERED.
 8187
 8188  026320  005037  001662                CLR     INTSET          ;CLEAR INT FLAG
 8189  026324  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8190  026326  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8191
```

```
8192  026330  104423                          TWAIT6                  ;WAIT FOR INTERRUPT FROM DRIVE
8193                                                                  ;THAT SIGNALSDRIVE OFF TRACK ERROR
8194  026332  104002                          ERROR   2
8195
8196  026334  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8197  026340  000101                          SELDRV                  ;SELDRV
8198  026342  000000                          0                       ;0 WORDS
8199  026344  000000                          0                       ;0 IS BUFF ADDRESS
8200  026346     000                          .BYTE   0               ;SECTOR 0
8201  026347     000                          .BYTE   0               ;TRACK 0
8202  026350  000000                          0                       ;CYLINDER 0
8203  026352  005737  001616                  CLR     L.MR1           ;RESET DIAG MODE
8204
8205  026356  104417                          TLOADRK                 ;LOAD RK REGS
8206  026360  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
8207  026362  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8208
8209  026364  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR EXPECTED
8210  026366  000400                          DROTERR                 ;DRIVE OFF TRACK
8211  026370  000000                          0
8212  026372  000000                          0
8213  026374  104004                          ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
8214                                ;;*****************************************************************
8215                                ;*TEST 104      FILE UNSAFE
8216                                ;*
8217                                ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECLAIBRATE.  ISSUE
8218                                ;*      A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR
8219                                ;*      FORMAT.  DO A SELECT COMMAND IN 26 SECTOR FORMAT.
8220                                ;*      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
8221                                ;*      HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR
8222                                ;*      FORMAT.  CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.
8223                                ;*      SIMULATE INDEX PULSE.  TURN OFF DIAGNOSTIC MODE.  FILE
8224                                ;*      UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE
8225                                ;*      THROUGH SECTOR PULSE.  REPEAT FOR ALL AVAILIABLE DRIVES.
8226                                ;*
8227                                ;*****************************************************************
8228  026376  000004                TST104: SCOPE
8229  026400  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
8230  026406  012737  177777  001664          MOV     #-1,REFMT       ;SET REFORMAT SWITCH
8231  026414  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8232  026416  104003                          ERROR   3               ;BAD INIT ERROR
8233
8234  026420  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8235  026424  000113                          RECAL                   ;RECAL
8236  026426  000000                          0                       ;0 WORDS
8237  026430  000000                          0                       ;0 IS BUFF ADDRESS
8238  026432     000                          .BYTE   0               ;SECTOR 0
8239  026433     000                          .BYTE   0               ;TRACK 0
8240  026434  000000                          0                       ;CYLINDER 0
8241
8242  026436  104417                          TLOADRK                 ;LOAD RK REGS
8243  026440  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
8244  026442  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8245
8246  026444  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8247  026446  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
```

```
8248
8249  026450  005037  001662          CLR     INTSET          ;CLEAR INT FLAG
8250  026454  104437                  TWAT8S                  ;WAIT FOR SECOND INT
8251  026456  104002                  ERROR   2
8252
8253  026460  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8254  026462  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8255
8256  026464  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
8257  026470  000105                  CLEAR                   ;CLEAR
8258  026472  000000                  0                       ;0 WORDS
8259  026474  000000                  0                       ;0 IS BUFF ADDRESS
8260  026476     000                  .BYTE   0               ;SECTOR 0
8261  026477     000                  .BYTE   0               ;TRACK 0
8262  026500  000000                  0                       ;CYLINDER 0
8263
8264  026502  104417                  TLOADRK                 ;LOAD RK REGS
8265  026504  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8266  026506  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8267
8268  026510  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8269  026512  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8270
8271  026514  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
8272  026520  010125                  RDHEAD!CFMT                     ;RDHEAD!CFMT
8273  026522  000000                  0                       ;0 WORDS
8274  026524  000000                  0                       ;0 IS BUFF ADDRESS
8275  026526     000                  .BYTE   0               ;SECTOR 0
8276  026527     000                  .BYTE   0               ;TRACK 0
8277  026530  000000                  0                       ;CYLINDER 0
8278
8279  026532  104417                  TLOADRK                 ;LOAD RK REGS
8280  026534  104424                  TWAT32                  ;WAIT FOR INTERRUPT
8281  026536  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8282
8283  026540  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8284  026542  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8285
8286  026544  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
8287  026550  000101                  SELDRV                  ;SELDRV
8288  026552  000000                  0                       ;0 WORDS
8289  026554  000000                  0                       ;0 IS BUFF ADDRESS
8290  026556     000                  .BYTE   0               ;SECTOR 0
8291  026557     000                  .BYTE   0               ;TRACK 0
8292  026560  000000                  0                       ;CYLINDER 0
8293
8294  026562  104417                  TLOADRK                 ;LOAD RK REGS
8295  026564  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8296  026566  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8297
8298  026570  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
8299  026574  000127                  WRHEAD                  ;WRHEAD
8300  026576  177777                  -1                      ;-1 WORDS
8301  026600  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
8302  026602     000                  .BYTE   0               ;SECTOR 0
8303  026603     000                  .BYTE   0               ;TRACK 0
```

```
8304  026604  000000                    0                       ;CYLINDER 0
8305  026606  012737  000040  001616    MOV     #DMD.L.MR1      ;SET DIAGNOSTIC-MODE
8306
8307  026614  104417                    TLOADRK                 ;LOAD RK REGS
8308  026616  004437  035050            JSR     R4,MCLOCK       ;CLOCK THROUGH SEEK AND DRIVE CLEAR
8309  026622  001064                    1064
8310
8311  026624  052762  000200  000026    BIS     #MIND,RKMR1(R2) ;SET INDEX
8312
8313  026632  004437  035050            JSR     R4,MCLOCK       ;CLOCK INDEX
8314  026636  001001                    1001
8315
8316  026640  042762  000200  000026    BIC     #MIND,RKMR1(R2) ;CLEAR INDEX
8317
8318  026646  004437  035050            JSR     R4,MCLOCK       ;CLOCK CLEAR
8319  026652  001001                    1001
8320
8321  026654  005062  000026            CLR     RKMR1(R2)       ;CLEAR DIAGNOSTIC MODE
8322
8323  026660  104426                    TWAT64                  ;WAIT FOR INTERRUPT
8324  026662  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8325
8326  026664  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8327  026666  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
8328
8329  026670  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
8330  026674  000101                    SELDRV                  ;SELDRV
8331  026676  000000                    0                       ;0 WORDS
8332  026700  000000                    0                       ;0 IS BUFF ADDRESS
8333  026702     000                    .BYTE   0               ;SECTOR 0
8334  026703     000                    .BYTE   0               ;TRACK 0
8335  026704  000000                    0                       ;CYLINDER 0
8336
8337  026706  005037  001616            CLR     L.MR1           ;CLEAR DIAG MODE
8338
8339  026712  104417                    TLOADRK                 ;LOAD RK REGS
8340  026714  104423                    TWAT16                  ;WAIT FOR INTERRUPT
8341  026716  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8342
8343  026720  104422                    TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
8344  026722  040400                    UNSERR!DROTERR          ;UNSAFE AND DRIVE OFF TRACK
8345  026724  000000                    0
8346  026726  000000                    0
8347  026730  104004                    ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
8348
8349  026732  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
8350  026734  104003                    ERROR   3               ;BAD INIT ERROR
8351
8352  026736  004437  034574            JSR     R4,LRLOAD       ;LOAD "L" REGS
8353  026742  000101                    SELDRV                  ;SELDRV
8354  026744  000000                    0                       ;0 WORDS
8355  026746  000000                    0                       ;0 IS BUFF ADDRESS
8356  026750     000                    .BYTE   0               ;SECTOR 0
8357  026751     000                    .BYTE   0               ;TRACK 0
8358  026752  000000                    0                       ;CYLINDER 0
8359
```

```
8360  026754  012737  000001  001616          MOV     #1,L.MR1        ;SET MESSAGE SELECT ONE
8361
8362  026762                          1$:
8363  026762  104417                          TLOADRK                 ;LOAD RK REGS
8364  026764  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8365  026766  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8366
8367  026770  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8368  026772  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8369
8370  026774  032737  000040  001574          BIT     #S.HDHM,T.MR2   ;TEST IF HEADS HOME
8371  027002  001767                          BEQ     1$
8372
8373  027004  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8374  027006  104003                          ERROR   3               ;BAD INIT ERROR
8375
8376  027010  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8377  027014  104434                          TWAT159                 ;WAIT FOR APPROX 160 MS
8378  027016  000240                          NOP                     ;DON'T CARE ERROR RETURN
8379
8380  027020  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8381  027022  104003                          ERROR   3               ;BAD INIT ERROR
8382
8383  027024  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
8384
8385  027032  104437                          TWAT8S                  ;WAIT FOR SECOND INTERRUPT
8386  027034  104002                          ERROR   2
8387
8388  027036  005037  001616                  CLR     L.MR1           ;CLEAR MR1
8389
8390  027042  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8391  027046  000105                          CLEAR                   ;CLEAR
8392  027050  000000                          0                       ;0 WORDS
8393  027052  000000                          0                       ;0 IS BUFF ADDRESS
8394  027054     000                          .BYTE   0               ;SECTOR 0
8395  027055     000                          .BYTE   0               ;TRACK 0
8396  027056  000000                          0                       ;CYLINDER 0
8397
8398  027060  104417                          TLOADRK                 ;LOAD RK REGS
8399  027062  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8400  027064  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8401
8402  027066  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8403  027070  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8404
8405  027072  004437  040776                  JSR     R4,GENCOM       ;BUILD HEADERS
8406  027076  001200                          1200
8407
8408  027100  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8409  027104  000127                          WRHEAD                  ;WRHEAD
8410  027106  177676                          -102                    ;-102 WORDS
8411  027110  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
8412  027112     000                          .BYTE   0               ;SECTOR 0
8413  027113     000                          .BYTE   0               ;TRACK 0
8414  027114  000000                          0                       ;CYLINDER 0
8415
```

```
8416  027116  104417                          TLOADRK                 ;LOAD RK REGS
8417  027120  104426                          TWAT64                  ;WAIT FOR INTERRUPT
8418  027122  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8419
8420  027124  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8421  027126  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8422
8423                                  ;;*****************************************************************
8424                                  ;*TEST 105      DUMMY TEST FOR PREVIOUS TEST EXIT
8425                                  ;*      THIS TEST IS PRESENT TO MAKE $SW08TB TABLE HAVE AN ENTRY
8426                                  ;*      WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS
8427                                  ;*      IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST.
8428                                  ;*      IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE
8429                                  ;*      "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".
8430                                  ;*
8431                                  ;*      IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED
8432                                  ;*      TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS
8433                                  ;*      FROM THE DRIVE COMING READY AT A LATER TIME.
8434                                  ;;*****************************************************************
8435  027130  000004                  TST105: SCOPE
8436  027132  012737  000001  001262          MOV     #1,$TIMES       ;;DO 1 ITERATION
8437
8438  027140  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8439  027142  104003                          ERROR   3               ;BAD INIT ERROR
8440
8441  027144  013762  001626  000010          MOV     DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
8442  027152  012762  000001  000000          MOV     #1,RKCS1(R2)    ;SELECT THE DRIVE
8443  027160  032762  000200  000012  1$:     BIT     #DRDY,RKDS(R2)  ;TEST IF DRIVE READY
8444  027166  001774                          BEQ     1$              ;NO LOOP
8445
8446  027170  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8447  027172  104003                          ERROR   3               ;BAD INIT ERROR
8448
8449  027174  004437  040776                  JSR     R4,GENCOM       ;GENERATE HEADERS FOR CYL 0
8450  027200  001200                          1200
8451
8452  027202  004437  034574                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8453  027206  000127                          WRHEAD                  ;WRHEAD
8454  027210  177676                          -102                    ;-102 WORDS
8455  027212  062604                          OBUFF                   ;OBUFF IS BUFF ADDRESS
8456  027214     000                          .BYTE   0               ;SECTOR 0
8457  027215     000                          .BYTE   0               ;TRACK 0
8458  027216  000000                          0                       ;CYLINDER 0
8459
8460  027220  104417                          TLOADRK                 ;LOAD RK REGS
8461  027222  104426                          TWAT64                  ;WAIT FOR INTERRUPT
8462  027224  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8463
8464  027226  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8465  027230  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8466  027232  005037  001664                  CLR     REFMT           ;CLEAR REFORMAT SWITCH
```

```
                                        .SBTTL  **MULTI-DRIVE OPERATIONS
027236 000004                           NEWDRV: SCOPE
027240 012737 000001 001262                     MOV     #1,$TIMES       ;DO ONLY ONCE
027246 032737 000200 001630                     BIT     #BIT7,DRVBIT    ;WERE WE TESTING DRIVE 7?
027254 001022                                   BNE     3$              ;YES-SKIP
                                        
027256 005237 001626                    1$:     INC     DRVNUM          ;BUMP TO NEXT SEQUENTIAL ADDRESS
027262 006337 001630                            ASL     DRVBIT          ;BUMP DRIVEBIT TO THAT POSITION
027266 033737 001630 001354                     BIT     DRVBIT,$DEVM    ;IS THIS DRIVE TO BE TESTED?
027274 001005                                   BNE     2$              ;YES-EXIT
027276 032737 000400 001630                     BIT     #BIT8,DRVBIT    ;ALL DRIVES TESTED?
027304 001006                                   BNE     3$              ;YES-EXIT
027306 000763                                   BR      1$              ;ELSE CHECK NEXT DRIVE AVAILABLE
                                        
027310 112737 000004 001102             2$:     MOVB    #4,$TSTNM       ;SET TEST NUMBER FOR REPORTS
027316 000137 004450                            JMP     @#ST$LUP        ;GO TO TEST LOOP TO CHECK THIS DRIVE
027322 005037 001630                    3$:     CLR     DRVBIT          ;CLEAR DRIVE BIT
027326 005037 001626                            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.
                                        ;*
                                        ;;*****************************************************************
027332 000004                           ST106: SCOPE
027334 012737 000012 001262                     MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
027342 005000                                   CLR     R0              ;CLEAR DRIVE POSITION COUNTER
027344 012701 000001                            MOV     #1,R1           ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
027350 013703 001354                            MOV     $DEVM,R3        ;GET DEVICE MAP
027354 104416                                   TSSINIT                 ;CLEAR SUBSYSTEM
027356 104003                                   ERROR   3               ;BAD INIT ERROR
027360 030103                           1$:     BIT     R1,R3           ;TEST IF THIS DRIVE AVAILABLE
027362 001006                                   BNE     2$              ;YES-SKIP TO SEEK
027364 006301                           3$:     ASL     R1              ;SHIFT DRIVE SELECT BIT
027366 005200                                   INC     R0              ;BUMP DRIVE POSITION COUNTER
027370 032701 000400                            BIT     #BIT8,R1        ;ALL DRIVE POSITIONS CHECKED
027374 001771                                   BEQ     1$              ;NO-LOOP
027376 000441                                   BR      4$              ;SKIP TO RESET
                                        
027400 010037 001610                    2$:     MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
027404 012737 000113 001600                     MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
                                        
027412 104417                                   TLOADRK                 ;LOAD RK REGS
027414 104423                                   TWAIT6                  ;WAIT FOR INTERRUPT
027416 104002                                   ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                        
027420 005037 001662                            CLR     INTSET          ;CLEAR INTERRUPT FLAG
027424 012705 000764                            MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
027430 012762 000012 000000                     MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
027436 016237 000016 001556             12$:    MOV     RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
027444 113704 001557                            MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
027450 042704 177400                            BIC     #177400,R4      ;CLEAR UNUSED BITS
027454 030104                                   BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
027456 001006                                   BNE     10$             ;YES - SKIP
```

```
                                        TWAIT16                 ;WAIT FOR 16 MS
                                        NOP                     ;DON'T CARE RETURNS
                                        NOP
                                        DEC     R5              ;TOTAL WAIT TIME IS 8 SECONDS
                                        BNE     12$             ;CHECK ATTENTION EACH 16 MS
                                        ERROR   2               ;REPORT IF NO ATTENTION IN 8 SEC

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

                                        BR      3$              ;LOOP FOR NEXT DRIVE
                                4$:     RESET                   ;UNIBUS RESET
                                        JSR     PC,$TKINT       ;RESET KEYBOARD INTERRUPT ENABLE

                                        MOV     #25.,R1         ;DO A SHORT DELAY
                                5$:     DEC     R1
                                        BNE     5$
                                        JSR     PC,OPTTST       ;SET UP OPTIONS

                                        TGETRK                  ;GET RK611 REGS

                                        TSTB    T.ASOF+1        ;ALL ATTENTION RESET?
                                        BEQ     6$              ;YES-SKIP

                                        MOV     #EMDA,EM12N     ;"DRIVE ATT NOT RESET RESULT OF
                                        MOV     #EMUR,DF011A    ;UNIBUS RESET"
                                        ERROR   12

                                6$:
                                ;;*****************************************************************
                                ;*TEST 107      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
                                ;*
                                ;*      DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
                                ;*      ISSUE A SUBSYSTEM CLEAR.  MAKE SURE ALL ATTENTIONS
                                ;*      RESET.
                                ;*
                                ;;*****************************************************************
                                TST107: SCOPE
                                        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
                                        CLR     R0              ;CLEAR DRIVE POSITION COUNTER
                                        MOV     #1,R1           ;PRESET TO TEST POSITION 0
                                        MOV     $DEVM,R3        ;CUT DEVICE MAP
                                        TSSINIT                 ;CLEAR SUBSYSTEM
                                        ERROR   3               ;BAD INIT ERROR
                                1$:     BIT     R1,R3           ;THIS DRIVE AVAILABLE?
                                        BNE     2$              ;YES-SKIP TO SEEK
                                3$:     ASL     R1              ;SHIFT TO NEXT DRIVE POSITION
                                        INC     R0              ;DUMP POSITION COUNTER
                                        BIT     #BIT8,R1        ;ALL POSITIONS CHECKED
                                        BEQ     1$              ;NO-LOOP
                                        BR      4$              ;YES-SKIP TO CLEAR

                                2$:     MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
                                        MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
```

```
8579  027632  104417                          TLOADRK                 ;LOAD RK REGS
8580  027634  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
8581  027636  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8582
8583  027640  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8584  027644  012705  000764                  MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
8585  027650  012762  000012  000000          MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
8586  027656  016237  000016  001556  12$:    MOV     RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8587  027664  113704  001557                  MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
8588  027670  042704  177400                  BIC     #177400,R4      ;CLEAR UNUSED BITS
8589  027674  030104                          BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8590  027676  001006                          BNE     10$             ;YES - SKIP
8591
8592  027700  104423                          TWAIT6                  ;WAIT FOR 16 MS
8593  027702  000240                          NOP                     ;DON'T CARE RETURNS
8594  027704  000240                          NOP
8595  027706  005305                          DEC     R5              ;TATOL WAIT TIME IS 8 SECONDS
8596  027710  001362                          BNE     12$             ;CHECK ATTENTION EACH 16 MS
8597  027712  104002                          ERROR   2               ;REPORT IF NO ATTENTION IN 8 SEC
8598
8599  027714                          10$:
8600  027714  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8601  027716  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8602
8603  027720  000731                          BR      3$              ;LOOP FOR NEXT DRIVE
8604
8605  027722  052762  000040  000010  4$:     BIS     #SCLR,RKCS2(R2) ;DO SUBSYSTEM CLEAR
8606  027730  012701  000031                  MOV     #25.,R1         ;DO A SHORT DELAY
8607  027734  005301                  5$:     DEC     R1
8608  027736  001376                          BNE     5$
8609
8610  027740  104420                          TGETRK                  ;GET RK611 REGS
8611
8612  027742  105737  001557                  TSTB    T.ASOF+1        ;TEST ALL ATTENTION RESET
8613  027746  001407                          BEQ     6$              ;YES-SKIP
8614
8615  027750  012737  055421  001470          MOV     #EMDA,EM12N     ;"DRIVE ATT NOT RESET AS RESULT OF
8616  027756  012737  055635  057702          MOV     #EMSCLR,DF011A  ;SUBSYSTEM CLEAR"
8617  027764  104012                          ERROR   12
8618
8619  027766                          6$:
8620                                  ;;*****************************************************************
8621                                  ;*TEST 110      SVAL AND ATTENTION FROM OTHER DRIVE
8622                                  ;*
8623                                  ;*      DO A RECALIBRATE ON ONE AVAILABLE DRIVE.  DO A SELECT
8624                                  ;*      ON ANOTHER AVAILABLE DRIVE.  MAKE SURE STATUS VALID
8625                                  ;*      IS SET.  WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
8626                                  ;*      MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
8627                                  ;*      CHANGE REMAINS RESET.
8628                                  ;*
8629                                  ;*      REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.
8630                                  ;*
8631                                  ;*      NOTE:   THIS TEST WILL ONLY BE DONE IF AT LEAST
8632                                  ;*              TWO DRIVES ARE AVAILABLE.
8633                                  ;*
8634                                  ;;*****************************************************************
```

```
027766 000004                          TST110: SCOPE
027770 012737  000024  001262                  MOV     #20.,$TIMES         ;;DO 20. ITERATIONS
027776 013746  001354                          MOV     $DEVM,-(SP)         ;PUT DEVICE MAP ON STACK
030002 004437  034774                          JSR     R4,BITCNT           ;COUNT NUMBER OF BITS(# OF DRIVES)
030006 022627  000001                          CMP     (SP)+,#1            ;COMPARE TO 1
030012 101007                                  BHI     2$                  ;SKIP IF MORE THAN 1
030014 005737  001304                          TST     $PASS               ;CHECK IF PASS 0
030020 001002                                  BNE     1$                  ;NO-SKIP

030022 104401  051477                          TYPE    ,OPR018             ;"OVERLAPPED OPERATION BYPASSED"
030026 000137  031242                  1$:     JMP     $EOP                ;GET OUT.

030032 012737  177777  001224          2$:     MOV     #-1,$TMP1           ;SET LOOP CONTROL FLAG
030040 013705  001354                          MOV     $DEVM,R5            ;GET DEVICE MAP
030044 005000                                  CLR     R0                  ;CLEAR FOR DRIVE #A
030046 005001                                  CLR     R1                  ;CLEAR FOR DRIVE #B
030050 012703  000001                          MOV     #1,R3               ;SET DRIVE POSITION A
030054 012704  000001                          MOV     #1,R4               ;SET DRIVE POSITION B
030060 012737  030070  001110                  MOV     #3$,$LPERR          ;SET LOCAL LOOP ON ERROR
030066 000477                                  BR      11$                 ;GO SET UP POINTERS

030070                                 3$:
030070 104416                                  TSSINIT                     ;CLEAR SUBSYSTEM
030072 104003                                  ERROR   3                   ;BAD INIT ERROR

030074 010037  001610                          MOV     R0,L.CS2            ;LOAD DRIVE A
030100 010037  001626                          MOV     R0,DRVNUM           ;LOAD FOR REPORT
030104 012737  000113  001600                  MOV     #RECAL,L.CS1        ;LOAD RECAL

030112 104417                                  TLOADRK                     ;LOAD RK REGS
030114 104423                                  TWAT16                      ;WAIT FOR INTERRUPT
030116 104002                                  ERROR   2                   ;TO SLOW/NOT COMPLETE ERROR
030120 104421                                  TCHKOP                      ;CHECK OPERATION FOR ANY ERRORS
030122 104004                                  ERROR   4  ;OR 5, 6, 7      ;REPORT ALL ERRORS

030124 005037  001662                          CLR     INTSET              ;CLEAR INT FLAG

030130 010137  001610                          MOV     R1,L.CS2            ;LOAD DRIVE B
030134 010137  001626                          MOV     R1,DRVNUM           ;LOAD FOR REPORT
030140 012737  000101  001600                  MOV     #SELDRV,L.CS1       ;LOAD DRIVE SELECT

030146 104417                                  TLOADRK                     ;LOAD RK REGS
030150 104423                                  TWAT16                      ;WAIT FOR INTERRUPT
030152 104002                                  ERROR   2                   ;TO SLOW/NOT COMPLETE ERROR

030154 104421                                  TCHKOP                      ;CHECK OPERATION FOR ANY ERRORS
030156 104004                                  ERROR   4  ;OR 5, 6, 7      ;REPORT ALL ERRORS

030160 032737  100000  001552                  BIT     #SVAL,T.DS          ;CHECK IF STATUS VALID SET
030166 001007                                  BNE     4$                  ;YES - SKIP
030170 012737  055235  001460                  MOV     #EMSVAL,EM11N       ;"STATUS VALID NOT SET RESULT OF
030176 012737  046676  057702                  MOV     #OPER00,DF011A      ;DRIVE SELECT"
030204 104011                                  ERROR   11

030206 005037  001662                  4$:     CLR     INTSET              ;CLEAR INT FLAG
030212 104436                                  TWAT2S                      ;WAIT FOR SEEK COMPLETE INTERRUPT
```

```
8691  030214  000401                    BR      44$             ;NONE RECEIVED - SKIP
8692  030216  000406                    BR      55$             ; RECEIVED - SKIP
8693
8694  030220  010037  001626  44$:      MOV     R0,DRVNUM       ;SET DRIVE FOR REPORT
8695  030224  012737  057272  001372    MOV     #DH017,DH2N     ;"COMMAND - SELECT AFTER RECAL"
8696  030232  104002                    ERROR   2
8697
8698  030234  104420  55$:      TGETRK                          ;GET RK REGS
8699  030236  032737  100000  001552    BIT     #SVAL,T.DS      ;TEST IF SVAL STILL SET
8700  030244  001007                    BNE     5$              ;YES - SKIP
8701
8702  030246  012737  055235  001510    MOV     #EMSVAL,EM14N   ;"STATUS VALID RESET RESULT OF
8703  030254  012737  057315  057702    MOV     #DH018,DF011A   ;RECAL COMPLETE ATTENTION AFTER SEL"
8704  030262  104014                    ERROR   14
8705
8706  030264  104415  5$:       SCOP1                           ;LOCAL LOOP TO 3$
8707
8708                    ;       THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
8709                    ;       OF DRIVES AVAILABLE.  THE FIRST PASS OF THE PROGRAM WILL USE THE
8710                    ;       LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
8711                    ;       B.  THE SECOND PASS SWAPS DRIVE A & B.  THE THIRD PASS USES
8712                    ;       THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
8713                    ;       AS A.  THE FORTH PASS SWAPS A & B AGAIN.  THIS CONTINUES
8714                    ;       UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
8715                    ;       DRIVE.
8716                    ;
8717                    ;       THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
8718                    ;       THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
8719                    ;
8720                    ;       THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
8721                    ;       CHECKED.
8722
8723  030266  005237  001224  11$:      INC     $TMP1           ;INCREMENT PASS CONTROL
8724  030272  001024                    BNE     16$             ;SKIP IF NOT ZERO
8725                                                            ;(IT WILL BE ZERO ON THE 1ST PASS)
8726
8727  030274  030305  12$:       BIT     R3,R5           ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8728  030276  001006                    BNE     13$             ;YES-SKIP
8729
8730  030300  005200  22$:       INC     R0              ;BUMP R0 (DRIVE A)
8731  030302  006303                    ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
8732  030304  032703  000400            BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
8733  030310  001771                    BEQ     12$             ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8734  030312  000464                    BR      50$             ;DONE-EXIT
8735
8736  030314  010001  13$:       MOV     R0,R1           ;SET DRIVE B TO THE SAME AS A
8737  030316  010304                    MOV     R3,R4
8738  030320  005201  14$:       INC     R1              ;BUMP R1 (DRIVE B)
8739  030322  006304                    ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
8740  030324  030405                    BIT     R4,R5           ;IS THIS DRIVE AVAIL?
8741  030326  001004                    BNE     15$             ;YES-SKIP
8742  030330  032704  000400            BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
8743  030334  001771                    BEQ     14$             ;NO-LOOP
8744  030336  000452                    BR      50$             ;DONE-EXIT
8745
8746  030340  000137  030070  15$:      JMP     3$              ;GO DO THE TEST ON THE DRIVE A & B
```

```
8747                                                    ;CONTAINED IN R0 & R1
8748  030344  032737  000001  001224  16$:    BIT     #BIT0,$TMP1     ;IS PASS FLAGS ODD?
8749  030352  001410                       BEQ     17$             ;NO-SKIP
8750
8751  030354  010046                       MOV     R0,-(SP)        ;
8752  030356  010346                       MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
8753  030360  010403                       MOV     R4,R3           ;TO EXCHANGE DRIVE A & B
8754  030362  010100                       MOV     R1,R0
8755  030364  012604                       MOV     (SP)+,R4
8756  030366  012601                       MOV     (SP)+,R1
8757  030370  000137  030070               JMP     3$              ;REPEAT TEST ON THIS COMBO.
8758
8759  030374  032737  000002  001224  17$:    BIT     #BIT1,$TMP1     ;TEST IF PASS FLAGS AT HALF MODULE 4?
8760  030402  001410                       BEQ     19$             ;NO-SKIP TO BUMP DRIVE B
8761  030404  005200                 18$:  INC     R0              ;BUMP DRIVE A
8762  030406  006303                       ASL     R3              ;SHIFT DRIVE SELECT BIT
8763  030410  030305                       BIT     R3,R5           ;AVAILABLE?
8764  030412  001014                       BNE     20$             ;YES-SKIP
8765  030414  032703  000400               BIT     #BIT8,R3        ;ALL CHECKED?
8766  030420  001771                       BEQ     18$             ;NO-SKIP
8767  030422  000412                       BR      21$             ;GO TO NEXT PASS
8768
8769  030424  005201                 19$:  INC     R1              ;BUMP DRIVE B
8770  030426  006304                       ASL     R4              ;SHIFT DRIVE SELECT BIT
8771  030430  030405                       BIT     R4,R5           ;AVAILABLE?
8772  030432  001004                       BNE     20$             ;YES-SKIP
8773  030434  032704  000400               BIT     #BIT8,R4        ;ALL CHECKED?
8774  030440  001771                       BEQ     19$             ;NO-LOOP
8775  030442  000404                       BR      23$             ;YES-SKIP TO NEXT PASS
8776
8777  030444  000137  030070         20$:  JMP     3$              ;GO TEST THIS COMBO
8778
8779  030450  010100                 21$:  MOV     R1,R0           ;SET DRIVE 0 TO LOW POSITION THIS PASS
8780  030452  010403                       MOV     R4,R3           ;SET SELECT BITS TO AGREE
8781  030454  005037  001224         23$:  CLR     $TMP1           ;CLEAR PASS FLAGS
8782  030460  000137  030300               JMP     22$             ;GO SET UP A & B
8783  030464                         50$:
8784                                       ;;*****************************************************************
8785                                       ;*TEST 111      OVERLAPPED OPERATIONS
8786                                       ;*
8787                                       ;*      DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B.  ISSUE  A
8788                                       ;*      SEEK ON DRIVE A TO CYLINDER 1.  IMMEDIATELY ISSUE A WRITE
8789                                       ;*      DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B.
8790                                       ;*      AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD
8791                                       ;*      BE SET AND DRIVE A HAS ATTENTION SET.
8792                                       ;*
8793                                       ;*      REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.
8794                                       ;*
8795                                       ;*      NOTE:   IF ONLY ONE DRIVE IS AVAILABLE THE
8796                                       ;*              TEST WILL NOT BE DONE.
8797                                       ;*
8798                                       ;;*****************************************************************
8799  030464  000004                 TST111: SCOPE
8800  030466  012737  000024  001262       MOV     #20.,$TIMES     ;;DO 20. ITERATIONS
8801
8802  030474  012737  177777  001224  2$:    MOV     #-1,$TMP1       ;SET LOOP CONTROL FLAG
```

```
 8803  030502  013705  001354          MOV     $DEVM,R5        ;GET DEVICE MAP
 8804  030506  005000                  CLR     R0              ;CLEAR FOR DRIVE #A
 8805  030510  005001                  CLR     R1              ;CLEAR FOR DRIVE #B
 8806  030512  012703  000001          MOV     #1,R3           ;SET DRIVE POSITION A
 8807  030516  012704  000001          MOV     #1,R4           ;SET DRIVE POSITION B
 8808  030522  012737  030532  001110  MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
 8809  030530  000545                  BR      11$             ;GO SET UP POINTERS
 8810  030532                  3$:
 8811  030532  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
 8812  030534  104003                  ERROR   3               ;BAD INIT ERROR
 8813
 8814  030536  010037  001626          MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
 8815  030542  010037  001610          MOV     R0,L.CS2        ;SETUP DRIVE A TO RECAL
 8816  030546  012737  000113  001600  MOV     #RECAL,L.CS1
 8817
 8818  030554  104417                  TLOADRK                 ;LOAD RK REGS
 8819  030556  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8820  030560  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8821  030562  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
 8822
 8823  030566  104437                  TWAT8S                  ;WAIT FOR SECOND INTERRUPT
 8824  030570  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8825
 8826  030572  012737  000105  001600  MOV     #CLEAR,L.CS1    ;SET UP TO CLEAR DRIVE
 8827  030600  104417                  TLOADRK                 ;LOAD RK REGS
 8828  030602  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8829  030604  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8830  030606                  4$:
 8831  030606  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8832  030610  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8833  030612  032737  040000  001540  BIT     #DI,T.CS1       ;TEST IF DI STILL SET
 8834  030620  001372                  BNE     4$              ;YES - LOOP
 8835  030622  010137  001626          MOV     R1,DRVNUM       ;STORE DRIVE FOR REPORT
 8836  030626  010137  001610          MOV     R1,L.CS2        ;SETUP DRIVE B TO RECAL
 8837  030632  012737  000113  001600  MOV     #RECAL,L.CS1
 8838
 8839  030640  104417                  TLOADRK                 ;LOAD RK REGS
 8840  030642  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8841  030644  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8842  030646  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
 8843  030652  104437                  TWAT8S                  ;WAIT FOR SECOND INTERRUPT
 8844  030654  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8845  030656  012737  000105  001600  MOV     #CLEAR,L.CS1    ;SET UP DRIVE CLEAR
 8846  030664  104417                  TLOADRK                 ;LOAD RK REGS
 8847  030666  104423                  TWAT16                  ;WAIT FOR INTERRUPT
 8848  030670  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8849  030672                  5$:
 8850  030672  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8851  030674  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8852  030676  032737  040000  001540  BIT     #DI,T.CS1       ;TEST IF DI STILL SET
 8853  030704  001372                  BNE     5$              ;YES - LOOP
 8854
 8855  030706  010037  001626          MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
 8856  030712  010037  001610          MOV     R0,L.CS2        ;SETUP DRIVE A TO SEEK
 8857  030716  012737  000001  001614  MOV     #1,L.DCYL       ;TO CYL 1
 8858  030724  012737  0001.7  001600  MOV     #SEEK,L.CS1
```

```
8859
8860  030732  104417                          TLOADRK                 ;LOAD RK REGS
8861  030734  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8862  030736  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERRCR
8863
8864  030740  010137  001626                  MOV     R1,DRVNUM       ;STORE DRIVE FOR REPORT
8865  030744  010137  001610                  MOV     R1,L.CS2        ;SETUP DRIVE B TO WRITE DATA
8866  030750  012737  000100  001614          MOV     #100,L.DCYL     ;AT CYL 100
8867  030756  012737  177400  001602          MOV     #-400,L.WC      ;400 WORDS
8868  030764  012737  062604  001604          MOV     #OBUFF,L.BA
8869  030772  012737  000123  001600          MOV     #WRDATA,L.CS1
8870
8871  031000  104417                          TLOADRK                 ;LOAD RK REGS-DO WRITE
8872
8873  031002  104427                          TWAT80                  ;WAIT FOR INTERRUPT
8874  031004  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8875
8876  031006  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8877  031010  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8878
8879  031012  010037  001626                  MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
8880  031016  130337  001557                  BITB    R3,T.ASOF+1     ;CHECK IF DRIVE ATTENTION IS DRIVE A
8881  031022  001007                          BNE     10$             ;YES-SKIP
8882  031024  012737  055421  001460          MOV     #EMDA,EM11N     ;"DRIVE ATTENTION NOT SET RESULT OF
8883  031032  012737  047010  057702          MOV     #EMSK,DF011A    ;SEEK"
8884  031040  104011                          ERROR   11
8885
8886  031042  104415                  10$:    SCOP1                   ;LOCAL LOOP TO 3$
8887
8888                                  ;       THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
8889                                  ;       OF DRIVES AVAILABLE.  THE FIRST PASS OF THE PROGRAM WILL USE THE
8890                                  ;       LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
8891                                  ;       B.  THE SECOND PASS SWAPS DRIVE A & B.  THE THIRD PASS USES
8892                                  ;       THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
8893                                  ;       AS A.  THE FORTH PASS SWAPS A & B AGAIN.  THIS CONTINUES
8894                                  ;       UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
8895                                  ;       DRIVE.
8896                                  ;
8897                                  ;       THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
8898                                  ;       THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
8899                                  ;
8900                                  ;       THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
8901                                  ;       CHECKED.
8902
8903  031044  005237  001224          11$:    INC     $TMP1           ;INCREMENT PASS CONTROL
8904  031050  001024                          BNE     16$             ;SKIP IF NOT ZERO
8905                                                                  ;(IT WILL BE ZERO ON THE 1ST PASS)
8906
8907  031052  030305                  12$:    BIT     R3,R5           ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8908  031054  001006                          BNE     13$             ;YES-SKIP
8909
8910  031056  005200                  22$:    INC     R0              ;BUMP R0 (DRIVE A)
8911  031060  006303                          ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
8912  031062  032703  000400                  BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
8913  031066  001771                          BEQ     12$             ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8914  031070  000464                          BR      50$             ;DONE-EXIT
```

```
8915
8916  031072  010001                  13$:    MOV     R0,R1           ;SET DRIVE B TO THE SAME AS A
8917  031074  010304                          MOV     R3,R4
8918  031076  005201                  14$:    INC     R1              ;BUMP R1 (DRIVE B)
8919  031100  006304                          ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
8920  031102  030405                          BIT     R4,R5           ;IS THIS DRIVE AVAIL?
8921  031104  001004                          BNE     15$             ;YES-SKIP
8922  031106  032704  000400                  BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
8923  031112  001771                          BEQ     14$             ;NO-LOOP
8924  031114  000452                          BR      50$             ;DONE-EXIT
8925
8926  031116  000137  030532          15$:    JMP     3$              ;GO DO THE TEST ON THE DRIVE A & B
8927                                                                  ;CONTAINED IN R0 & R1
8928  031122  032737  000001  001224  16$:    BIT     #BIT0,$TMP1     ;IS PASS FLAGS ODD?
8929  031130  001410                          BEQ     17$             ;NO-SKIP
8930
8931  031132  010046                          MOV     R0,-(SP)        ;
8932  031134  010346                          MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
8933  031136  010403                          MOV     R4,R3           ;TO EXCHANGE DRIVE A & B
8934  031140  010100                          MOV     R1,R0
8935  031142  012604                          MOV     (SP)+,R4
8936  031144  012601                          MOV     (SP)+,R1
8937  031146  000137  030532                  JMP     3$              ;REPEAT TEST ON THIS COMBO.
8938
8939  031152  032737  000002  001224  17$:    BIT     #BIT1,$TMP1     ;TEST IF PASS FLAGS AT HALF MODULE 4?
8940  031160  001410                          BEQ     19$             ;NO-SKIP TO BUMP DRIVE B
8941  031162  005200                  18$:    INC     R0              ;BUMP DRIVE A
8942  031164  006303                          ASL     R3              ;SHIFT DRIVE SELECT BIT
8943  031166  030305                          BIT     R3,R5           ;AVAILABLE?
8944  031170  001014                          BNE     20$             ;YES-SKIP
8945  031172  032703  000400                  BIT     #BIT8,R3        ;ALL CHECKED?
8946  031176  001771                          BEQ     18$             ;NO-SKIP
8947  031200  000412                          BR      21$             ;GO TO NEXT PASS
8948
8949  031202  005201                  19$:    INC     R1              ;BUMP DRIVE B
8950  031204  006304                          ASL     R4              ;SHIFT DRIVE SELECT BIT
8951  031206  030405                          BIT     R4,R5           ;AVAILABLE?
8952  031210  001004                          BNE     20$             ;YES-SKIP
8953  031212  032704  000400                  BIT     #BIT8,R4        ;ALL CHECKED?
8954  031216  001771                          BEQ     19$             ;NO-LOOP
8955  031220  000404                          BR      23$             ;YES-SKIP TO NEXT PASS
8956
8957  031222  000137  030532          20$:    JMP     3$              ;GO TEST THIS COMBO
8958
8959  031226  010100                  21$:    MOV     R1,R0           ;SET DRIVE 0 TO LOW POSITION THIS PASS
8960  031230  010403                          MOV     R4,R3           ;SET SELECT BITS TO AGREE
8961  031232  005037  001224          23$:    CLR     $TMP1           ;CLEAR PASS FLAGS
8962  031236  000137  031056                  JMP     22$             ;GO SET UP A & B
8963  031242                          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

031242                          $EOP:
031242  000004                          SCOPE
031244  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
031250  005037  001262                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
031254  005237  001304                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
031260  042737  100000  001304          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
031266  005327                          DEC     (PC)+           ;;LOOP?
031270  000001                  $EOPCT: .WORD   1
031272  003063                          BGT     $DOAGN          ;;YES
031274  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
031276  000001                  $ENDCT: .WORD   1
031300  031270                          $EOPCT
031302  104401  031310                  TYPE    ,65$            ;;TYPE ASCIZ STRING
031306  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
031326                          64$:
031326  013746  001304                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
031332  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
031334  104401  031342                  TYPE    ,67$            ;;TYPE ASCIZ STRING
031340  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
031404                          66$:
031404  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
031410  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
031412  104401  001273                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
031416  005037  001112                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
031422  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
031426  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
031430  000005                          RESET                   ;;CLEAR THE WORLD
031432  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
031434  000240                          NOP                     ;;SAVE ROOM
031436  000240                          NOP                     ;;FOR
031440  000240                          NOP                     ;;ACT11
031442                          $DOAGN:
031442  000137                          JMP     @(PC)+          ;;RETURN
031444  003116                  $RTNAD: .WORD   TST1
031446     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
031452                                  .EVEN
                                .SBTTL  ROUTINE TO SIZE MEMORY

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

```
9020                                    ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
9021                                    ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
9022                                    ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
9023                                    ;*$KT11 IS THE MEMORY MANAGEMENT KEY
9024                                    ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
9025                                    ;*      MUST BE SETUP BEFORE THE CALL
9026                                    ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
9027                                    ;*      DETERMINED BY ROUTINE
9028
9029  031452  010046                    $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
9030  031454  010146                            MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
9031  031456  010246                            MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
9032  031460  010346                            MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
9033  031462  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
9034  031466  013746  000006                    MOV     @#ERRVEC+2,-(SP)
9035  031472  010600                            MOV     SP,R0           ;;SAVE THE STACK POINTER
9036                                    ;;SET THE ERRVEC PS TO THE PRESENT PS
9037  031474  104400                            TRAP                    ;;PUSH OLD PSW AND PC ON STACK
9038  031476  012637  000006                    MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
9039  031502  012701  003776                    MOV     #3776,R1        ;;SETUP ADDRESS
9040  031506  105727                            TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
9041  031510  000200                    $KT11:  .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
9042  031512  100062                            BPL     $CORE           ;;BR IF NO
9043  031514  012737  031652  000004            MOV     #$KTNEX,@#ERRVEC        ;;SET FOR TIMEOUT
9044  031522  005737  177572                    TST     @#SR0           ;;KT11 ARE YOU THERE?
9045  031526  052737  100000  031510            BIS     #100000,$KT11   ;;YES--SET KT11 KEY
9046  031534  005046                            CLR     -(SP)           ;;INITIALIZE FOR "PAR" LOADING
9047  031536  012702  172340                    MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST "PAR"
9048  031542  012703  000010                    MOV     #^D8,R3         ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
9049  031546  012762  077406  177740    1$:     MOV     #77406,-40(R2)  ;;PDR = 4K, UP, READ/WRITE
9050  031554  011622                            MOV     (SP),(R2)+      ;;LOAD "PAR"
9051  031556  062716  000200                    ADD     #200,(SP)       ;;UPDATE FOR NEXT "PAR"
9052  031562  077307                            SOB     R3,1$           ;;LOOP UNTIL ALL EIGHT ARE LOADED
9053  031564  012742  177600                    MOV     #177600,-(R2)   ;;SETUP KIPAR7 FOR I/O
9054  031570  005042                            CLR     -(R2)           ;;SETUP KIPAR6 FOR TESTING
9055  031572  012737  031610  000004            MOV     #2$,@#ERRVEC    ;;CATCH TIMEOUT IF NO SR3
9056  031600  012737  000020  172516            MOV     #20,@#SR3       ;;ENABLE 22 BIT MODE
9057  031606  000401                            BR      3$              ;;THIS PDP-11 HAS A SR3 REGISTER
9058  031610  022626                    2$:     CMP     (SP)+,(SP)+     ;;CLEAN OFF THE STACK--NO SR3
9059  031612  005237  177572            3$:     INC     @#SR0           ;;TURN ON MEMORY MANAGEMENT
9060  031616  012737  031642  000004            MOV     #$KTOUT,@#ERRVEC        ;;SET FOR TIME OUT
9061  031624  005737  143776            4$:     TST     @#143776        ;;TRAP ON NON-EX-MEM
9062  031630  062712  000040                    ADD     #40,(R2)        ;;MAKE A 1K STEP
9063  031634  023712  172356                    CMP     @#KIPAR7,(R2)   ;;LAST ONE?
9064  031640  101371                            BHI     4$              ;;NO--TRY IT
9065  031642  011202                    $KTOUT: MOV     (R2),R2         ;;GET LAST BANK+1
9066  031644  005037  177572                    CLR     @#SR0           ;;TURN OFF MEMORY MANAGEMENT
9067  031650  000421                            BR      $SIZEX
9068  031652  042737  100000  031510  $KTNEX: BIC     #100000,$KT11   ;;KT11 NON-EXISTENT
9069  031660  012737  031710  000004  $CORE:  MOV     #$CROUT,@#ERRVEC        ;;SET FOR TIMEOUT
9070  031666  005002                            CLR     R2              ;;SET UP BANK
9071  031670  062701  004000            1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
9072  031674  062702  000040                    ADD     #40,R2          ;;1K STEP
9073  031700  005711                            TST     (R1)            ;;TRAP ON TIME OUT
9074  031702  022701  177776                    CMP     #177776,R1      ;;LAST ONE
9075  031706  001370                            BNE     1$              ;;NO--TRY AGAIN
```

```
9076  031710  162701  004000          $CROUT: SUB     #4000,R1
9077  031714  162702  000040          $SIZEX: SUB     #40,R2          ;;DROP BACK
9078  031720  010006                          MOV     R0,SP           ;;RESTORE THE STACK
9079  031722  012637  000006                  MOV     (SP)+,@#ERRVEC+2;;RESTORE ERROR VECTOR
9080  031726  012637  000004                  MOV     (SP)+,@#ERRVEC
9081  031732  010137  031754                  MOV     R1,$LSTAD       ;;LAST ADDRESS
9082  031736  010237  031756                  MOV     R2,$LSTBK       ;;LAST BANK
9083  031742  012603                          MOV     (SP)+,R3        ;;RESTORE R3
9084  031744  012602                          MOV     (SP)+,R2        ;;RESTORE R2
9085  031746  012601                          MOV     (SP)+,R1        ;;RESTORE R1
9086  031750  012600                          MOV     (SP)+,R0        ;;RESTORE R0
9087  031752  000207                          RTS     PC
9088  031754  000000          $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
9089  031756  000000          $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
9090                          .SBTTL  SCOPE HANDLER ROUTINE
9091
9092                          ;;**************************************************************
9093                          ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9094                          ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
9095                          ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
9096                          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9097                          ;*SW14=1        LOOP ON TEST
9098                          ;*SW11=1        INHIBIT ITERATIONS
9099                          ;*SW09=1        LOOP ON ERROR
9100                          ;*SW08=1        LOOP ON TEST IN SWR<7:0>
9101                          ;*CALL
9102                          ;*      SCOPE           ;;SCOPE=IOT
9103
9104  031760                  $SCOPE:
9105  031760  104407                  CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
9106                          ;;GO TO ERROR ROUTINE IF RETURN PC LESS THAN 1002
9107                          ;;OTHERWISE CONTINUE
9108  031762  021627  001002          CMP     (SP),#1002      ;;UNEXPECTED TRAP OR INTERRRUPT
9109  031766  101002                  BHI     1$              ;;ARE TRAPPED HERE VIA IOT
9110  031770  000137  033000          JMP     $ERROR          ;;GO PROCESS UNEXPECTED TRAP
9111  031774  032777  040000  147136 1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
9112  032002  001131                  BNE     $OVER           ;;YES IF SW14=1
9113                          ;*****START OF CODE FOR THE XOR TESTER*****
9114  032004  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
9115                                                          ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
9116  032006  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
9117  032012  012737  032032  000004  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
9118  032020  005737  177060          TST     @#177060        ;;TIME OUT ON XOR?
9119  032024  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
9120  032030  000500                  BR      $SVLAD          ;;GO TO THE NEXT TEST
9121  032032  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
9122  032034  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
9123  032040  000440                  BR      7$              ;;LOOP ON THE PRESENT TEST
9124  032042                  6$:;*****END OF CODE FOR THE XOR TESTER*****
9125  032042  032777  000400  147070      BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
9126  032050  001421                  BEQ     2$              ;;BR IF NO
9127  032052  005046                  CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
9128  032054  117716  147060          MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
9129  032060  001414                  BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
9130  032062  022716  000111          CMP     #111,(SP)       ;;CHECK THE NUMBER IN THE SWR
9131  032066  002411                  BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
```

```
 9132  032070  011637  001102                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
 9133  032074  005316                          DEC     (SP)            ;;BACKUP BY ONE
 9134  032076  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
 9135  032100  062716  032304                  ADD     #$SWOBTBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
 9136  032104  013637  001106                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
 9137  032110  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
 9138  032112  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
 9139  032114  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 9140  032120  001421                          BEQ     3$              ;;BR IF NO
 9141  032122  123737  001115  001103        CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 9142  032130  101015                          BHI     3$              ;;BR IF NO
 9143  032132  032777  001000  147000        BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 9144  032140  001404                          BEQ     4$              ;;BR IF NO
 9145  032142  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 9146  032150  000446                          BR      $OVER
 9147  032152  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 9148  032156  005037  001262                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 9149  032162  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
 9150  032164  032777  004000  146746  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 9151  032172  001011                          BNE     1$              ;;BR IF YES
 9152  032174  005737  001304                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 9153  032200  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
 9154  032202  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
 9155  032206  023737  001262  001104        CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 9156  032214  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 9157  032216  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 9158  032224  013737  032302  001262        MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 9159  032232  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 9160  032236  113737  001102  001302        MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
 9161  032244  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 9162  032250  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 9163  032254  005037  001264                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 9164  032260  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 9165  032266  013777  001102  146646  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 9166  032274  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 9167  032300  000002                          RTI                     ;;FIXES PS
 9168  032302  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 9169  032304                          $SWOBTBL:
 9170  032304  003120                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
 9171  032306  003236                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
 9172  032310  003324                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
 9173  032312  004350                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
 9174  032314  004452                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
 9175  032316  004630                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
 9176  032320  004756                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
 9177  032322  005062                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
 9178  032324  006006                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
 9179  032326  006124                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
 9180  032330  006234                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
 9181  032332  006476                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
 9182  032334  006630                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
 9183  032336  007040                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
 9184  032340  007230                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
 9185  032342  007352                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
 9186  032344  010140                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
 9187  032346  010536                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
```

```
 9188  032350  010750                .WORD   TST23+2          ;;STARTING ADDRESS OF TEST 23
 9189  032352  011166                .WORD   TST24+2          ;;STARTING ADDRESS OF TEST 24
 9190  032354  011312                .WORD   TST25+2          ;;STARTING ADDRESS OF TEST 25
 9191  032356  011444                .WORD   TST26+2          ;;STARTING ADDRESS OF TEST 26
 9192  032360  011622                .WORD   TST27+2          ;;STARTING ADDRESS OF TEST 27
 9193  032362  011754                .WORD   TST30+2          ;;STARTING ADDRESS OF TEST 30
 9194  032364  012250                .WORD   TST31+2          ;;STARTING ADDRESS OF TEST 31
 9195  032366  012440                .WORD   TST32+2          ;;STARTING ADDRESS OF TEST 32
 9196  032370  012700                .WORD   TST33+2          ;;STARTING ADDRESS OF TEST 33
 9197  032372  013066                .WORD   TST34+2          ;;STARTING ADDRESS OF TEST 34
 9198  032374  013320                .WORD   TST35+2          ;;STARTING ADDRESS OF TEST 35
 9199  032376  013506                .WORD   TST36+2          ;;STARTING ADDRESS OF TEST 36
 9200  032400  013730                .WORD   TST37+2          ;;STARTING ADDRESS OF TEST 37
 9201  032402  014116                .WORD   TST40+2          ;;STARTING ADDRESS OF TEST 40
 9202  032404  014310                .WORD   TST41+2          ;;STARTING ADDRESS OF TEST 41
 9203  032406  014504                .WORD   TST42+2          ;;STARTING ADDRESS OF TEST 42
 9204  032410  014710                .WORD   TST43+2          ;;STARTING ADDRESS OF TEST 43
 9205  032412  015114                .WORD   TST44+2          ;;STARTING ADDRESS OF TEST 44
 9206  032414  015574                .WORD   TST45+2          ;;STARTING ADDRESS OF TEST 45
 9207  032416  016000                .WORD   TST46+2          ;;STARTING ADDRESS OF TEST 46
 9208  032420  016204                .WORD   TST47+2          ;;STARTING ADDRESS OF TEST 47
 9209  032422  016422                .WORD   TST50+2          ;;STARTING ADDRESS OF TEST 50
 9210  032424  016626                .WORD   TST51+2          ;;STARTING ADDRESS OF TEST 51
 9211  032426  017032                .WORD   TST52+2          ;;STARTING ADDRESS OF TEST 52
 9212  032430  017156                .WORD   TST53+2          ;;STARTING ADDRESS OF TEST 53
 9213  032432  017302                .WORD   TST54+2          ;;STARTING ADDRESS OF TEST 54
 9214  032434  017426                .WORD   TST55+2          ;;STARTING ADDRESS OF TEST 55
 9215  032436  017552                .WORD   TST56+2          ;;STARTING ADDRESS OF TEST 56
 9216  032440  017676                .WORD   TST57+2          ;;STARTING ADDRESS OF TEST 57
 9217  032442  017742                .WORD   TST60+2          ;;STARTING ADDRESS OF TEST 60
 9218  032444  020014                .WORD   TST61+2          ;;STARTING ADDRESS OF TEST 61
 9219  032446  020272                .WORD   TST62+2          ;;STARTING ADDRESS OF TEST 62
 9220  032450  020454                .WORD   TST63+2          ;;STARTING ADDRESS OF TEST 63
 9221  032452  020712                .WORD   TST64+2          ;;STARTING ADDRESS OF TEST 64
 9222  032454  021100                .WORD   TST65+2          ;;STARTING ADDRESS OF TEST 65
 9223  032456  021604                .WORD   TST66+2          ;;STARTING ADDRESS OF TEST 66
 9224  032460  022014                .WORD   TST67+2          ;;STARTING ADDRESS OF TEST 67
 9225  032462  022120                .WORD   TST70+2          ;;STARTING ADDRESS OF TEST 70
 9226  032464  022540                .WORD   TST71+2          ;;STARTING ADDRESS OF TEST 71
 9227  032466  023160                .WORD   TST72+2          ;;STARTING ADDRESS OF TEST 72
 9228  032470  023270                .WORD   TST73+2          ;;STARTING ADDRESS OF TEST 73
 9229  032472  024122                .WORD   TST74+2          ;;STARTING ADDRESS OF TEST 74
 9230  032474  024600                .WORD   TST75+2          ;;STARTING ADDRESS OF TEST 75
 9231  032476  025014                .WORD   TST76+2          ;;STARTING ADDRESS OF TEST 76
 9232  032500  025230                .WORD   TST77+2          ;;STARTING ADDRESS OF TEST 77
 9233  032502  025310                .WORD   TST100+2                 ;;STARTING ADDRESS OF TEST 100
 9234  032504  025502                .WORD   TST101+2                 ;;STARTING ADDRESS OF TEST 101
 9235  032506  025726                .WORD   TST102+2                 ;;STARTING ADDRESS OF TEST 102
 9236  032510  026054                .WORD   TST103+2                 ;;STARTING ADDRESS OF TEST 103
 9237  032512  026400                .WORD   TST104+2                 ;;STARTING ADDRESS OF TEST 104
 9238  032514  027132                .WORD   TST105+2                 ;;STARTING ADDRESS OF TEST 105
 9239  032516  027334                .WORD   TST106+2                 ;;STARTING ADDRESS OF TEST 106
 9240  032520  027554                .WORD   TST107+2                 ;;STARTING ADDRESS OF TEST 107
 9241  032522  027770                .WORD   TST110+2                 ;;STARTING ADDRESS OF TEST 110
 9242  032524  030466                .WORD   TST111+2                 ;;STARTING ADDRESS OF TEST 111
 9243  032526  031244                .WORD   $EOP+2           ;ADDRESS OF END OF PASS
```

```
 9244  032530  042710                        .WORD   ABTFAIL+2       ;ADDRESS OF ABORT FAILURE HANDLER
 9245
 9246                                  .SBTTL  APT COMMUNICATIONS ROUTINE
 9247
 9248                                  ;;**********************************************************************
 9249  032532  112737  000001  032776  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 9250  032540  112737  000001  032774  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 9251  032546  000403                        BR      $ATYC
 9252  032550  112737  000001  032776  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 9253  032556                        $ATYC:
 9254  032556  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9255  032560  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9256  032562  105737  032774                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 9257  032566  001450                        BEQ     5$              ;;IF NOT:  BR
 9258  032570  122737  000001  001316        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 9259  032576  001031                        BNE     3$              ;;IF NOT:  BR
 9260  032600  132737  000100  001317        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 9261  032606  001425                        BEQ     3$              ;;IF NOT:  BR
 9262  032610  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 9263  032614  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9264  032622  005737  001276          1$:   TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 9265  032626  001375                        BNE     1$              ;;IF NOT:  WAIT
 9266  032630  010037  001312                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 9267  032634  105720                  2$:   TSTB    (R0)+           ;;FIND END OF MESSAGE
 9268  032636  001376                        BNE     2$
 9269  032640  163700  001312                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 9270  032644  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 9271  032646  010037  001314                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 9272  032652  012737  000004  001276        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 9273  032660  000413                        BR      5$
 9274  032662  017637  000004  032706  3$:   MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 9275  032670  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 9276  032676  013746  177776                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 9277  032702  004737  042720                JSR     PC,$TYPE        ;;CALL TYPE MACRO
 9278  032706  000000                  4$:   .WORD   0
 9279  032710                           5$:
 9280  032710  105737  032776          10$:  TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9281  032714  001416                        BEQ     12$             ;;IF NOT:  BR
 9282  032716  005737  001316                TST     $ENV            ;;RUNNING UNDER APT?
 9283  032722  001413                        BEQ     12$             ;;IF NOT:  BR
 9284  032724  005737  001276          11$:  TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9285  032730  001375                        BNE     11$             ;;IF NOT:  WAIT
 9286  032732  017637  000004  001300        MOV     @4(SP),$FATAL   ;;GET ERROR #
 9287  032740  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9288  032746  005237  001276                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9289  032752  105037  032776          12$:  CLRB    $FFLG           ;;CLEAR FATAL FLAG
 9290  032756  105037  032775                CLRB    $LFLG           ;;CLEAR LOG FLAG
 9291  032762  105037  032774                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9292  032766  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9293  032770  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9294  032772  000207                        RTS     PC              ;;RETURN
 9295  032774     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
 9296  032775     000          $LFLG:  .BYTE   0               ;;LOG FLAG
 9297  032776     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
 9298             033000                  .EVEN
 9299             000200          APTSIZE=200
```

```
  9300            000001                  APTENV=001
  9301            000100                  APTSPOOL=100
  9302            000040                  APTCSUP=040
  9303                                    .SBTTL  ERROR HANDLER ROUTINE
  9304
  9305                                    ;;*****************************************************************
  9306                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
  9307                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
  9308                                    ;*AND GO TO TYPERR ON ERROR
  9309                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  9310                                    ;*SW15=1        HALT ON ERROR
  9311                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
  9312                                    ;*SW10=1        BELL ON ERROR
  9313                                    ;*SW09=1        LOOP ON ERROR
  9314                                    ;*CALL
  9315                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
  9316
  9317  033000                            $ERROR:
  9318  033000  104407                           CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
  9319  033002  105237  001103           7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
  9320  033006  001775                           BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
  9321  033010  013777  001102  146124          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
  9322  033016  032777  002000  146114          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
  9323  033024  001402                           BEQ     1$              ;;NO - SKIP
  9324  033026  104401  001266                   TYPE    ,$BELL          ;;RING BELL
  9325  033032  005237  001112           1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
  9326  033036  011637  001116                   MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
  9327  033042  162737  000002  001116          SUB     #2,$ERRPC
  9328  033050  117737  146042  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
  9329  033056  032777  020000  146054          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
  9330  033064  001055                           BNE     20$             ;;SKIP TYPEOUTS
  9331  033066  021627  001002                   CMP     (SP),#1002      ;;IF RETURN PC LESS THAN 1002
  9332  033072  101046                           BHI     12$             ;;ERROR IS ILLEGAL TRAP
  9333                                    ;;PROCESS UNEXPECTED TRAP OR INTERRUPT
  9334  033074  016637  000004  001116          MOV     4(SP),$ERRPC    ;;GET PC AT TIME OF FALSE TRAP
  9335  033102  162737  000002  001116          SUB     #2,$ERRPC       ;;ADJUST PC
  9336  033110  104401  033154                   TYPE    ,10$            ;;TYPE HEADER
  9337  033114  013746  001116                   MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
  9338  033120  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  9339  033122  104401  033162                   TYPE    ,11$
  9340  033126  162716  000004                   SUB     #4,(SP)         ;;GET FALSE TRAP VECTOR ADDR
  9341  033132  011637  001116                   MOV     (SP),$ERRPC
  9342  033136  013746  001116                   MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
  9343  033142  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  9344  033144  104401  001273                   TYPE    ,$CRLF
  9345  033150  022626                           CMP     (SP)+,(SP)+     ;;POP FALSE TRAP VECTOR PC&ADDR
  9346  033152  000422                           BR      20$
  9347  033154  050200  036503  000040  10$:     .ASCIZ  <200>'PC= '
  9348  033162  020040  047125  054105  11$:     .ASCIZ  '  UNEXPECTED TRAP TO '
  9349  033170  042520  052103  042105
  9350  033176  052040  040522  020120
  9351  033204  047524  000040
  9352                                          .EVEN
  9353  033210                            12$:
  9354  033210  004737  033322                   JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
  9355  033214  104401  001273                   TYPE    ,$CRLF
```

```
 9356  033220                          20$:
 9357  033220  122737  000001  001316          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9358  033226  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
 9359  033230  113737  001114  033242          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 9360  033236  004737  032550                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 9361  033242  000                     21$:    .BYTE   0
 9362  033243  000                             .BYTE   0
 9363  033244  000777                  22$:    BR      22$             ;;APT ERROR LOOP
 9364  033246  005777  145666          2$:     TST     @SWR            ;;HALT ON ERROR
 9365  033252  100002                          BPL     3$              ;;SKIP IF CONTINUE
 9366  033254  000000                          HALT                    ;;HALT ON ERROR!
 9367  033256  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9368  033260  032777  001000  145652  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 9369  033266  001402                          BEQ     4$              ;;BR IF NO
 9370  033270  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 9371  033274  005737  001264          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 9372  033300  001402                          BEQ     5$              ;;BR IF NONE
 9373  033302  013716  001264                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 9374  033306                          5$:
 9375  033306  022737  031432  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 9376  033314  001001                          BNE     6$              ;;BRANCH IF NO
 9377  033316  000000                          HALT                    ;;YES
 9378  033320                          6$:
 9379  033320  000002                          RTI                     ;;RETURN
 9380                                  ;;*****************************************************************
 9381                                  .SBTTL  TYPE ERROR ROUTINE
 9382                                  ;*ENTRY JSR PC,TYPERR
 9383                                  ;*RETURN RTS PC
 9384                                  ;*
 9385                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 9386                                  ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
 9387                                  ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
 9388                                  ;*THE ERROR.
 9389                                  ;;*****************************************************************
 9390  033322  104413                  TYPERR: SAVREG
 9391  033324  113737  001102  001302          MOVB    $TSTNM,$TESTN   ;GET TEST NUMBER FOR REPORT
 9392  033332  042737  177400  001302          BIC     #177400,$TESTN  ;CLEAR UNUSED BITS
 9393  033340  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
 9394  033344  042700  177400                  BIC     #177400,R0      ;CLEAR UNUSED BITS
 9395  033350  005300                          DEC     R0              ;FORM INDEX FOR ERROR TABLE
 9396  033352  006300                          ASL     R0
 9397  033354  006300                          ASL     R0
 9398  033356  006300                          ASL     R0
 9399  033360  062700  001360          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
 9400  033364  012037  033400                  MOV     (R0)+,2$        ;GET EM POINTER
 9401  033370  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
 9402  033372  104401  001273                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
 9403  033376  104401                          TYPE                    ;TYPE ERROR MESSAGE (EM)
 9404  033400  000000                  2$:     .WORD   0               ;EM POINTER GOES HERE
 9405  033402  012037  033416          3$:     MOV     (R0)+,4$        ;GET DH POINTER
 9406  033406  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
 9407  033410  104401  001273                  TYPE    ,$CRLF          ;TYPE CR-LF
 9408  033414  104401                          TYPE                    ;TYPE DATA HEADER
 9409  033416  000000                  4$:     .WORD   0               ;DH POINTER GOES HERE
 9410  033420  012001                  5$:     MOV     (R0)+,R1        ;GET DT POINTER
 9411  033422  001445                          BEQ     20$             ;BRANCH IF THERE ARE NONE
```

```
9412  033424  005004                    CLR     R4              ;SET INDENT SWITCH
9413  033426  012000                    MOV     (R0)+,R0        ;GET DF POINTER
9414  033430  012002                    MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
9415  033432  104401  001273            TYPE    ,$CRLF
9416  033436  112003            10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
9417  033440  105720                    TSTB    (R0)+           ;BUMP PAST FORMAT WORD
9418  033442  005703                    TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
9419  033444  001416                    BEQ     14$             ;NO - SKIP DATA PRINT
9420  033446  005704                    TST     R4              ;CHECK FOR INDENT
9421  033450  001004                    BNE     12$             ;YES, GO INDENT
9422  033452  013146            11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
9423  033454  104402                    TYPOC                   ;TYPE IT
9424  033456  005303                    DEC     R3              ;MORE DATA WORDS
9425  033460  001403                    BEQ     13$             ;NO-BRANCH
9426  033462  104401  047501    12$:    TYPE    ,SPACE2         ;TYPE SEPARATORS
9427  033466  000771                    BR      11$             ;LOOP
9428  033470  104401  001273    13$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
9429  033474  005710                    TST     (R0)            ;CHECK IF NEXT HEADER AVAILABLE
9430  033476  001401                    BEQ     14$             ;NO, DO NOT CHANGE INDENT
9431  033500  005104                    COM     R4              ;CHANGE INDENT
9432  033502  005302            14$:    DEC     R2              ;MORE DH'S?
9433  033504  003414                    BLE     20$             ;NO-BRANCH
9434  033506  012037  033526    15$:    MOV     (R0)+,18$       ;GET NEXT DH POINTER
9435  033512  001751                    BEQ     10$             ;IF 0 GET DATA
9436  033514  005704                    TST     R4              ;INDENT?
9437  033516  001402                    BEQ     17$             ;NO, BRANCH
9438  033520  104401  047501            TYPE    ,SPACE2         ;TYPE INDENT
9439  033524  104401            17$:    TYPE                    ;TYPE DH
9440  033526  000000            18$:    .WORD   0               ;DH POINTER GOES HERE
9441  033530  104401  001273            TYPE    ,$CRLF
9442  033534  000740                    BR      10$             ;GET DATA
9443  033536  104414            20$:    RESREG
9444  033540  005237  001632            INC     ERRCNT          ;INCREMENT THE ERROR COUNT
9445  033544  032777  010000  145366    BIT     #SW12,@SWR      ;CHECK IF SWITCH 12 SET
9446  033552  001421                    BEQ     25$             ;NO, RETURN
9447  033554  022737  000024  001632    CMP     #20.,ERRCNT     ;CHECK IF ERROR THRESHOLD EXCEEDED
9448  033562  103015                    BHIS    25$             ;NO, RETURN
9449  033564  104401  052130            TYPE    ,ABORT          ;TYPE "PROGRAM ABORTED BECAUSE
9450                                                            ;ERROR THRESHOLD EXCEEDED"
9451  033570  005737  000042            TST     42              ;CHECK IF CHAIN MODE
9452  033574  001407                    BEQ     22$             ;NO, HALT PROCESSOR
9453  033576  012706  001100            MOV     #STACK,SP       ;INITIALIZE STACK
9454  033602  012737  000001  031270    MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
9455  033610  000137  031242            JMP     $EOP
9456  033614  000000            22$:    HALT
9457  033616  000207            25$:    RTS     PC
9458                            .SBTTL  NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER
9459                            ;*      THIS ROUTINE SETS THE INTERRUPT FLAG AND DOES AN RTI.
9460                            ;*      THIS IS THE INDICATION TO THE ROUTINE CHECKING
9461                            ;*      NON-EXIXTANT MEMORY THAT AN INTERRUPT DID OCCUR.
9462
9463  033620  005237  001662    NEXINT: INC     INTSET          ;BUMP THE INTERRUPT COUNTER
9464  033624  000002                    RTI
9465
9466                            .SBTTL  RK611 INTERRUPT HANDLER
9467                            ;*      MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE.  ACTUAL
```

```
                                        ;*      PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
                                        ;*      PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
                                        ;*      INTERRUPT OCCURRED.

033626  005237  001662          INTHLR: INC     INTSET          ;BUMP THE INTERRUPT FLAG
033632  000002                          RTI                     ;RETURN.

                                        .SBTTL  MEMORY PARITY ERROR TRAP HANDLER
                                        ;*      MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE. THE REPORT
                                        ;*      WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.

033634  032777  020000  145276  PERHLR: BIT     #BIT13,@SWR     ;TEST IF INHIBIT REPORT
033642  001003                          BNE     1$              ;YES - SKIP
033644  104401  052357                  TYPE    ,EM3            ;TYPE PARITY ERROR MESSAGE
033650  104402                          TYPOC                   ;AND PC VALUE
033652  012737  000001  031270  1$:     MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
033660  012706  001100                  MOV     #STACK,SP       ;CLEAN OFF STACK
033664  000137  031242                  JMP     $EOP

                                        .SBTTL  LINE CLOCK INTERRUPT HANDLER
                                        ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
                                        ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.

033670  005237  001660          LCKHLR: INC     LCLKTK          ;INCREMENT CLOCK TICK COUNTER
033674  042777  000200  145774          BIC     #BIT7,@KWLADD   ;CLEAR MONITOR BIT
033702  000002                          RTI

                                        ;;*****************************************************************
                                        .SBTTL  OPTION PRESENT TEST AND SETUP
                                        ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
                                        ;*      LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF PRESENT; CLEARED
                                        ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
033704  104413                  OPTTST: SAVREG
033706  012701  000004                  MOV     #4,R1           ;SET POINTER TO NEM VECTOR ADDRESS
033712  012146                          MOV     (R1)+,-(SP)     ;STORE VECTOR CONTENTS
033714  011146                          MOV     (R1),-(SP)
033716  012711  000340                  MOV     #PR7,(R1)       ;SET PRIORITY
033722  012741  033620                  MOV     #NEXINT,-(R1)   ;SET VECTOR TO NEM TEST HANDLER
033726  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT COUNTER
033732  012777  000001  145742          MOV     #1,@MMCSR1      ;LOAD CSR FOR BANK 2
033740  000240                          NOP
033742  000240                          NOP
033744  000240                          NOP
033746  005737  001662                  TST     INTSET          ;CHECK IF INTERRUPT OCCURRED
                                                                ;BECAUSE OF NEM ON PARITY OPTION REFER.
033752  001011                          BNE     2$              ;YES - SKIP PARITY SETUP
033754  052737  000200  001656          BIS     #PARBKO,OPTFLG  ;SET PARITY OPTION FLAG
033762  013700  001706                  MOV     MMVECA,R0       ;SET POINTER TO VECTOR
033766  012720  033634                  MOV     #PERHLR,(R0)+   ;INSERT HANDLER ADDRESS
033772  012710  000340                  MOV     #PR7,(R0)       ;INSERT PRIORITY
033776  005037  001662          2$:     CLR     INTSET          ;CLEAR INTERRUPT FLAG
034002  012777  000001  145674          MOV     #1,@MMCSR2      ;LOAD CSR FOR BANK 1
034010  000240                          NOP
034012  000240                          NOP
034014  000240                          NOP
034016  005737  001662                  TST     INTSET          ;TEST IN INTERRUPT OCCURRED
```

```
9524  034022  001003                        BNE     3$                      ;YES - SKIP SETUP
9525  034024  052737  000100  001656        BIS     #PARBK1,OPTFLG          ;SET FLAG
9526  034032  005037  001662        3$:     CLR     INTSET                  ;CLEAR INTERRUPT COUNTER
9527  034036  013700  001700                MOV     KWLVEC,R0               ;SET POINTER TO VECTOR
9528  034042  012720  033670                MOV     #CLKHLR,(R0)+           ;INSERT ADDRESS OF INTERRUPT HDLR
9529  034046  012710  000340                MOV     #PR7,(R0)               ;INSERT PRIORITY
9530  034052  012777  000100  145616        MOV     #BIT6,@KWLADD           ;LOAD KW11-L FOR INTERRUPT ENABLE
9531  034060  000240                        NOP
9532  034062  000240                        NOP
9533  034064  000240                        NOP
9534  034066  005737  001662                TST     INTSET                  ;TEST IN NEM ON KW11-P REFERENCE
9535  034072  001003                        BNE     4$                      ;THIS BRANCH WILL BYPASS SET UP OF
9536                                                                        ;CLOCK OPTION
9537  034074  052737  100000  001656        BIS     #LCLKPR,OPTFLG          ;SET CLOCK PRESENT FLAG
9538  034102  012701  000006        4$:     MOV     #6,R1                   ;RESTORE OLD VECTOR
9539  034106  005037  001662                CLR     INTSET                  ;CLEAR INT FLAG
9540  034112  012611                        MOV     (SP)+,(R1)
9541  034114  012641                        MOV     (SP)+,-(R1)
9542  034116  104414                        RESREG
9543  034120  000207                        RTS     PC
9544
9545                                  ;;*****************************************************************
9546                                  .SBTTL  LOOP ON INTERNAL ERROR
9547                                  ;*      THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER
9548                                  ;*      IS EXPECTED TO SET $LPERR TO THE START OF THE SCOPE LOOP
9549                                  ;*      TO BE EXECUTED ON ERROR.
9550
9551  034122  032777  001000  145010  SCOP1$: BIT     #SW9,@SWR               ;CHECK IF LOOP ON ERROR
9552  034130  001405                        BEQ     5$                      ;NO, CONTINUE
9553  034132  105737  001103                TSTB    $ERFLG                  ;CHECK IF ERROR OCCURRED
9554  034136  001402                        BEQ     5$
9555  034140  013716  001110                MOV     $LPERR,(SP)             ;LOAD ERROR RETURN
9556  034144  000002                5$:     RTI                             ;RETURN
9557                                  .SBTTL  LINE CLOCK CALIBRATE
9558                                  ;*      WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS
9559                                  ;*      TO A  MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS
9560                                  ;*      THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE
9561                                  ;*      LINE CLOCK SIMULATOR.
9562
9563  034146  005037  001660        CLKCAL: CLR     LCLKTK                  ;CLEAR TICK COUNTER
9564  034152  032737  100000  001656        BIT     #LCLKPR,OPTFLG          ;TEST IF CLOCK PRESENT
9565  034160  001004                        BNE     1$                      ;YES - SKIP
9566  034162  012737  000020  001654        MOV     #16.,TIMCNT             ;ELSE PRESET TIMCNT
9567  034170  000410                        BR      2$                      ;AND EXIT
9568  034172  005737  001662        1$:     TST     INTSET                  ;TEST IF INTERRUPT HAS OCCURRED
9569  034176  001005                        BNE     2$                      ;YES- ABORT CALIBRATION
9570  034200  005737  001660                TST     LCLKTK                  ;WAIT FOR CLOCK TICK
9571  034204  001772                        BEQ     1$                      ;NOT YET - LOOP
9572  034206  005037  001660                CLR     LCLKTK                  ;CLEAR TICK COUNT
9573  034212  000207                2$:     RTS     PC      ;RETURN
9574
9575                                  .SBTTL  LINE CLOCK SIMULATION ROUTINE
9576                                  ;*      THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO
9577                                  ;*      DO THIS THE VALUE STORED IN MILCNT IS USED AS THE
9578                                  ;*      BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT
9579                                  ;*      AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE
```

```
                                        ;*      TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE
                                        ;*      LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT
                                        ;*      IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
                                        ;*      TICK).
                                        ;*
                                        ;*      THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND
                                        ;*      AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.
                                        ;*

034214  010046                 MYTIME: MOV     R0,-(SP)        ;SAVE R0
034216  013700  001652                 MOV     MILCNT,R0       ;SET COUNT
034222  005737  001662         1$:     TST     INTSET          ;TEST IF INTERRUPT SET
034226  001012                         BNE     2$              ;YES - SKIP
034230  005300                         DEC     R0              ;DECREMENT COUNT TO ZERO
034232  001373                         BNE     1$
034234  005337  001654                 DEC     TIMCNT          ;DECREMENT TIMCNT
034240  001005                         BNE     2$              ;IF NOT ZERO - EXIT
034242  005237  001660                 INC     LCLKTK          ;ELSE BUMP TICK COUNTER
034246  012737  000020  001654         MOV     #16.,TIMCNT     ;RESET TIME COUNT
034254  012600                 2$:     MOV     (SP)+,R0        ;RESTORE R0
034256  000207                         RTS     PC              ;RETURN

                                        .SBTTL  WAIT FOR INTERRUPT ROUTINE
                                        ;*      THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL
                                        ;*      SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE
                                        ;*      WHILE WAITING FOR INTERRUPT. IF INTERRUPT DOES NOT OCCUR
                                        ;*      IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED
                                        ;*      (BUT NOT PRINTED IN THE ROUTINE)AND THEN RETURNS TO THE
                                        ;*      LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE
                                        ;*      RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL
                                        ;*      BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.

034260  104413                 IWAT8S: SAVREG                  ;ENTRY FOR 8 SECOND WAIT
034262  012700  000764                 MOV     #500.,R0
034266  000463                         BR      WATSRT
034270  104413                 IWAT1M: SAVREG                  ;ENTRY FOR 1 MIN WAIT
034272  012700  007246                 MOV     #3750.,R0
034276  000457                         BR      WATSRT
034300  104413                 IWAT2S: SAVREG  ;ENTRY FOR 2 SECOND WAIT
034302  012700  000200                 MOV     #128.,R0
034306  000453                         BR      WATSRT
034310  104413                 IWAT1S: SAVREG                  ;ENTRY FOR 1 SECOND WAIT
034312  012700  000077                 MOV     #63.,R0
034316  000447                         BR      WATSRT
034320  104413                 IWAT159:        SAVREG          ;ENTRY FOR 160 MS WAIT
034322  012700  000012                 MOV     #10.,R0
034326  000443                         BR      WATSRT
034330  104413                 IWAT144:        SAVREG          ;ENTRY FOR 144 MS WAIT
034332  012700  000011                 MOV     #9.,R0
034336  000437                         BR      WATSRT
034340  104413                 IWAT128:        SAVREG          ;ENTRY FOR 128 MS WAIT
034342  012700  000010                 MOV     #8.,R0
034346  000433                         BR      WATSRT
034350  104413                 IWAT112:        SAVREG          ;ENTRY FOR 112 MS WAIT
034352  012700  000007                 MOV     #7,R0
034356  000427                         BR      WATSRT
```

```
 9636  034360  104413                  IWAT96: SAVREG                  ;ENTRY FOR 96 MS WAIT
 9637  034362  012700  000006                  MOV     #6,R0
 9638  034366  000403                          BR      WATSRT
 9639  034370  104413                  IWAT80: SAVREG                  ;ENTRY FOR 80 MS WAIT
 9640  034372  012700  000005                  MOV     #5,R0
 9641  034376  000417                          BR      WATSRT
 9642  034400  104413                  IWAT64: SAVREG                  ;ENTRY FOR 64 MS WAIT
 9643  034402  012700  000004                  MOV     #4,R0
 9644  034406  000413                          BR      WATSRT
 9645  034410  104413                  IWAT48: SAVREG                  ;ENTRY FOR 48 MS WAIT
 9646  034412  012700  000003                  MOV     #3,R0
 9647  034416  000407                          BR      WATSRT
 9648  034420  104413                  IWAT32: SAVREG                  ;ENTRY FOR 32 MS WAIT
 9649  034422  012700  000002                  MOV     #2,R0
 9650  034426  000403                          BR      WATSRT
 9651  034430  104413                  IWAT16: SAVREG                  ;ENTRY FOR 16 MS WAIT
 9652  034432  012700  000001                  MOV     #1,R0
 9653  034436  012746  000000          WATSRT: MOV     #PR0,-(SP)      ;LOAD PRIORITY 0 ON STACK
 9654  034442  012746  034450                  MOV     #5$,-(SP)       ;LOAD ADDRESS
 9655  034446  000002                          RTI
 9656
 9657  034450  012737  000020  001654  5$:     MOV     #16.,TIMCNT     ;PRESET TIME COUNTER
 9658  034456  004737  034146                  JSR     PC,CLKCAL       ;GO CALIBRATE THE CLOCK
 9659  034462  005737  001662          1$:     TST     INTSET          ;TEST IF INTERRUPT OCCURRED
 9660  034466  001036                          BNE     3$              ;YES - EXIT
 9661  034470  032737  100000  001656          BIT     #LCLKPR,OPTFLG  ;TEST IF KW11-L AVAILABLE
 9662  034476  001002                          BNE     2$              ;YES - SKIP
 9663  034500  004737  034214                  JSR     PC,MYTIME       ;ELSE CALL SIMULATOR
 9664  034504  023700  001660          2$:     CMP     LCLKTK,R0       ;TEST IF ENOUGH TICKS COUNTED
 9665  034510  103764                          BLO     1$              ;NO - LOOP
 9666  034512  104420                          TGETRK                  ;ELSE GET '611 REGS
 9667  034514  013701  001540                  MOV     T.CS1,R1        ;PUT CS1 IN R1- STRIP ALL BUT
 9668  034520  042701  177741                  BIC     #177741,R1      ;COMMAND CODE; INDEX INTO TABLE
 9669  034524  016137  046636  001372          MOV     CMNDLB(R1),DH2N ;AND SELECT HEADER TO IDENTIFY OPERATION
 9670  034532  012737  052427  001370          MOV     #EM4,EM2N       ;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
 9671  034540  013700  001302                  MOV     $TESTN,R0       ;GET NUMBER OF PRESENT TEST
 9672  034544  006300                          ASL     R0              ;SHIFT FOR INDEX
 9673  034546  016037  032304  001264          MOV     $SW08TB(R0),$ESCAPE  ;LOAD ESCAPE TO ABORT TESTS
 9674  034554  162737  000002  001264          SUB     #2,$ESCAPE      ;BUT GO TO NEXT SCOPE CALL
 9675  034562  000402                          BR      4$
 9676  034564  062716  000002          3$:     ADD     #2,(SP)         ;BUMP RETURN AROUND ERROR
 9677  034570  104414                  4$:     RESREG                  ;RESTORE REGS
 9678  034572  000002                          RTI                     ;RETURN
 9679
 9680                                  .SBTTL  "L" REGISTER LOADING ROUTINE
 9681                                  ;*      THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
 9682                                  ;*      THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT
 9683                                  ;*      LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED
 9684                                  ;*      FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.
 9685                                  ;*
 9686                                  ;*      CALL:   JSR     R4,LRLOAD
 9687                                  ;*              COMMAND
 9688                                  ;*              WORD COUNT
 9689                                  ;*              BUS ADDRESS
 9690                                  ;*              .BYTE           SECTOR ADDRESS
 9691                                  ;*              .BYTE           TRACK ADDRESS
```

```
 9692                                   ;*              CYLINDER ADDRESS
 9693
 9694  034574                           LRLOAD:
 9695  034574  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9696  034576  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9697  034600  012701  001600                   MOV     #L.CS1,R1       ;GET ADDRESS OF L REGS
 9698  034604  012700  000004                   MOV     #4,R0           ;PRESET COUNT
 9699  034610  012421                   1$:     MOV     (R4)+,(R1)+     ;MOVE FIRST FOUR WORDS INTO "L" REGS
 9700  034612  005300                           DEC     R0              ;CS1, WC, BA, DA
 9701  034614  001375                           BNE     1$
 9702  034616  013721  001626                   MOV     DRVNUM,(R1)+    ;LOAD DRIVE NUMBER
 9703  034622  005721                           TST     (R1)+           ;BUMP PAST ASOF
 9704  034624  012411                           MOV     (R4)+,(R1)      ;LOAD DCYL
 9705  034626  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
 9706  034630  012600                           MOV     (SP)+,R0        ;;POP STACK INTO R0
 9707  034632  000204                           RTS     R4
 9708
 9709                                   .SBTTL  LOAD RK611 FOR OPERATION
 9710                                   ;*      THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
 9711                                   ;*      THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
 9712                                   ;*      OF THE REGISTER CONTENTS.  L.CS1 IS TRANSFERRED LAST AS IT
 9713                                   ;*      SHOULD BE IF THE GO BIT IS SET.
 9714  034634  005037  001662           LOADRK: CLR     INTSET          ;CLEAR INTERRUPT FLAG.
 9715  034640  010046                           MOV     R0,-(SP)        ;STORE REGISTER
 9716  034642  010146                           MOV     R1,-(SP)
 9717  034644  012700  001602                   MOV     #L.WC,R0        ;GET ADDRESS OF SETUP STORAGE WC
 9718  034650  010201                           MOV     R2,R1           ;GET BASE ADDRESS
 9719  034652  062701  000002                   ADD     #2,R1           ;PUT R1 PAST RKCS1
 9720  034656  012021                           MOV     (R0)+,(R1)+     ;LOAD WORD COUNT
 9721  034660  012021                           MOV     (R0)+,(R1)+     ;LOAD BUS ADDRESS
 9722  034662  012021                           MOV     (R0)+,(R1)+     ;LOAD DISK ADDRESS
 9723  034664  012011                           MOV     (R0)+,(R1)      ;LOAD CS2
 9724  034666  062701  000006                   ADD     #6,R1           ;BUMP R1 TO ASOF
 9725  034672  012021                           MOV     (R0)+,(R1)+     ;LOAD OFFSET
 9726  034674  012021                           MOV     (R0)+,(R1)+     ;LOAD CYLINDER
 9727  034676  062701  000004                   ADD     #4,R1           ;BUMP R1 TO MR1
 9728  034702  011011                           MOV     (R0),(R1)       ;LOAD MR1
 9729  034704  013712  001600                   MOV     L.CS1,(R2)      ;LOAD CS1
 9730  034710  012601                           MOV     (SP)+,R1        ;RESTORE REGISTER
 9731  034712  012600                           MOV     (SP)+,R0
 9732  034714  000002                           RTI                     ;RETURN
 9733
 9734                                   .SBTTL  STORE RK611 REGISTERS
 9735                                   ;*      ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
 9736                                   ;*      WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT STORED IN
 9737                                   ;*      THIS ROUTINE.
 9738
 9739  034716  010046                   GETRK:  MOV     R0,-(SP)        ;STORE REGISTERS TO BE USED
 9740  034720  010146                           MOV     R1,-(SP)
 9741  034722  010346                           MOV     R3,-(SP)
 9742  034724  012700  001540                   MOV     #T.CS1,R0       ;SET POINTER TO TEST TABLE
 9743  034730  010201                           MOV     R2,R1           ;SET POINTER TO RK611 BASE
 9744  034732  012703  000012                   MOV     #10.,R3         ;SET COUNT FOR 1ST 10 REGS
 9745  034736  012120                   1$:     MOV     (R1)+,(R0)+     ;STORE RKCS1 THROUGH RKSPAR
 9746  034740  005303                           DEC     R3
 9747  034742  001375                           BNE     1$
```

```
 9748  034744  062701  000002                ADD     #2,R1           ;BUMP POINTER PAST RKDB
 9749  034750  005720                        TST     (R0)+           ;BUMP POINTER PAST T.RKDB
 9750  034752  012703  000004                MOV     #4,R3           ;SET COUNT FOR LAST 5 REGS
 9751  034756  012120              2$:       MOV     (R1)+,(R0)+     ;STORE RKMR1 THROUTH RKMR3
 9752  034760  005303                        DEC     R3
 9753  034762  001375                        BNE     2$
 9754  034764  012603                        MOV     (SP)+,R3        ;RESTORE REGISTERS
 9755  034766  012601                        MOV     (SP)+,R1
 9756  034770  012600                        MOV     (SP)+,R0
 9757  034772  000002                        RTI                     ;RETURN
 9758                              :SBTTL    BIT COUNTER IN A WORD
 9759                              ;*        THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE STACK
 9760                              ;*        BY THE CALLING ROUTINE.  THE NUMBER OF BITS FOUND IN THE WORD
 9761                              ;*        ARE PASSED BACK ON THE STACK.
 9762
 9763  034774  016637  000002  001256  BITCNT: MOV     2(SP),$TMP16    ;GET WORD WHOSE BITS ARE TO BE COUNTED
 9764  035002  010346                        MOV     R3,-(SP)        ;STORE R3
 9765  035004  005037  001260                CLR     $TMP17          ;CLEAR $TMP16 FOR COUNTING
 9766  035010  012703  000021                MOV     #17.,R3         ;SET A SHIFT COUNTER
 9767  035014  000241                        CLC                     ;CLEAR CARRY
 9768  035016  006037  001256              1$:       ROR     $TMP16          ;ROTATE WORD.
 9769  035022  103407                        BCS     3$              ;WAS BIT SHIFTED OUT A 1?
 9770  035024  005303                        DEC     R3              ;NO - DEC COUNT
 9771  035026  001373                        BNE     1$              ;LOOP IF NOT ZERO
 9772  035030  012603                        MOV     (SP)+,R3        ;RESTORE R3
 9773  035032  013766  001260  000002        MOV     $TMP17,2(SP)    ;PUT COUNT OF BITS ON STACK
 9774  035040  000204                        RTS     R4              ;RETURN
 9775  035042  005237  001260              3$:       INC     $TMP17          ;BUMP COUNT
 9776  035046  000766                        BR      2$              ;LOOP
 9777
 9778                              .SBTTL    MAINTENANCE CLOCK ROUTINE
 9779                              ;*        THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
 9780                              ;*        ADDRESS OFTER THE CALL. THE FIRST BYTE CANTAINS THE NUMBER
 9781                              ;*        OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
 9782                              ;*        CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
 9783                              ;*        CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
 9784
 9785  035050                    MCLOCK:
 9786  035050  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9787  035052  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9788  035054  112400                        MOVB    (R4)+,R0        ;GET NUMBER OF CONTROLLER PHASE ADDRESSES
 9789  035056  112401                        MOVB    (R4)+,R1        ;GET PARTIAL PHASE ADDRESS COUNT
 9790
 9791  035060  006300                        ASL     R0              ;MULTIPLY PHASE ADDRESS  COUNT BY 4
 9792  035062  006300                        ASL     R0
 9793  035064  060100                        ADD     R1,R0           ;ADD IN PARTIALS
 9794  035066  052762  000400  000026  1$:   BIS     #MCLK,RKMR1(R2) ;SET CLOCK
 9795  035074  042762  000400  000026        BIC     #MCLK,RKMR1(R2) ;CLEAR MCLK
 9796  035102  005300                        DEC     R0              ;DECREMENT COUNT
 9797  035104  001370                        BNE     1$              ;LOOP IF NOT ZERO
 9798  035106  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9799  035110  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9800  035112  000204                        RTS     R4
 9801                              .SBTTL    READ AND SORT HEADERS
 9802                              ;*        THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
 9803                              ;*        THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
```

```
9804                                    ;*      ASSCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
9805                                    ;*      READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
9806                                    ;*      EACH READ OF THE DATA BUFFER.
9807                                    ;*
9808                                    ;*      CALL:   JSR     R4,RDSTHD
9809                                    ;*              TCHKOP                  ;RETURN POINT IF CERR IN READ HDR
9810                                    ;*              ERROR   4  ;OR 5, 6, 7
9811                                    ;*              ERROR   13              ;RETURN IF DATA LATE IN DB UNLOAD
9812                                    ;*              ERROR   2               ;RETURN IF TO SLOW OR
9813                                    ;*                                      ;IF HDR 0 NOT FOUND
9814
9815  035114  104413                    RDSTHD: SAVREG
9816  035116  032737  100000  001656            BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
9817  035124  001402                            BEQ     20$             ;NO - SKIP
9818  035126  005077  144544                    CLR     @KWLADD ;RESET INTERRUPT
9819  035132  012700  000026            20$:    MOV     #26,R0          ;PRESET FOR 26 SECTOR FORMAT
9820  035136  032737  010000  001600            BIT     #CFMT,L.CS1     ;IS 24 SECTOR MODE SET?
9821  035144  001402                            BEQ     1$              ;NO - SKIP
9822  035146  012700  000024                    MOV     #24,R0          ;ELSE CHANGE TO 24 SECTOR MODE
9823  035152  012701  060604            1$:     MOV     #IBUFF,R1       ;SET POINTER TO INPUT BUFFER
9824  035156  010005                            MOV     R0,R5           ;SAVE NUMBER OF SECTORS
9825  035160  010104                            MOV     R1,R4           ;SAVE IBUFF ADDRESS
9826  035162  010203                            MOV     R2,R3           ;SET UP POINTER TO RKDB
9827  035164  062703  000024                    ADD     #RKDB,R3
9828  035170  013762  001626  000010            MOV     DRVNUM,RKCS2(R2) ;LOAD DRIVE NUM
9829  035176  013762  001614  000020            MOV     L.DCYL,RKDCYL(R2) ;LOAD CYLINDER NUM
9830  035204  013762  001606  000006            MOV     L.DA,RKDA(R2)   ;LOAD TRACK AND SECTOR
9831
9832  035212  012737  000020  001654  2$:       MOV     #16.,TIMCNT     ;SET TIME COUNTER
9833  035220  005037  001662                    CLR     INTSET          ;CLEAR INTERRUPT FLAG
9834  035224  005037  001660                    CLR     LCLKTK          ;CLEAR TICK COUNTER
9835  035230  013762  001600  000000            MOV     L.CS1,RKCS1(R2) ;LOAD COMMAND
9836
9837  035236  005737  001662            3$:     TST     INTSET          ;TEST IF INT OCCURRED
9838  035242  001020                            BNE     4$              ;YES - SKIP
9839  035244  004737  034214                    JSR     PC,MYTIME       ;WAIT 1 MS
9840  035250  005737  001660                    TST     LCLKTK          ;HAVE WE WAITED 16 MS?
9841  035254  001770                            BEQ     3$              ;NO - LOOP ON WAIT
9842
9843  035256  062766  000006  000006            ADD     #6,6(SP)        ;SET RETURN FOR TO SLOW
9844  035264  104420                            TGETRK                  ;GET RK REGS
9845  035266  012737  052427  001370            MOV     #EM4,EM2N       ;LOAD MESSAGE "TO SLOW/NOT COMPLETE"
9846  035274  012737  047042  001372            MOV     #OPER24,DH2N    ;LOAD COMMAND "READ HEADER" FOR REPORT
9847  035302  000466                            BR      10$             ;SKIP
9848
9849  035304  005762  000000            4$:     TST     RKCS1(R2)       ;TEST FOR CONTROLLER ERROR
9850  035310  100474                            BMI     11$             ;YES - SKIP
9851
9852  035312  011324                            MOV     (R3),(R4)+      ;STORE HEADERS
9853  035314  011324                            MOV     (R3),(R4)+
9854  035316  011324                            MOV     (R3),(R4)+
9855
9856  035320  005762  000010                    TST     RKCS2(R2)       ;TEST IF DATA LATE
9857  035324  100443                            BMI     8$              ;YES - SKIP
9858
9859  035326  005300                            DEC     R0              ;DEC SECTOR COUNT
```

```
 9860  035330  001330                        BNE     2$              ;IF NOT ZERO - LOOP
 9861
 9862  035332  032737  100000  001656        BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
 9863  035340  001403                        BEQ     5$              ;NO - SKIP
 9864  035342  012777  000100  144326        MOV     #BIT6,@KWLADD   ;SET INTERRUPT ENABLE
 9865  035350  032761  000037  000002  5$:   BIT     #37,2(R1)       ;HEADER AT TOP OF BUFF=HEAD 0?
 9866  035356  001413                        BEQ     6$              ;YES - SKIP
 9867  035360  012124                        MOV     (R1)+,(R4)+     ;ELSE MOV THIS HEADER TO BOTTOM
 9868  035362  012124                        MOV     (R1)+,(R4)+
 9869  035364  012124                        MOV     (R1)+,(R4)+
 9870
 9871  035366  005305                        DEC     R5              ;TEST FO INSURE HEAD 0 IS FOUND
 9872  035370  001367                        BNE     5$              ;IF ALL HEADERS NOT CHECKED - LOOP
 9873  035372  012737  055157  001370        MOV     #EM56,EM2N      ;ELSE "HEADER 0 NOT FOUND" MESSAGE
 9874  035400  005037  001372                CLR     DH2N
 9875  035404  000421                        BR      9$              ;SKIP
 9876
 9877  035406  012700  060604        6$:     MOV     #IBUFF,R0       ;GET TOP OF IBUFF
 9878  035412  012120                7$:     MOV     (R1)+,(R0)+     ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
 9879  035414  012120                        MOV     (R1)+,(R0)+
 9880  035416  012120                        MOV     (R1)+,(R0)+
 9881  035420  020004                        CMP     R0,R4           ;ALL HEADERS MOVED?
 9882  035422  001373                        BNE     7$              ;NO - LOOP
 9883
 9884  035424  062766  000010  000006        ADD     #10,6(SP)       ;SET UP FOR GOOD RETURN
 9885  035432  000423                        BR      11$
 9886
 9887  035434  012737  053442  001500  8$:   MOV     #EMDLT,EM13N    ;"DATA LATE SET RESULT OF READ DB"
 9888  035442  012737  055570  057702        MOV     #EMRDB,DF011A
 9889  035450  062766  000004  000006  9$:   ADD     #4,6(SP)        ;SET ERROR RETURN
 9890  035456  104420                        TGETRK                  ;GET RK REGS
 9891  035460  013700  001302        10$:    MOV     $TESTN,R0       ;GET TEST NUMBER
 9892  035464  006300                        ASL     R0              ;SHIFT FOR INDEX
 9893  035466  016037  032304  001264        MOV     $SW08TB(R0),$ESCAPE  ;SET ESCAPE
 9894  035474  162737  000002  001264        SUB     #2,$ESCAPE      ;TO NEXT SCOPE CALL
 9895
 9896  035502  104414                11$:    RESREG
 9897  035504  000204                        RTS     R4
 9898
 9899                                  .SBTTL  GET DRIVE STATUS
 9900                                  ;*      THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
 9901                                  ;*      THROUGH $REG17.   THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
 9902                                  ;*      THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
 9903                                  ;*      IN THE REGISTERS.
 9904                                  ;*
 9905                                  ;*CALL: JSR     R4,GETDRS
 9906                                  ;*      BR      ERROR PROCESSING        ERROR RETURN
 9907                                  ;*      BR      NO ERROR PROCESSING     GOOD RETURN
 9908
 9909  035506  104413                GETDRS: SAVREG
 9910  035510  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
 9911  035516  012700  001202                MOV     #$REG10,R0      ;PRESET ALL STATUS STORAGE TO
 9912  035522  012701  000010                MOV     #8.,R1          ;ALL ONES
 9913  035526  012720  177777        1$:     MOV     #177777,(R0)+
 9914  035532  005301                        DEC     R1
 9915  035534  001374                        BNE     1$
```

```
 9916  035536  012700  001206                        MOV     #$REG12,R0      ;SET POINTER TOREG12 FOR AC1 & BC1
 9917  035542  012701  000001                        MOV     #1,R1           ;PRESET FOR PAIR ONE.
 9918  035546  005037  001230                        CLR     $TMP3           ;CLEAR ERROR SWITCH
 9919  035552  013762  001610  000010  2$:         MOV     L.CS2,RKCS2(R2) ;LOAD DRIVE #
 9920  035560  010162  000026                        MOV     R1,RKMR1(R2)    ;LOAD MR1
 9921  035564  012762  000001  000000                MOV     #B!TO,RKCS1(R2) ;DO SELECT
 9922  035572  012703  000050                        MOV     #40.,R3         ;WAIT FOR A FEW MICRO RECORDS TO
 9923  035576  005303                      3$:       DEC     R3              ;BIT SELECT FINISH
 9924  035600  001376                                BNE     3$
 9925  035602  032762  100000  000000                BIT     #CERR,RKCS1(R2) ;ANY ERROR SET AS A RESULT OF SELECT?
 9926  035610  001415                                BEQ     4$              ;NO - SKIP
 9927  035612  032762  024000  000000                BIT     #CTO!SPAR,RKCS1(R2) ;CHECK IF TIMEOUT OR PARITY ERROR
 9928  035620  001004                                BNE     8$              ;YES - SKIP
 9929  035622  032762  037400  000010                BIT     #37400,RKCS2(R2) ;TEST FOR ERRORS:
 9930                                                                        ;    NED!UPE!MDS!UFE!NEM!PGE
 9931  035630  001405                                BEQ     4$              ;NO - SKIP
 9932  035632  012737  000001  001230  8$:         MOV     #1,$TMP3        ;SET ERROR FLAG
 9933  035640  022020                                CMP     (R0)+,(R0)+     ;BUMP TO LET THAT PAIR STAY ALL 1'S.
 9934  035642  000404                                BR      5$              ;SKIP
 9935  035644  016220  000034              4$:       MOV     RKMR2(R2),(R0)+ ;STORE A WORD
 9936  035650  016220  000036                        MOV     RKMR3(R2),(R0)+ ;STORE B WORD
 9937  035654  012762  100000  000000  5$:         MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERROR IN CONTROLLER
 9938  035662  005701                                TST     R1              ;IS R1 A 0 (LAST TRANSFER, PAIR 0)
 9939  035664  001410                                BEQ     6$              ;YES - SKIP
 9940  035666  005201                                INC     R1              ;ELSE BUMP TO NEXT PAIR
 9941  035670  022701  000004                        CMP     #4,R1           ;PAIR 3  JUST STORED?
 9942  035674  001326                                BNE     2$              ;NO - SKIP
 9943  035676  005001                                CLR     R1              ;ELSE SET TO PAIR 0
 9944  035700  012700  001202                        MOV     #$REG10,R0      ;PRESET POINTER FOR PAIR 0
 9945  035704  000722                                BR      2$              ;GO GET THEM
 9946  035706  104414              6$:       RESREG                          ;EXIT HERE
 9947  035710  005737  001230                        TST     $TMP3           ;ANY ERROR IN STATUS GETTING
 9948  035714  001001                                BNE     7$              ;YES - SKIP
 9949  035716  005724                                TST     (R4)+           ;ELSE BUMP PART ERROR
 9950  035720  000204              7$:       RTS     R4                      ;RETURN
 9951
 9952                                      .SBTTL  SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
 9953                                      ;*      THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
 9954                                      ;*      COMMAND.  CERR AND DI ARE MONITORED FOR A SHORT
 9955                                      ;*      PEIROD OF TIME DURING WHICH THEY SHOULD BOTH RESET.
 9956                                      ;*
 9957                                      ;*      IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
 9958                                      ;*
 9959                                      ;*      IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE
 9960                                      ;*      ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE
 9961                                      ;*      RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN
 9962                                      ;*      SKIPS OVER THE ERROR CALL AND TEST ABORT.
 9963                                      ;*
 9964                                      ;*      THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY
 9965                                      ;*      AN ERROR MESSAGE AND BRANCH TO END OF TEST.  THIS
 9966                                      ;*      IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
 9967                                      ;*      THE TEST.
 9968
 9969  035722                              SSINIT:
 9970  035722  010046                              MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9971  035724  010146                              MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
 9972  035726  012701  000007                    MOV     #7,R1           ;SET CLEAR COUNT
 9973  035732  012700  001600                    MOV     #L.CS1,R0       ;GET ADDRESS OF "L" REGS
 9974  035736  012720  000100                    MOV     #100,(R0)+      ;PRESET CS1
 9975  035742  005020                    7$:     CLR     (R0)+           ;CLEAR THE NEXT
 9976  035744  005301                            DEC     R1              ;COUNT D?
 9977  035746  001375                            BNE     7$              ;NO - LOOP
 9978  035750  012762  000040  000010            MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
 9979  035756  012737  000012  001222            MOV     #10.,$TMP0      ;SET A COUNTER
 9980  035764  016237  000000  001540    1$:     MOV     RKCS1(R2),T.CS1 ;GET CS1
 9981  035772  032737  140000  001540            BIT     #CERR!DI,T.CS1  ;TEST IF ERROR OR DI SET
 9982  036000  001433                            BEQ     2$              ;NO - SKIP TO READY TEST
 9983  036002  005337  001222                    DEC     $TMP0           ;ELSE DECREMENT COUNTER
 9984  036006  001366                            BNE     1$              ;AND LOOP
 9985  036010  032737  100000  001540            BIT     #CERR,T.CS1     ;TEST - IS IT CERR STILL SET
 9986  036016  001404                            BEQ     3$              ;NO  - SKIP TO DI MESSAGE
 9987  036020  012737  052522  001400            MOV     #EM5,EM3N       ;MESSAGE (SUBSYS CLR NOT RESET ERROR)
 9988  036026  000403                            BR      4$
 9989  036030  012737  052566  001400    3$:     MOV     #EM6,EM3N       ;MESSAGE (SUBSYS CLEAR NOT RESET DI)
 9990  036036  104420                    4$:     .GETRK
 9991  036040  013700  001302                    MOV     $TESTN,R0       ;GET PRESENT TEST NUMBER
 9992  036044  006300                            ASL     R0              ;SHIFT FOR INDEX
 9993  036046  016037  032304  001264            MOV     $SW08TBL(R0),$ESCAPE ;LOAD ESCAPE TO ABORT TEST
 9994  036054  162737  000002  001264            SUB     #2,$ESCAPE      ;SET TO NEXT SCOPE CALL
 9995  036062  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 9996  036064  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
 9997  036066  000414                            BR      6$              ;SKIP TO EXIT
 9998  036070  032737  000200  001540    2$:     BIT     #RDY,T.CS1      ;TEST READY SET
 9999  036076  001004                            BNE     5$              ;YES - GOOD EXIT
10000  036100  012737  052645  001400            MOV     #EM7,EM3N       ;MESSAGE (SUBSYS CLR NOT SET READY)
10001  036106  000753                            BR      4$
10002  036110  012601                    5$:     MOV     (SP)+,R1        ;RESTORE REGS
10003  036112  012600                            MOV     (SP)+,R0
10004  036114  062716  000002                    ADD     #2,(SP)         ;GOOD RETURN
10005  036120  000002                    6$:     RTI
10006
10007                                      .SBTTL  WORD COUNT AT END OF OPERATION CHECK
10008                                      ;*      THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR
10009                                      ;*      THE WORD COUNT AGAINST THE SUPPLIED VALUE.  IF UNEQUAL, THE
10010                                      ;*      ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)
10011                                      ;*
10012                                      ;*CALL: JSR     R4,CHKWC
10013                                      ;*      .WORD                   ;EXPECTED WC VALUE
10014
10015  036122  012437  046462      CHKWC:  MOV     (R4)+,EXPWC     ;STORE EXPECTED VALUE
10016                                                              ;
10017  036126  023737  046462  001542      CMP     EXPWC,T.WC      ;COMPARE
10018  036134  001406                      BEQ     1$              ;EQUAL - SKIP
10019  036136  052737  000001  046520      BIS     #WCERR,GRP4ER   ;SET ERROR FLAG
10020  036144  013737  001542  046476      MOV     T.WC,REALWC     ;STORE REAL WORD COUNT
10021  036152  000204              1$:     RTS     R4              ;RETURN.
10022
10023                                      .SBTTL  BUS  ADDRESS AT END OF OPERATION CHECK
10024                                      ;*      THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF
10025                                      ;*      A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE
10026                                      ;*      INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.
10027                                      ;*      IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA
```

```
10028                                   ;*      AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER
10029                                   ;*
10030                                   ;*      IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS
10031                                   ;*      ADDRESS IS THE STARTING BUS ADDRESS.
10032                                   ;*CALL: JSR     R4,CHKBA
10033
10034  036154  010046                  CHKBA:  MOV     R0,-(SP)
10035  036156  010146                          MOV     R1,-(SP)
10036  036160  010346                          MOV     R3,-(SP)
10037  036162  032737  000020  001610          BIT     #BAI,L.CS2      ;TEST IF BAI SET
10038  036170  001404                          BEQ     4$              ;NO - SKIP
10039  036172  013737  001604  046466          MOV     L.BA,EXPBA      ;STORE EXPECTED BA (SAME AS STARTING BA)
10040  036200  000441                          BR      3$
10041  036202  013700  001602          4$:     MOV     L.WC,R0         ;GET INITIAL WORD COUNT
10042  036206  005400                          NEG     R0
10043  036210  113703  001601                  MOVB    L.CS1+1,R3      ;GET BA16 & BA17
10044  036214  042703  177774                  BIC     #177774,R3      ;CLEAR UNWANTED BITS
10045
10046  036220  005700                          TST     R0              ;TEST IF INITIAL WORD COUNT 0
10047  036222  001003                          BNE     6$              ;NO - SKIP
10048  036224  062703  000002                  ADD     #2,R3           ;ADD 2 TO BA16,17 (65K WORD XFER)
10049  036230  000407                          BR      9$              ;SKIP
10050  036232  005700          6$:     TST     R0              ;TEST IF INITIAL WC BIT 15 SET
10051  036234  100001                          BPL     5$              ;NO - SKIP
10052  036236  005203                          INC     R3              ;BUMP BA16,17 (32K WORD XFER)
10053  036240  006300          5$:     ASL     R0              ;SHIFT WORD COUNT TO MAKE MEM ADD CNT
10054  036242  063700  001604                  ADD     L.BA,R0         ;ADD IN START BUFFER ADD
10055  036246  005503                          ADC     R3              ;IF CARRY - ADD INTO BA16,17
10056  036250  013701  001542          9$:     MOV     T.WC,R1         ;GET END OF OPERATION WORD COUNT
10057  036254  001411                          BEQ     1$              ;BRANCH IF ZERO
10058  036256  005401                          NEG     R1
10059  036260  005701                          TST     R1              ;TEST END OPERATION WC BIT 15 SET
10060  036262  100001                          BPL     7$              ;NO - SKIP
10061  036264  005303                          DEC     R3              ;DEC BA 16,17 (32K WC LEFT)
10062  036266  006301          7$:     ASL     R1              ;SHIFT WC TO MAKE MEM ADD CNT
10063  036270  160100                          SUB     R1,R0           ;SUB FROM COMPUTED BUS ADDRESS
10064  036272  005603                          SBC     R3              ;SUB CARRY FROM BA16,17
10065  036274  010337  046464                  MOV     R3,EXPUBA       ;STORE EXPECTED UPPER BA BITS
10066  036300  010037  046466          1$:     MOV     R0,EXPBA
10067  036304  020037  001544          3$:     CMP     R0,T.BA         ;EQUAL TO COMPUTED?
10068  036310  001406                          BEQ     2$              ;YES - SKIP
10069  036312  052737  000004  046520          BIS     #BAERR,GRP4ER   ;ELSE SET BAERR FLAG
10070  036320  013737  001544  046502          MOV     T.BA,REALBA     ;STORE REAL BUS ADDRESS
10071  036326  113703  001541          2$:     MOVB    T.CS1+1,R3      ;GET REAL UPPER BA
10072  036332  042703  177774                  BIC     #177774,R3      ;CLEAR UNWANTED BITS
10073  036336  020337  046464                  CMP     R3,EXPUBA       ;CHECK IF EQUAL
10074  036342  001405                          BEQ     8$              ;YES - SKIP
10075  036344  052737  000002  046520          BIS     #UBAERR,GRP4ER  ;ELSE SET UBA ERROR
10076  036352  010337  046500                  MOV     R3,REALUB       ;STORE REAL UPPER BA
10077  036356  012603          8$:     MOV     (SP)+,R3
10078  036360  012601                  MOV     (SP)+,R1
10079  036362  012600                  MOV     (SP)+,R0
10080  036364  000204                  RTS     R4
10081
10082                                   .SBTTL  CYLINDER,TRACK,SECTOR TEST AT END OF OPERATION
10083                                   ;*      THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA
```

```
10084                                   :*      ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE
10085                                   :*      OPERATION.  THE CONTENTS OF THE LOAD REGISTER STORAGE ARE
10086                                   :*      COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE
10087                                   :*      NECESSARY CALCULATION.
10088                                   :*
10089                                   :*      ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES
10090                                   :*      STORAGE EXPCYL, EXPTRK, EXPSEC.  ALL 3 ARE CHECKED AND
10091                                   :*      IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE
10092                                   :*      ERROR FLAGS FIELD (GRP4ER) IS SET.
10093                                   :*
10094                                   :*CALL: JSR     R4,CHKCTS
10095
10096  036366  104413                  CHKCTS: SAVREG
10097  036370  013700  001602                  MOV     L.WC,R0         ;GET SPECIFIED WORD COUNT
10098  036374  005400                          NEG     R0              ;NEGATE IT
10099  036376  013701  001542                  MOV     T.WC,R1         ;GET END OF OPERATION WORD COUNT
10100  036402  001401                          BEQ     10$             ;IF ZERO - SKIP
10101  036404  005401                          NEG     R1              ;NEGATE IT
10102  036406  160100                  10$:    SUB     R1,R0           ;COMPUTE ACTUAL WORDS TRANSFERRED
10103  036410  005001                          CLR     R1              ;CLEAR R1 FOR COUNTING
10104                                   ;       THE FOLLOWING CODE DETERMINES HOW MAY SECTORS OF DATA HAS BEEN
10105                                   ;       TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE
10106                                   ;       END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR
10107                                   ;       IS CALCULATED.
10108  036412  022700  000400           1$:     CMP     #400,R0
10109  036416  003004                          BGT     2$
10110  036420  005201                          INC     R1
10111  036422  162700  000400                  SUB     #400,R0
10112  036426  000771                          BR      1$
10113  036430  005700                  2$:     TST     R0
10114  036432  001401                          BEQ     3$
10115  036434  005201                          INC     R1
10116                                   ;       AT THIS POINT R1 HAS A COUNT OF THE
10117                                   ;       NUMBER OF FULL SECTOR TRANSFER + 1 IF A
10118                                   ;       PARTIAL SECTOR WAS TRANSFERRED.
10119  036436  012703  000026           3$:     MOV     #26,R3
10120  036442  032737  010000  001600           BIT     #CFMT,L.CS1
10121  036450  001402                          BEQ     4$
10122  036452  012703  000024                  MOV     #24,R3
10123                                   ;       R3 HAS BEEN SET UP WTH THE NUMBER
10124                                   ;       OF SECTORS IN A TRACK FOR THE FORMAT USED
10125  036456  013737  001614  046470   4$:     MOV     L.DCYL,EXPCYL   ;GET STARTING VALUES FOR CYLINDER
10126  036464  113704  001607                  MOVB    L.DT,R4         ;TRACK
10127  036470  042704  177400                  BIC     #177400,R4
10128  036474  113705  001606                  MOVB    L.DS,R5         ;SECTOR
10129  036500  042705  177400                  BIC     #177400,R5
10130  036504  005301                          DEC     R1              ;ADJUST COUNT FOR ZERO DETECT
10131  036506  005205                  5$:     INC     R5              ;BUMP SECTOR COUNT
10132  036510  020503                          CMP     R5,R3           ;DID THIS MAKE SECTOR COUTN > 1 TRACK?
10133  036512  001010                          BNE     6$              ;NO - SKIP
10134  036514  005005                          CLR     R5              ;ELSE CLEAR SECTOR COUNT
10135  036516  005204                          INC     R4              ;BUMP TRACK COUNT
10136  036520  022704  000003                  CMP     #3,R4           ;DID THIS MAKE TRK COUNT > 1 CYLINDER?
10137  036524  001003                          BNE     6$              ;NO - SKIP
10138  036526  005004                          CLR     R4              ;ELSE CLEAR TRACK COUNT
10139  036530  005237  046470                  INC     EXPCYL          ;BUMP CYLINDER COUNT
```

```
10140  036534  005301                  6$:     DEC     R1              ;DEC COUNT
10141  036536  001363                          BNE     5$              ;IF ZERO - EXIT
10142  036540  010437  046474                  MOV     R4,EXPTRK       ;STORE EXPECTED TRACK
10143  036544  010537  046472                  MOV     R5,EXPSEC       ;STORE EXPECTED SECTOR (CYL ALREADY SLOW)
10144  036550  023737  001560  046470          CMP     T.DCYL,EXPCYL   ;TEST IF CYLINDER OK
10145  036556  001403                          BEQ     7$              ;YES - SKIP
10146  036560  052737  000010  046520          BIS     #CYLERR,GRP4ER  ;NO - SET ERROR FLAG
10147  036566  120437  001547          7$:     CMPB    R4,T.DA+1       ;TEST TRACK OK
10148  036572  001403                          BEQ     8$              ;YES - SKIP
10149  036574  052737  000020  046520          BIS     #TRKERR,GRP4ER  ;NO - SET ERROR FLAG
10150  036602  120537  001546          8$:     CMPB    R5,T.DA         ;TEST SECTOR COUNT OK
10151  036606  001403                          BEQ     9$              ;YES - SKIP
10152  036610  052737  000040  046520          BIS     #SECERR,GRP4ER  ;USE SET ERROR FLAG
10153  036616  012700  046476          9$:     MOV     #REALWC,R0
10154  036622  013720  001542                  MOV     T.WC,(R0)+      ;STORE REAL WORD COUNT
10155  036626  013720  001544                  MOV     T.BA,(R0)+      ;STORE REAL BUS ADDRESS
10156  036632  013720  001560                  MOV     T.DCYL,(R0)+    ;STORE REAL CYLINDER ADDRESS
10157  036636  113710  001547                  MOVB    T.DA+1,(R0)     ;STORE REAL TRACK ADDRESS
10158  036642  005720                          TST     (R0)+
10159  036644  113710  001546                  MOVB    T.DA,(R0)       ;STORE REAL SECTOR ADDRESS
10160  036650  104414                          RESREG
10161  036652  000204                          RTS     R4
10162
10163                                  .SBTTL  OPERATION CHECK ROUTINE
10164                                  ;*      THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME  SOFTWARE
10165                                  ;*      ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS:
10166                                  ;*      THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION
10167                                  ;*      FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH
10168                                  ;*      THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN
10169                                  ;*      ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED
10170                                  ;*      BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL
10171                                  ;*      BE THE PROPER ITEM FOR  THE FORMAT REQUIRED BY THIS
10172                                  ;*      ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4,5,6,
10173                                  ;*      7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION
10174                                  ;*      OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT
10175                                  ;*      FORMAT.
10176
10177  036654  104413                  CHKWE:  SAVREG
10178  036656  011600                          MOV     (SP),R0         ;GET POINTER TO ERROR WORDS
10179  036660  012037  001242                  MOV     (R0)+,$TMP10    ;STORE EXPECTED ERROR GROUP 1
10180  036664  012037  001244                  MOV     (R0)+,$TMP11    ;STORE EXPECTED ERROR GROUP 2
10181  036670  012037  001246                  MOV     (R0)+,$TMP12    ;STORE EXPECTED ERROR GROUP 3
10182  036674  010016                          MOV     R0,(SP)         ;STORE RETURN
10183  036676  012737  177777  001250          MOV     #-1,$TMP13      ;SET FLAG - EXPECTED ERROR
10184  036704  000403                          BR      CHKST
10185
10186  036706  104413                  CHKOP:  SAVREG
10187  036710  005037  001250                  CLR     $TMP13          ;RESET EXPECTED ERROR FLAG
10188
10189  036714  104420                  CHKST:  TGETRK                  ;GET 611 REGS IO TRAP
10190  036716  005037  046512                  CLR     GRP1ER          ;CLEAR ERROR FLAGS
10191  036722  005037  046514                  CLR     GRP2ER
10192  036726  005037  046516                  CLR     GRP3ER
10193  036732  005037  046520                  CLR     GRP4ER
10194  036736  005037  046634                  CLR     GPSUMF          ;CLEAR SUMMARY FLAGS
10195  036742  032737  024000  001540          BIT     #CS1ERBIT,T.CS1 ;TEST IF ERROR SET IN CS1
```

```
 10136  036750  001111                          BNE     4$                      ;YES - SKIP
 10197  036752  032737  177400  001550          BIT     #CS2ERBIT,T.CS2 ;TEST IF ERROR SET IN CS2
 10198  036760  001105                          BNE     4$                      ;YES - SKIP
 10199  036762  032737  000070  001552          BIT     #DSERBIT,T.DS   ;TEST IF ERROR SET IN DS
 10200  036770  001101                          BNE     4$                      ;YES - SKIP
 10201  036772  005737  001554                  TST     T.ER                    ;TEST IF ERROR SET IN ER
 10202  036776  001076                          BNE     4$                      ;YES - SKIP
 10203  037000  032737  100000  001540          BIT     #CERR,T.CS1     ;COMBINED ERROR SET?
 10204  037006  001405                          BEQ     9$                      ;NO - SKIP
 10205  037010  052737  100000  046512          BIS     #CERNER,GRP1ER  ;SET ERROR FLAG IN GROUP 1
 10206  037016  000137  037546                  JMP     25$                     ;SKIP
 10207
 10208                                  ;       CODE TO CHECK WORD COUNT, BUFFER ADDRESS, CYLINDER, TRACK,
 10209                                  ;       AND SECTOR AT THE END OF THE OPERATION. THIS IS DONE ONLY
 10210                                  ;       IF CERR NOT SET BY THE OPERATION.
 10211                                  ;
 10212                                  ;       ALL OF THE ABOVE CONDITIONS ARE CHECKED AND A BIT SET FOR
 10213                                  ;       EACH CHECK THAT FAILS. HOWEVER, ONLY ONE ERROR IS REPORTED.
 10214                                  ;       THE ORDER OF PRIORITY FOR REPORTING THE ERROR IS THE ORDER
 10215                                  ;       LISTED ABOVE.
 10216
 10217  037022  005737  001250          9$:     TST     $TMP13                  ;TEST IF ERROR EXPECTED
 10218  037026  001402                          BEQ     62$                     ;NO - SKIP
 10219  037030  000137  037546                  JMP     25$                     ;YES - JUMP
 10220  037034  013700  001540          62$:    MOV     T.CS1,R0                ;GET CS1
 10221  037040  042700  177741                  BIC     #177741,R0              ;CHECK IF OPERATION IS READ DATA,
 10222  037044  022700  000020                  CMP     #20,R0                  ;WRITE DATA, OR WRITE CHECK. IF
 10223  037050  002445                          BLT     3$                      ;NOT, SKIP ALL CHECKING IN GROUP
 10224  037052  022700  000030                  CMP     #30,R0                  ;FOUR
 10225  037056  003042                          BGT     3$
 10226  037060  004437  036122                  JSR     R4,CHKWC                ;CHECK WORD COUNT
 10227  037064  000000                          .WORD   0                       ;EXPECTED WORD COUNT
 10228  037066  004437  036154                  JSR     R4,CHKBA                ;CHECK BUS ADDRESSS
 10229  037072  004437  036366                  JSR     R4,CHKCTS               ;CHECK CYL, TRACK, & SECTOR
 10230  037076  005737  046520                  TST     GRP4ER                  ;ANY GROUP 4 ERRORS?
 10231  037102  001430                          BEQ     3$                      ;NO - SKIP
 10232  037104  016037  046636  057662          MOV     CMNDLB(R0),DF010A ;LOAD ADDRESS OF COMMAND MESSAGE
 10233  037112  013700  046520                  MOV     GRP4ER,R0               ;PUT GROUP 4 ERROR FLAG IN R0
 10234  037116  005001                          CLR     R1                      ;CLEAR R1 FOR INDEX COUNTER
 10235  037120  006200                  1$:     ASR     R0                      ;SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
 10236  037122  103402                          BCS     2$                      ;BE REPORTED, REST ARE IGNORED
 10237  037124  005720                          TST     (R0)+                   ;WHEN AN ERROR BIT IS FOUND,
 10238  037126  000774                          BR      1$                      ;GET THE ERROR MESSAGE ASSOCIATED
 10239  037130  016037  046522  001450  2$:     MOV     GRP4MS(R0),EM10N ;WITH IT AND SET ERROR TABLE ITEM TO
 10240  037136  016037  046462  001202          MOV     EXPWC(R0),$REG10 ;POINT TO THE MESSAGE. LOAD REG10 & 11
 10241  037144  016037  046476  001204          MOV     REALWC(R0),$REG11 ;WITH EXPECTED & IS VALUES
 10242  037152  104414                          RESREG                          ;RESTORE REGISTER
 10243  037154  012776  000010  000000          MOV     #10,2(SP)               ;MAKE THE ERROR CALL POINT TO THE
 10244  037162  000002                          RTI                             ;RIGHT TABLE ENTRY & RETURN.
 10245  037164                          3$:
 10246  037164  104414                          RESREG
 10247  037166  062716  000002                  ADD     #2,(SP)                 ;BUMP RETURN PAST ERROR
 10248  037172  000002                          RTI                             ;RETURN
 10249
 10250                                  ;       THE FOLLOWING CODE BUILDS THE GROUP 1,2, & 3 ERROR WORDS.
 1025:
```

```
10252  037174  012700  046512          4$:     MOV     #GRPIER,R0      ;SET UP GENERAL REGISTER AS POINTER
10253  037200  012701  001540                  MOV     #T.CS1,R1       ;CS1
10254  037204  012703  001550                  MOV     #T.CS2,R3       ;CS2
10255  037210  012704  001552                  MOV     #T.DS,R4        ;DS
10256  037214  012705  001554                  MOV     #T.ER,R5        ;AND ER
10257
10258  037220  051510                          BIS     (R5),(R0)       ;SET ALL BITS IN GRPIER THAT
10259                                                                  ;CORRESPOND TO ERROR BITS IN RKER
10260  037222  042710  120701                  BIC     #ILF!ECH!BSE!HVRC!OPI!DCK,(R0) ; CLEAR ALL THAT DON'T BELONG GRPI
10261
10262  037226  032711  020000                  BIT     #SPAR,(R1)      ;TEST IF SPAR SET
10263  037232  001402                          BEQ     5$              ;NO - SKIP
10264  037234  052710  000001                  BIS     #SPARERR,(R0)   ;SET SPAR ERROR FLAG
10265
10266  037240  032714  000010          5$:     BIT     #ACLO,(R4)      ;TEST ACLO SET
10267  037244  001402                          BEQ     6$              ;NO - SKIP
10268  037246  052710  000100                  BIS     #ACLOERR,(R0)   ;SET ACLO ERROR FLAG
10269
10270  037252  032714  000020          6$:     BIT     #SPDLSS,(R4)    ;TEST SPEED LOSS SET
10271  037256  001402                          BEQ     7$              ;NO - SKIP
10272  037260  052710  000200                  BIS     #SPDERR,(R0)    ;SET SPEED LOSS ERROR FLAG
10273
10274  037264  032714  000040          7$:     BIT     #DROT,(R4)      ;TEST IF DROT SET
10275  037270  001402                          BEQ     8$              ;NO - SKIP
10276  037272  052710  000400                  BIS     #DROTERR,(R0)   ;SET DROT ERROR FLAG
10277
10278  037276  032711  100000          8$:     BIT     #CERR,(R1)      ;TEST CERR ITSELF SET
10279  037302  001002                          BNE     10$             ;YES - SKIP
10280  037304  032710  020000                  BIT     #NCERWE,(R0)    ;SET NO CERR WITH ERROR FLAG
10281
10282  037310  012700  046514          10$:    MOV     #GRP2ER,R0      ;SET POINTER TO GROUP 2 ERROR FLAGS
10283
10284  037314  032715  000100                  BIT     #ECH,(R5)       ;TEST IF ECH SET
10285  037320  001402                          BEQ     11$             ;NO - SKIP
10286  037322  052710  000001                  BIS     #ECHERR,(R0)    ;SET ECH FLAG
10287
10288  037326  032715  100000          11$:    BIT     #DCK,(R5)       ;TEST DCK SET
10289  037332  001402                          BEQ     12$             ;NO - SKIP
10290  037334  052710  000002                  BIS     #DCKERR,(R0)    ;SET DCK ERROR FLAG.
10291
10292  037340  032713  040000          12$:    BIT     #WCE,(R3)       ;TEST WRITE CHECK ERROR
10293  037344  001402                          BEQ     120$            ;NO - SKIP
10294  037346  052710  000004                  BIS     #WCKERR,(R0)    ;SET WCE BIT
10295  037352  032713  100000          120$:   BIT     #DLT,(R3)       ;TEST DATA LATE
10296  037356  001402                          BEQ     13$             ;NO - SKIP
10297  037360  052710  000010                  BIS     #DLTERR,(R0)    ;SET DLT ERROR FLAG
10298
10299  037364  032715  020000          13$:    BIT     #OPI,(R5)       ;TEST OPI SET
10300  037370  001402                          BEQ     14$             ;NO - SKIP
10301  037372  052710  000020                  BIS     #OPIERR,(R0)    ;SET OPI ERROR FLAG
10302
10303  037376  032715  000400          14$:    BIT     #HVRC,(R5)      ;TEST HVRC SET
10304  037402  001402                          BEQ     16$             ;NO - SKIP
10305  037404  052710  000040                  BIS     #HVRCERR,(R0)   ;SET HVRC FLAG
10306
10307  037410  032715  000200          16$:    BIT     #BSE,(R5)       ;TEST BSE ERROR FLAG
```

```
10308  037414  001402                  BEQ     17$             ;NO - SKIP
10309  037416  052710  000100          BIS     #BSERR,(R0)     ;SET BSE FLAG
10310
10311  037422  012700  046516  17$:    MOV     #GRP3ER,R0      ;SET POINTER TO GROUP 3 FLAGS
10312
10313  037426  032713  010000          BIT     #NED,(R3)       ;TEST NED SET
10314  037432  001402                  BEQ     18$             ;NO - SKIP
10315  037434  052710  000001          BIS     #NEDERR,(R0)    ;SET NED FLAG
10316
10317  037440  032711  004000  18$:    BIT     #CTO,(R1)       ;TEST CTO SET
10318  037444  001402                  BEQ     19$             ;NO - SKIP
10319  037446  052710  000002          BIS     #CTOERR,(R0)    ;SET CTO FLAG
10320
10321  037452  032713  000400  19$:    BIT     #UFE,(R3)       ;TEST UFE SET
10322  037456  001402                  BEQ     20$             ;NO - SKIP
10323  037460  052710  000004          BIS     #UFERR,(R0)     ;SET UFE FLAG
10324
10325  037464  032713  001000  20$:    BIT     #MDS,(R3)       ;TEST MDS SET
10326  037470  001402                  BEQ     21$             ;NO - SKIP
10327  037472  052710  000010          BIS     #MDSERR,(R0)    ;SET MDE FLAG
10328
10329  037476  032713  002000  21$:    BIT     #PGE,(R3)       ;TEST PGE SET
10330  037502  001402                  BEQ     22$             ;NO - SKIP
10331  037504  052710  000020          BIS     #PGERR,(R0)     ;SET PGE FLAG
10332
10333  037510  032713  004000  22$:    BIT     #NEM,(R3)       ;TEST NEM SET
10334  037514  001402                  BEQ     23$             ;NO - SKIP
10335  037516  052710  000040          BIS     #NEMERR,(R0)    ;SET NEM FLAG
10336
10337  037522  032713  020000  23$:    BIT     #UPE,(R3)       ;TEST UPE SET
10338  037526  001402                  BEQ     24$             ;NO - SKIP
10339  037530  052710  000100          BIS     #UPERR,(R0)     ;SET UPE FLAG
10340
10341  037534  032715  000001  24$:    BIT     #ILF,(R5)       ;TEST ILF SET
10342  037540  001402                  BEQ     25$             ;NO - SKIP
10343  037542  052710  000200          BIS     #ILFERR,(R0)    ;SET ILF FLAG.
10344
10345                          ;       THE FOLLOWING CODE IS EXECUTED ONLY IF ERRORS WERE EXPECTED.
10346                          ;       THE FLAG IN $TMP13 INDICATES IF
10347                          ;       AN ERROR WAS EXPECTED AND THE CONTENTS OF TMP10,
10348                          ;       TEMP11, & TEMP12 SPECIFY WHICH ERRORS. THESE ARE COMPARED AGAINST
10349                          ;       THE ERRORS FOUND AND STORED IN GRP1ER, GRP2ER, AND GRP3ER.
10350                          ;       THE CONTENTS OF GRP1,2, & 3 ARE MODIFIED TO INDICATE ERRORS THAT
10351                          ;       OCCURRED BUT WERE NOT EXPECTED. THE CONTENTS OF $TMP10,11,
10352                          ;       & 12 ARE MODIFIED TO INDICATE EXPECTED ERRORS THAT DID NOT
10353                          ;       OCCUR. BOTH CONDITIONS CAN EXIST AT THE SAME TIME AND MUST
10354                          ;       BE REPORTED.
10355
10356  037546  005737  001250  25$:    TST     $TMP13          ;CHECK IF AN ERROR WAS EXPECTED
10357  037552  001423                  BEQ     110$            ;NO - SKIP
10358  037554  012704  046512          MOV     #GRP1ER,R4      ;GET ADDRESS OF ERROR
10359  037560  012705  001242          MOV     #$TMP10,R5      ;GET ADDRESS OF EXPECTED ERRORS
10360
10361  037564  011500  26$:    MOV     (R5),R0         ;GET EXPECTED ERROR
10362  037566  011401          MOV     (R4),R1         ;GET GROUP ERROR FLAGS
10363  037570  020001          CMP     R0,R1           ;ARE THEY EQUAL?
```

```
10364  037572  001003                  BNE     27$             ;NO - SKIP
10365  037574  005000                  CLR     R0              ;CLEAR EXPECTED ED
10366  037576  005001                  CLR     R1              ;CLEAR OCCURED
10367  037600  000403                  BR      28$
10368
10369  037602  010003          27$:    MOV     R0,R3           ;STORE EXPECTED ERRORS
10370  037604  040100                  BIC     R1,R0           ;RESET EXPECTED THAT OCCURRED
10371  037606  040301                  BIC     R3,R1           ;RESET OCCURRED THAT EXPECTED
10372  037610  010025          28$:    MOV     R0,(R5)+        ;STORE EXPECTED THAT DID NOT OCCUR
10373  037612  010124                  MOV     R1,(R4)+        ;STORE OCCURRED THAT WERE NOT EXPECTED
10374  037614  022705  001250          CMP     #$TMP13,R5      ;ALL GROUPS CHECKED.
10375  037620  001361                  BNE     26$             ;NO - LOOP
10376
10377                          ;       THE FOLLOWING CODE:
10378                          ;
10379                          ;       A. DETERMINES WHICH FORMAT IS TO BE USED
10380                          ;       B. LOADS THE ADDRESSES OF THE ASCIZ TEXT INTO THE SELECTED
10381                          ;          ERROR TABLE ITEM AND FORMAT FIELD
10382                          ;       C. COUNTS THE NUMBER OF ERRORS THAT MUST BE REPORTED
10383                          ;       D. GETS DRIVE STATUS IF GROUP 1 ERROR.
10384                          ;
10385                          ;       THE DECISION OF WHICH ERROR IS TO BE USED
10386                          ;       IS BASED ON THE ERROR GROUP (OR GROUPS) THAT HAVE
10387                          ;       FLAGS SET. IF ANY BIT IS SET IN GROUP 1,2, OR 3, GROUP 1 FORMAT (ERROR 4 OR 5)
10388                          ;       WILL BE USED; ANY SET IN GROUP 2 OR 3, GROUP 2 (ERROR 6) WILL BE USED; AND A
10389                          ;       FLAG SET IN GROUP 3 ONLY, GROUP 3 (ERROR 7) IS USED.
10390
10391                          ;       THE FORMAT TO BE USED IN THE CONTROLLING FACTOR IN HOW THE
10392                          ;       ERROR TRAP IS CHANGED IN THE MAIN CALL. IF GROUP 1 FORMAT IS
10393                          ;       USED THE ERROR TRAP WILL BE CHANGED TO ERROR 4 OR 5(DEPENDING ON
10394                          ;       AVAILABILITY OF DRIVE STATUS), GROUP 2 FORMAT WILL BE ERROR
10395                          ;       6, AND GROUP 3 WILL BE ERROR 7. ONLY THE LOW ORDER BYTE
10396                          ;       OF THE ERROR TRAP WILL BE ALTERED. THE SP WILL BE
10397                          ;       POINTING TO THE LOCATION THAT CONTAINS THE ERROR CALL TRAP.
10398                          ;
10399                          ;       IF THE STATUS IS READ FROM THE DRIVE WITH NO PROBLEM.
10400                          ;       ERROR 4 IS USED. IF ANY ERROR IS ENCOUNTERED READING
10401                          ;       STATUS, ERROR 5 IS USED. ERROR 5 INCLUDES A WARNING MESSAGE.
10402
10403  037622  005004          110$:   CLR     R4              ;CLEAR COUNTERS
10404  037624  005005                  CLR     R5
10405  037626  012700  001224          MOV     #$TMP1,R0       ;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
10406  037632  012701  001226          MOV     #$TMP2,R1       ;WHERE ASCIZ ADDRESSES GO
10407  037636  012703  046634          MOV     #GPSUMF,R3      ;POINTERS TO GROUP SUMMARY FLAGS
10408  037642  012710  057642          MOV     #DF007A,(R0)    ;PRESET FOR GRP3 ERR MESSAGE BUILD
10409  037646  012711  001442          MOV     #DH7N,(R1)
10410  037652  013746  046516          MOV     GRP3ER,-(SP)    ;GET GROUP 3 ERRORS, PUT ON STACK
10411  037656  004437  034774          JSR     R4,BITCNT       ;GO COUNT NUMBER AT ERRORS
10412  037662  005716                  TST     (SP)            ;ANY ERRORS?
10413  037664  001403                  BEQ     29$             ;NO - SKIP
10414  037666  061605                  ADD     (SP),R5         ;ADD IN ERROR TOTAL
10415  037670  052713  000004          BIS     #GRP3ST,(R3)    ;SET BIT TO INDICATE GROUP 3 ERROR
10416
10417  037674  005726          29$:    TST     (SP)+           ;CLEAR OFF STACK
10418  037676  005737  001250          TST     $TMP13          ;ERROR EXPECTED
10419  037702  001412                  BEQ     31$             ;NO - SKIP
```

```
10420  037704  013746  001246          MOV     $TMP12,-(SP)    ;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
10421  037710  004437  034774          JSR     R4,BITCNT       ;COUNT NUMBER OF ERRORS.
10422  037714  005716                  TST     (SP)            ;WERE THERE ANY
10423  037716  001403                  BEQ     30$             ;NO - SKIP
10424  037720  052713  000040          BIS     #GP3NR,(R3)     ;SET GROUP 3 NOT RECEIVED ERROR FLAG
10425  037724  061604                  ADD     (SP),R4         ;ADD COUNT TO TOTAL THESE
10426
10427  037726  005726          30$:    TST     (SP)+           ;CLEAR OFF STACK
10428  037730  013746  046514  31$:    MOV     GRP2ER,-(SP)    ;GET GROUP 2 ERRORS FOR COUNTING
10429  037734  004437  034774          JSR     R4,BITCNT       ;COUNT BITS
10430  037740  005716                  TST     (SP)            ;ANY SET?
10431  037742  001407                  BEQ     32$             ;NO - SKIP
10432  037744  052713  000002          BIS     #GRP2ST,(R3)    ;SET FLAG FOR GROUP 2 ERRORS
10433  037750  061605                  ADD     (SP),R5         ;ADD INTO TOTAL
10434  037752  012710  057616          MOV     #DF006A,(R0)    ;STORE ADDRESS FOR BUILDING REPORT
10435  037756  012711  001432          MOV     #DH6N,(R1)
10436
10437  037762  005726          32$:    TST     (SP)+           ;CLEAR OFF STACK
10438  037764  005737  001250          TST     $TMP13          ;ANY EXPECTED ERRORS
10439  037770  001416                  BEQ     34$             ;NO - SKIP
10440  037772  013746  001244          MOV     $TMP11,-(SP)    ;GET GROUP 2 NOT RECEIVED ERRORS
10441  037776  004437  034774          JSR     R4,BITCNT       ;COUNT NUMBER OF BITS
10442  040002  005716                  TST     (SP)            ;ANY SET?
10443  040004  001407                  BEQ     33$             ;NO - SKIP
10444  040006  052713  000020          BIS     #GP2NR,(R3)     ;SET FLAG FOR GROUP 2 NOT RECEIVED
10445  040012  061604                  ADD     (SP),R4         ;ADD INTO TOTAL
10446  040014  012710  057616          MOV     #DF006A,(R0)    ;STORE ADDRESS FOR BUILDING REPORT
10447  040020  012711  001432          MOV     #DH6N,(R1)
10448
10449  040024  005726          33$:    TST     (SP)+           ;CLEAR OFF STACK
10450  040026  013746  046512  34$:    MOV     GRP1ER,-(SP)    ;GET GROUP 1 ERROR FLAGS
10451  040032  004437  034774          JSR     R4,BITCNT       ;COUNT THE NUMBER OF BITS
10452  040036  005716                  TST     (SP)            ;ANY SET?
10453  040040  001407                  BEQ     35$             ;NO - SKIP
10454  040042  052713  000001          BIS     #GRP1ST,(R3)    ;SET FLAG FOR GROUP 1 ERRORS SET
10455  040046  061605                  ADD     (SP),R5         ;ADD INTO TOTAL
10456  040050  012710  057532          MOV     #DF004A,(R0)    ;LOAD ADDRESS FOR BUILDING REPORT
10457  040054  012711  001412          MOV     #DH4N,(R1)
10458  040060  005726          35$:    TST     (SP)+           ;CLEAR OFF STACK
10459  040062  005737  001250          TST     $TMP13          ;ANY EXPECTED ERRORS?
10460  040066  001416                  BEQ     60$             ;NO - SKIP
10461  040070  013746  001242          MOV     $TMP10,-(SP)    ;GET GROUP 1 NO RECEIVED ERROR
10462  040074  004437  034774          JSR     R4,BITCNT       ;COUNT # OF BITS
10463  040100  005716                  TST     (SP)            ;ANY SET?
10464  040102  001407                  BEQ     36$             ;NO - SKIP
10465  040104  052713  000010          BIS     #GP1NR,(R3)     ;SET FLAG FOR GROUP 1 NOT RECEIVED
10466  040110  061604                  ADD     (SP),R4         ;ADD INTO TOTAL
10467  040112  012710  057532          MOV     #DF004A,(R0)    ;LOAD ADDRESS FOR BUILDING REPORT
10468  040116  012711  001412          MOV     #DH4N,(R1)
10469  040122  005726          36$:    TST     (SP)+           ;CLEAR OFF STACK.
10470  040124  032713  000011  60$:    BIT     #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERROR
10471  040130  001414                  BEQ     52$             ;NO - SKIP
10472  040132  042713  040000          BIC     #DRSTER,(R3)
10473  040136  004437  035506          JSR     R4,GETDRS
10474  040142  000401                  BR      51$             ;ERROR RETURN
10475  040144  000406                  BR      52$             ;NO ERROR RETURN
```

```
 10476  040146  012710  057562          51$:    MOV     #DF005A,(R0)    ;CHANGE TO FORMAT 5 - STORE ADDRESS
 10477  040152  012711  001422                  MOV     #DH5N,(R1)      ;FOR BUILDING REPORT
 10478  040156  052713  040000                  BIS     #DRSTER,(R3)    ;SET DRIVE STATUS ERROR
 10479  040162                          52$:
 10480
 10481                                  ;       THE ERRORS ARE COUNTED, FLAGS SET TO INDICATE WHICH ERRORS
 10482                                  ;       ARE TO BE REPORTED, AND THE ERROR FORMAT HAS BEEN SELECTED.
 10483                                  ;       THE FOLLOWING CODE WILL TYPE ALL THE ERRORS, LOAD THE
 10484                                  ;       PROPER HEADER MESSAGE ADDRESS IN THE ERROR ITEM TABLE
 10485                                  ;       AND LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE PROPER
 10486                                  ;       DF TABLE.
 10487                                  ;
 10488                                  ;       AT THIS TIME
 10489                                  ;           R5 CONTAINS EITHER THE NUMBER OF ERRORS THAT OCCURRED BUT WERE
 10490                                  ;               NOT EXPECTED OR
 10491                                  ;               THE NUMBER OF ERRORS THE OCCURRED IF NONE WERE EXPECTED
 10492                                  ;           R4 CONTAINS THE NUMBER OF ERRORS THAT WERE EXPECTED BUT
 10493                                  ;               DID NOT OCCUR.
 10494                                  ;           $TMP10 CONTAINS GROUP 1 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
 10495                                  ;           $TMP11 CONTAINS GROUP 2 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
 10496                                  ;           $TMP12 CONTAINS GROUP 3 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
 10497                                  ;           GRP1ER CONTAINS GROUP 1 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
 10498                                  ;           GRP2ER CONTAINS GROUP 2 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
 10499                                  ;           GRP3ER CONTAINS GROUP 3 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
 10500                                  ;       (R1)=#$TMP2 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
 10501                                  ;               DF THAT MUST BE ALTERED TO IDENTIFY THE OPERATION
 10502                                  ;       (R0)=#$TMP1 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
 10503                                  ;               THE ERROR ITEM TABLE THAT MUST BE ALTERED TO PROVIDE A
 10504                                  ;               PROPER MESSAGE TO REPORT.
 10505                                  ;       (R3)=#GRSUMF CONTAIN FLAGS TO INDICATE WHICH OF THE GROUP
 10506                                  ;               ERROR FLAG FIELDS HAVE ERROR BITS STORED.
 10507  040162  032777  020000  140750          BIT     #SW13,@SWR      ;IS REPORT INHIBITED?
 10508  040170  001402                          BEQ     37$             ;NO - SKIP
 10509  040172  000137  040650                  JMP     53$             ;ELSE EXIT
 10510  040176  005737  001250          37$:    TST     $TMP13          ;WERE ERRORS EXPECTED?
 10511  040202  001004                          BNE     38$             ;YES - SKIP
 10512
 10513                                  ;       IF NO ERRORS WERE EXPECTED, $TMP10,11, &12 ARE NOT MEANINGFUL
 10514
 10515  040204  012771  056536  000000          MOV     #DH007,@(R1)    ;HEADER = ERROR IN OPERATION
 10516  040212  000411                          BR      39$
 10517  040214  012771  056357  000000  38$:    MOV     #DH005,@(R1)    ;PRESET HDRMSG = EXPECTED NOT SET
 10518  040222  032713  000070                  BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS?
 10519  040226  001003                          BNE     39$             ;YES - SKIP
 10520  040230  012771  056454  000000          MOV     #DH006,@(R1)    ;SET MESSAGE TO UNEXPECTED ERROR SET
 10521  040236  013701  001540          39$:    MOV     T.CS1,R1        ;GET CS1
 10522  040242  042701  177741                  BIC     #177741,R1      ;CLEAR ALL BUT COMMAND
 10523
 10524  040246  016170  046636  000000          MOV     CMNDLB(R1),@(R0) ;MOVE ADDRESS OF COMMAND MESSAGE
 10525                                                                  ;INTO REPORT
 10526  040254  032713  000007                  BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY GPR ERRORS?
 10527  040260  001462                          BEQ     46$             ;NO - SKIP GPR REPORT
 10528
 10529                                  ;       PRINT ALL THE ERRORS CONTAINED IN THE GPR1,2,3ER(UNEXPECTED ERRORS)
 10530
 10531  040262  013701  046516                  MOV     GRP3ER,R1       ;GET GROUP 3 ERROR FLAGS
```

```
10532  040266  012700  046536                 MOV     #GRP3MS,R0      ;SET POINTER TO GRP3 ERROR MESSAGES
10533  040272  005037  001252          140$:  CLR     $TMP14          ;CLEAR GROUP PRINTING INDICATOR
10534  040276  012737  000021  001254  40$:   MOV     #17.,$TMP15     ;PRESET SHIFT COUNT
10535  040304  000241                         CLC                     ;CLEAR CARRY
10536  040306  006001                 41$:    ROR     R1              ;ROTATE ERROR FLAGS
10537  040310  103406                         BCS     42$             ;WAS BIT SHIFTED OUT SET?
10538  040312  062700  000002         141$:   ADD     #2,R0           ;NO - BUMP POINTER
10539  040316  005337  001254                 DEC     $TMP15          ;DEC SHIFT COUNT
10540  040322  001371                         BNE     41$             ;LOOP IF SHIFT NOT ZERO
10541  040324  000411                         BR      44$
10542  040326  011037  040340         42$:    MOV     (R0),43$        ;GET ERROR MESSAGE ADDRESS FROM TABLE
10543  040332  104401  001273                 TYPE    ,$CRLF          ;TYPE CRLF
10544  040336  104401                         TYPE                    ;TYPE ERROR MESSAGE
10545  040340  000000                 43$:    .WORD                   ;MESSAGE ADDRESS GOES HERE
10546  040342  005305                         DEC     R5              ;DECEREMENT TOTAL ERROR COUNT.
10547  040344  001362                         BNE     141$            ;LOOP IF ZERO
10548  040346  000427                         BR      46$             ;ELSE EXIT GPR ERROR PRINT LOOP
10549
10550  040350  005713                 44$:    TST     (R3)            ;TEST GPSUMF FLAG FOR PRINTING ERROP NOT RECEIVED
10551  040352  100455                         BMI     47$             ;YES - SKIP
10552  040354  005737  001252                 TST     $TMP14          ;PRINTING GROUP 3?
10553  040360  001007                         BNE     45$             ;NO -SKIP
10554  040362  013701  046514                 MOV     GRP2ER,R1       ;ELSE SET TO GROUP 2, GET GRP2ER
10555  040366  012700  046556                 MOV     #GRP2MS,R0      ;& SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
10556  040372  005237  001252                 INC     $TMP14          ;BUMP TO INDICATE PRINTING GROUP 2
10557  040376  000737                         BR      40$             ;GO RESTART PRINT LOOP
10558  040400  022737  000002  001252  45$:   CMP     #2,$TMP14       ;PRINTING GROUP 1?
10559  040406  001407                         BEQ     46$             ;YES - EXIT GPR ERROR PRINT LOOP.
10560  040410  013701  046512                 MOV     GRP1ER,R1       ;ELSE SET TO GROUP 1, GET GROUP 1 ERROR
10561  040414  012700  046574                 MOV     #GRP1MS,R0      ;SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
10562  040420  005237  001252                 INC     $TMP14          ;BUMP TO INDICATE PRINTING GROUP 1
10563  040424  000724                         BR      40$             ;RESTART PRINT LOOP.
10564
10565  040426  005737  001250         46$:    TST     $TMP13          ;EXPECTING ERRORS?
10566  040432  001452                         BEQ     49$             ;NO - SKIP
10567
10568                          ;      PRINT ALL ERRORS CONTAINED IN $TMP10, 11, 12(NOT RECEIVED ERRORS)
10569
10570  040434  005713                         TST     (R3)            ;TEST IF PRINTING NOT RECEIVED ERRORS
10571  040436  100423                         BMI     47$             ;YES - SKIP
10572  040440  032713  000070                 BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS
10573  040444  001445                         BEQ     49$             ;NO - SKIP
10574  040446  032713  000007                 BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY NOT RECEIVED ERRORS?
10575  040452  001404                         BEQ     146$            ;NO - SKIP LABEL FOR UNEXPECTED ERRORS
10576  040454  104401  001273                 TYPE    ,$CRLF          ;TYPE CRLF
10577  040460  104401  056454                 TYPE    ,DH006          ;TYPE HEADER FOR PREVIOUS ERRORS
10578  040464  052737  100000  046634  146$:  BIS     #REPNR,GPSUMF   ;SET PRINTING NOT RECEIVED ERRORS SWITCH
10579  040472  010405                         MOV     R4,R5           ;MOVE TOTAL ERRORS TO R5
10580  040474  013701  001246                 MOV     $TMP12,R1       ;GET GRP3 NOT RECEIVED ERRORS
10581  040500  012700  046536                 MOV     #GRP3MS,R0      ;SET POINTER TO GROUP 3 MESSAGES
10582  040504  000672                         BR      140$            ;GO START PRINT LOOP
10583  040506  005737  001252         47$:    TST     $TMP14          ;PRINTING GROUP 3?
10584  040512  001007                         BNE     48$             ;NO - SKIP
10585  040514  013701  001244                 MOV     $TMP11,R1       ;ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
10586  040520  012700  046556                 MOV     #GRP2MS,R0      ;& SET POINTER TO GROUP 2 MESSAGE TABLE
10587  040524  005237  001252                 INC     $TMP14          ;BUMP TO INDICATE GROUP 2 PRINTING
```

```
10588  040530  000662                        BR      40$             ;GO START PRINT LOOP
10589  040532  022737  000002  001252  48$:  CMP     #2,$TMP14       ;PRINTING GROUP 1?
10590  040540  001407                        BEQ     49$             ;YES - EXIT LOOP
10591  040542  013701  001242                MOV     $TMP10,R1       ;SET POINTER TO GROUP 1 MESSAGE
10592  040546  012700  046574                MOV     #GRP1MS,R0      ;TABLE AND GET GROUP 1 ERRORS.
10593  040552  005237  001252                INC     $TMP14          ;BUMP TO INDICATE GROUP 1 PRINTING
10594  040556  000647                        BR      40$             ;START LOOP AGAIN.
10595
10596  040560  032713  000077        49$:    BIT     #77,(R3)        ;TEST IF ANY ERRORS TO BE REPORTED
10597                                                                ;  GRP1ST!GRP2ST!GRP3ST
10598                                                                ;  GP1NR!GP2NR!GP3NR
10599  040564  001004                        BNE     61$             ;YES - SKIP
10600  040566  104414                        RESREG                  ;ELSE EXIT
10601  040570  062716  000002                ADD     #2,(SP)         ;BUMP FOR GOOD RETURN
10602  040574  000002                        RTI
10603
10604  040576  112776  000007  000000  61$:  MOVB    #7,@(SP)        ;PRESET FOR GROUP 3 ERROR RETURN.
10605  040604  032713  000022                BIT     #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
10606  040610  001403                        BEQ     50$             ;NO - SKIP
10607  040612  112776  000006  000000        MOVB    #6,@(SP)        ;ELSE SET FOR GROUP 2 ERROR RETURN
10608
10609  040620  032713  000011        50$:    BIT     #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
10610  040624  001411                        BEQ     53$             ;NO - SKIP
10611  040626  112776  000004  000000        MOVB    #4,@(SP)        ;ELSE SET FOR GROUP 1 ERROR RETURN.
10612  040634  032713  040000                BIT     #DRSTER,(R3)    ;CHECK IF ERROR GETTING DRIVE STATUS
10613  040640  001403                        BEQ     53$             ;NO - SKIP
10614  040642  112776  000005  000000        MOVB    #5,@(SP)        ;ELSE CHANGE RETURN FORM GROUP 1
10615
10616  040650  013700  001302        53$:    MOV     $TESTN,R0       ;SET UP $ESCAPE TO FORCE
10617  040654  006300                        ASL     R0              ;ABORT TO PRESENT TEST AFTER
10618  040656  016037  032304  001264        MOV     $SW08TB(R0),$ESCAPE ;ERROR IS REPORTED
10619  040664  162737  000002  001264        SUB     #2,$ESCAPE      ;BUT GO TO NEXT SCOPE STATEMENT
10620  040672  104414                        RESREG
10621  040674  000002                        RTI                     ;RETURN
10622
10623                                ;;*****************************************************************
10624                                .SBTTL  BAD SECTOR CHECK
10625                                ;*      THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
10626                                ;*      CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN
10627                                ;*      THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN
10628                                ;*      L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD
10629                                ;*      THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER
10630                                ;*      IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS
10631                                ;*      IS PLACED ON THE STACK AFTER THE ENTIRE
10632                                ;*      FIELD IS SEARCHED.
10633                                ;*
10634                                ;*      CALL:   JSR     R4,BDSRCK
10635                                ;*              <ADDRESS OF FIELD TO BE SEARCHED>
10636                                ;*****************************************************************
10637
10638  040676  012637  001236        BDSRCK: MOV     (SP)+,$TMP6     ;STORE OLD R4 CONTENTS
10639  040702  010437  001240                MOV     R4,$TMP7        ;GET RETURN ADDRESS
10640  040706  011404                        MOV     (R4),R4         ;GET POINTER TO FIELD TO BE CHECKED
10641  040710  005037  001234                CLR     $TMP5           ;CLEAR A COUNTER
10642  040714  005714                1$:     TST     (R4)            ;TEST IF FIELD HAS NO (OR NO MORE)ENTRIES
10643  040716  100417                        BMI     4$              ;YES - EXIT
```

```
10644  040720  023724  001614          CMP     L.DCYL,(R4)+    ;IS THIS ENTRY FOR THIS CYLINDER?
10645  040724  001012                  BNE     3$              ;NO - SKIP
10646  040726  005204                  INC     R4              ;BUMP TO TRACK
10647  040730  123714  001607          CMPB    L.DT,(R4)       ;IS ENTRY FOR THIS TRACK?
10648  040734  001005                  BNE     2$              ;NO - SKIP
10649  040736  005046                  CLR     -(SP)           ;CLEAR STACK LOCATION
10650  040740  114416                  MOVB    -(R4),(SP)      ;PUT SECTOR NUMBER ON STACK
10651  040742  005237  001234          INC     $TMP5           ;BUMP COUNTER
10652  040746  000401                  BR      3$              ;BRANCH
10653
10654  040750  005304          2$:     DEC     R4              ;DECREMENT POINTER TO WORD ALIGN
10655  040752  005724          3$:     TST     (R4)+           ;BUMP TO NEXT ENTRY
10656  040754  000757                  BR      1$              ;TEST NEXT ENTRY
10657
10658  040756  013746  001234  4$:     MOV     $TMP5,-(SP)     ;PUT COUNT ON STACK
10659  040762  013746  001236          MOV     $TMP6,-(SP)     ;PUT OLD R4 CONTENTS BACK ON STACK
10660  040766  013704  001240          MOV     $TMP7,R4        ;SET UP RETURN
10661  040772  005724                  TST     (R4)+           ;BUMP PAST PARAMETER
10662  040774  000204                  RTS     R4              ;RETURN
10663
10664                          ;;*****************************************************************
10665                          .SBTTL  DATA GENERATION AND COMPARE ROUTINE
10666                          ;*  CALLS:      JSR     R4,GENCOM
10667                          ;*              CONTROL WORD
10668                          ;*
10669                          ;*              JSR     R4,GENCOM
10670                          ;*              CONTROL WORD
10671                          ;*              LENGTH
10672                          ;*
10673                          ;*              JSR     R4,GENCOM
10674                          ;*              CONTROL WORD
10675                          ;*              RELOCATION CONSTANT
10676                          ;*              LENGTH
10677                          ;*
10678                          ;*  RETURN:     RTS     R4
10679                          ;*      R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
10680                          ;*      THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
10681                          ;*      CONTROL WORD. THIS IS UNCONDITIONAL.
10682                          ;*
10683                          ;*      THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF
10684                          ;*      THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED.
10685                          ;*      IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE
10686                          ;*      ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.
10687                          ;*
10688                          ;*      THE THIRD CALL IS IDENTICAL TO THE SECOND.
10689                          ;*
10690                          ;*  DEFINITION OF CONTROL WORD:
10691                          ;*      BIT 15 - DO COMPARE OPERATION OF IBUFF (SOURCE) TO OBUFF
10692                          ;*               (DESTINATION). EXPECTED VALUES ARE IN OBUFF (DESTINATION.
10693                          ;*      BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
10694                          ;*      BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
10695                          ;*      BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
10696                          ;*      BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
10697                          ;*      BIT 10 - CLEAR IBUFF TO PATTERN SELECTED.
10698                          ;*      BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
10699                          ;*      BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
```

```
10700
10701                   ;*      BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
10702                   ;*      BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES
10703                   ;*              NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES.
10704                   ;*              OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16.
10705                   ;*
10706                   ;*  EXPLANATION OF CALLS:
10707                   ;*      THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR
10708                   ;*      BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.
10709                   ;*
10710                   ;*      THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED
10711                   ;*      FOR DATA GENERATION OR COMPARE AND FOR IBUFF INITALIZATION.
10712                   ;*
10713                   ;*      THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS
10714                   ;*      USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.
10715                   ;*
10716                   ;*
10717                   ;*  DESCRIPTION:
10718                   ;*
10719                   ;*      THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:
10720                   ;*        A.    BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE
10721                   ;*              ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8)
10722                   ;*              OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR
10723                   ;*               EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE
10724                   ;*              SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK
10725                   ;*              ARE LISTED THEREIN (BIT 9).
10726                   ;*
10727                   ;*        B.    COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT 15). THE
10728                   ;*              CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A
10729                   ;*              HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH
10730                   ;*              WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8)
10731                   ;*              WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT
10732                   ;*              BIT THAT WAS LAST SPECIFIED IN L.CSI. THE HEADERS
10733                   ;*              MAY BE BUILT BEFORE THE COMPARE AS PART OF THE
10734                   ;*              OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE
10735                   ;*              GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
10736                   ;*
10737                   ;*        C.    RESUME COMPARE OPERATION. IF A COMPARE OPERATION
10738                   ;*              DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER
10739                   ;*              BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE
10740                   ;*              RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14
10741                   ;*              SET IN THE CONTROL WORD.
10742                   ;*
10743                   ;*        D.    DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT.
10744                   ;*              MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER
10745                   ;*              SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS
10746                   ;*              MANNER, DATA GENERATION CAN BE MADE TO
10747                   ;*              PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE
10748                   ;*              DATA COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO
10749                   ;*              ANY AREA OF AVAILABLE MEMORY.
10750  040776         GENCOM:
10751  040776  010046          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10752  041000  010146          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10753  041002  010346          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10754  041004  010546          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10755  041006  012400          MOV     (R4)+,R0        ;GET PARAMETER WORD
```

```
10756  041010  012737  000103  001520          MOV     #EM54,EM15N     ;PRESET FOR HEADER COMPARE ERROR
10757  041016  032700  000200                  BIT     #BIT7,R0        ;HEADER OPERATION SPECIFIED?
10758  041022  001005                          BNE     18$             ;YES - SKIP
10759  041024  012737  000132  001520          MOV     #EM55,EM15N     ;CHANGE FOR DATA COMPARE ERROR
10760  041032  000137  041564                  JMP     17$             ;ELSE JUMP TO DATA ROUTINE
10761  041036                          18$:
10762  041036  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10763  041040  032700  001400                  BIT     #BIT8!BIT9,R0   ;MUST HEADERS BE BUILT?
10764  041044  001002                          BNE     19$             ;YES - SKIP
10765  041046  000137  041346                  JMP     11$             ;ELSE JUMP TO HEADER COMPARE
10766  041052  113701  001607          19$:    MOVB    L.DT,R1         ;START HEADER BUILD ROUTINE
10767  041056  013703  001614                  MOV     L.DCYL,R3       ;GET TRACK AND CYL
10768  041062  012705  000005                  MOV     #5,R5           ;SET COUNT TO SHIFT TRACK FOR HDR WORD
10769
10770  041066  006301                  1$:     ASL     R1              ;SHIFT TRACK
10771  041070  005305                          DEC     R5              ;DECREMENT TRACK
10772  041072  001375                          BNE     1$              ;LOOP UNTIL COUNT 0
10773
10774  041074  012704  000026                  MOV     #26,R4          ;PRESET FOR 26 SECTOR MODE
10775  041100  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS IT 24 SECTOR MODE?
10776  041106  001404                          BEQ     2$              ;NO - SKIP
10777  041110  012704  000024                  MOV     #24,R4          ;CHANGE COUNT FOR 24 SECTOR MODE
10778  041114  052701  001000                  BIS     #BIT9,R1        ;SET 24 SECTOR MODE BIT IN WRD 2 OF HDR
10779
10780  041120  052701  140000          2$:     BIS     #BIT15!BIT14,R1 ;SET BS BITS TO INDICATE GOOD SECTOR
10781  041124  012705  062604                  MOV     #OBUFF,R5       ;SET POINTER TO ADDRESS WHERE HEADERS GO
10782  041130  010325                  3$:     MOV     R3,(R5)+        ;INSERT CYLINDER
10783  041132  010125                          MOV     R1,(R5)+        ;INSERT TRACK AND SECTOR
10784  041134  010337  001224                  MOV     R3,$TMP1        ;CALCULATE HVRC WORD
10785  041140  010115                          MOV     R1,(R5)
10786  041142  040137  001224                  BIC     R1,$TMP1
10787  041146  040315                          BIC     R3,(R5)
10788  041150  053725  001224                  BIS     $TMP1,(R5)+     ;COMPLETE HVRC WORD INSERTION
10789
10790  041154  005304                          DEC     R4              ;DECREMENT HEADER COUNT
10791  041156  001402                          BEQ     4$              ;DONE? - YES, SKIP
10792  041160  005201                          INC     R1              ;BUMP SECTOR
10793  041162  000762                          BR      3$              ;LOOP
10794
10795  041164  032700  001000          4$:     BIT     #BIT9,R0        ;MUST HEADERS BE CORRECTED FOR TABLE ENTRIES?
10796  041170  001003                          BNE     5$              ;YES - SKIP
10797  041172  005700                  10$:    TST     R0              ;IS THIS A COMPARE OPERATION?
10798  041174  100464                          BMI     11$             ;YES-GO DO HDR COMPARE
10799  041176  000534                          BR      50$             ;ELSE GET OUT
10800
10801  041200  013737  001640  041236  5$:     MOV     BSF26P,6$       ;PRESET FOR BS FACTORY LIST
10802  041206  012737  100000  001224          MOV     #BIT15,$TMP1    ;SET BIT TO BE RESET IN BAD HEADER
10803  041214  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS THIS 26 SECTOR MODE?
10804  041222  001403                          BEQ     8$              ;YES - SKIP
10805  041224  013737  001636  041236          MOV     BSF24P,6$       ;ELSE CHANGE FOR 24 SECTOR MODE
10806
10807  041232  004437  040676          8$:     JSR     R4,BDSRCK       ;GO CHECK FOR BAD SECTOR THIS ADDRESS
10808  041236  000000                  6$:     .WORD   0               ;POINTER TO FILE TO BE CHECKED GOES HERE
10809  041240  012605                          MOV     (SP)+,R5        ;GET # OF BAD SECTORS THIS PACK ADDRESS
10810  041242  001417                          BEQ     9$              ;SKIP IF ZERO
10811
```

```
 10812  041244  011601                  7$:     MOV     (SP),R1         ;GET 1ST BAD SECTOR NUMBER
 10813  041246  006301                          ASL     R1              ;MULTIPLY SECTOR NUMBER BY 6 TO
 10814  041250  006301                          ASL     R1              ;LOCACATE SECTOR TO BE MARKED BAD
 10815  041252  061601                          ADD     (SP),R1
 10816  041254  062601                          ADD     (SP)+,R1
 10817  041256  062701  000002                  ADD     #2,R1           ;ADD 2 FOR 2ND WORD THAT SECTOR
 10818  041262  043761  001224  062604          BIC     $TMP1,OBUFF(R1) ; CLEAR BIT FOR BAD SECTOR IN HDR
 10819  041270  043761  001224  062606          BIC     $TMP1,OBUFF+2(R1) ; CORRECT THE HVRC BIT
 10820  041276  005305                          DEC     R5              ;DECREMENT BAD SECTOR COUNT
 10821  041300  001361                          BNE     7$              ;LOOP IF NOT ZERO
 10822
 10823  041302  032737  100000  001224  9$:     BIT     #BIT15,$TMP1    ;WERE WE DOING BS FACTORY LIST?
 10824  041310  001730                          BEQ     10$             ;NO - GO CHECK IF COMPARE MUST BE DONE
 10825  041312  012737  040000  001224          MOV     #BIT14,$TMP1    ;ELSE SET BIT TO BE RESET IN BAD HDR
 10826  041320  013737  001644  041236          MOV     BSS26P,6$       ;PRESET POINTER FOR 26 SECTOR MODE
 10827  041326  032737  010000  001600          BIT     #CFMT,L.CS1     ;TEST IF WE ARE DOING 26 SECTOR MODE
 10828  041334  001736                          BEQ     8$              ;YES - SKIP TO START CHECK
 10829  041336  013737  001642  041236          MOV     BSS24P,6$       ;CHANGE POINTER TOR 24 SECTOR MODE
 10830  041344  000732                          BR      8$              ;SKIP TO START CHECK
 10831
 10832                                  ;       START OF COMPARE
 10833  041346  012701  000102          11$:    MOV     #102,R1         ;PRESET FOR 102 WORDS OF HEADER
 10834  041352  032737  010000  001600          BIT     #CFMT,L.CS1     ;CHECK IF 26 SECTOR MODE
 10835  041360  001402                          BEQ     12$             ;YES - SKIP
 10836  041362  012701  000074                  MOV     #74,R1          ;CHANGE TO 74 WORDS OF HEADER
 10837
 10838  041366  012704  060604          12$:    MOV     #IBUFF,R4       ;SET START OF HEADERS TO BE COMPARED
 10839  041372  012705  062604                  MOV     #OBUFF,R5       ;SET START OF GOOD HEADERS
 10840  041376  005003                          CLR     R3              ;CLEAR COUNTER
 10841  041400  032700  040000                  BIT     #BIT14,R0       ;IS THIS A CONTINUATION OF EARLIER COMPARE
 10842  041404  001412                          BEQ     13$             ;NO - SKIP
 10843  041406  013705  001666          28$:    MOV     DESHLD,R5       ;GET VALUES WHERE PREVIOUS CHECK STOPPED
 10844  041412  013704  001670                  MOV     SRCHLD,R4       ;  DESTINATION AND SOURCE
 10845  041416  013703  001672                  MOV     WRDNUM,R3       ;  WORD NUMBER IN ERROR
 10846  041422  013701  001674                  MOV     WRDCNT,R1       ;  WORD COUNT LEFT IN COMPARE
 10847  041426  005701                          TST     R1              ;TEST IF WORD COUNT LEFT = 0
 10848  041430  001417                          BEQ     50$             ;YES - EXIT
 10849
 10850  041432  032700  030000          13$:    BIT     #BIT12!BIT13,R0 ;MEM MANAGE REQUIRED?
 10851  041436  001402                          BEQ     25$             ;NO - SKIP
 10852  041440  005237  177572                  INC     @#SR0           ;TURN IT ON
 10853  041444  022425                  25$:    CMP     (R4)+,(R5)+     ;COMPARE THE WORDS
 10854  041446  001012                          BNE     14$             ;SKIP IF NOT EQUAL
 10855  041450  005203                          INC     R3              ;BUMP WORD NUMBER IN ERROR
 10856  041452  005301                          DEC     R1              ;DEC WORD COUNT LEFT IN COMPARE
 10857  041454  001373                          BNE     25$             ;LOOP IF NOT ZERO
 10858  041456  032700  030000                  BIT     #BIT12!BIT13,R0 ;MEM MANAGE IN USE?
 10859  041462  001402                          BEQ     50$             ;NO - SKIP
 10860  041464  005337  177572                  DEC     @#SR0           ;TURN IT OFF
 10861  041470                          50$:
 10862  041470  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 10863  041472  000427                          BR      16$
 10864
 10865                                  ;       ERROR REPORT PREP AND PARAMETER STORAGE FOR CONTINUATION
 10866
 10867  041474  010537  001666          14$:    MOV     R5,DESHLD       ;STORE DESTINATION
```

```
 10868  041500  010437  001670          MOV     R4,SRCHLD       ;       SOURCE
 10869  041504  014537  001202          MOV     -(R5),$REG10    ;LOAD GOOD WORD FOR REPORT
 10870  041510  014437  001204          MOV     -(R4),$REG11    ;       BAD WORD
 10871  041514  010337  001206          MOV     R3,$REG12       ;       WORD NUMBER
 10872  041520  005301                  DEC     R1              ;DEC COUNT LEFTFOR CONTINUATION
 10873  041522  005203                  INC     R3              ;BUMP BAD WORD NUMBER
 10874  041524  010137  001674          MOV     R1,WRDCNT       ;STORE COUNT LEFT
 10875  041530  010337  001672          MOV     R3,WRDNUM       ;       WORD NUM IN ERROR
 10876  041534  032700  030000          BIT     #BIT12!BIT13,R0 ;MEM MANAGE IS USE?
 10877  041540  001402                  BEQ     15$             ;NO - SKIP
 10878  041542  005337  177572          DEC     @#SR0           ;TURN IT OFF
 10879
 10880  041546                  15$:
 10881  041546  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
 10882  041550  005724                  TST     (R4)+           ;ERROR RETURN
 10883
 10884  041552                  16$:
 10885  041552  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
 10886  041554  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
 10887  041556  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 10888  041560  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
 10889  041562  000204                  RTS     R4
 10890
 10891                          ;       DATA PATTERN PROCESSING ROUTINE
 10892
 10893  041564  032700  040000  17$:    BIT     #BIT14,R0       ;CONTINUE WITH COMPARE?
 10894  041570  001402                  BEQ     29$             ;NO - SKIP
 10895  041572  010446                  MOV     R4,-(SP)        ;STORE RETURN
 10896  041574  000704                  BR      28$             ;GO CONTINUE COMPARE
 10897
 10898  041576  012705  062604  29$:    MOV     #OBUFF,R5       ;GET DESTINATION
 10899  041602  012703  060604          MOV     #IBUFF,R3       ;GET SOURCE
 10900  041606  032700  030000          BIT     #BIT12!BIT13,R0 ;USE MEM MANAGE?
 10901  041612  001412                  BEQ     21$             ;NO - SKIP
 10902
 10903  041614  012437  172354          MOV     (R4)+,@#KIPAR6  ;LOAD PAR FOR RELOCATION
 10904  041620  032700  010000          BIT     #BIT12,R0       ;RELOCATE SOURCE?
 10905  041624  001403                  BEQ     20$             ;NO - SKIP
 10906  041626  012705  140070          MOV     #140070,R5      ;SET DESTINATION TO USE PAR6 + OFFSET
 10907  041632  000402                  BR      21$             ;SKIP
 10908  041634  012703  140070  20$:    MOV     #140070,R3      ;SET SOURCE TO USE PAR6 + OFFSET
 10909
 10910  041640  012401          21$:    MOV     (R4)+,R1        ;STORE COUNT
 10911  041642  010446                  MOV     R4,-(SP)        ;STORE RETURN
 10912  041644  010304                  MOV     R3,R4           ;PUT IN IBUFF POINTER
 10913  041646  005003                  CLR     R3              ;CLEAR R3 FOR WORD NUMBER COUNTER
 10914  041650  032700  000077          BIT     #77,R0          ;ANY DATA PATTERN SPECIFIED?
 10915  041654  001666                  BEQ     13$             ;NO - GO DO COMPARE
 10916
 10917                          ;       START OF GENERATION ROUTINE
 10918
 10919  041656  010537  001666          MOV     R5,DESHLD       ;STORE PARAMETERS FOR COMPARE
 10920  041662  010437  001670          MOV     R4,SRCHLD
 10921  041666  010337  001672          MOV     R3,WRDNUM
 10922  041672  010137  001674          MOV     R1,WRDCNT
 10923
```

```
10924                                   ;       CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
10925                                   ;       MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
10926                                   ;       WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
10927
10928  041676  032700  030000           BIT     #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
10929  041702  001402                   BEQ     33$             ;NO - SKIP
10930  041704  005237  177572           INC     @#SR0           ;TURN IT ON
10931  041710  032700  002000   33$:    BIT     #BIT10,R0       ;GENERATE PATTERN IN IBUFF?
10932  041714  001403                   BEQ     32$             ;NO - SKIP
10933  041716  010446                   MOV     R4,-(SP)        ;ELSE SWAP R4 AND R5
10934  041720  010504                   MOV     R5,R4
10935  041722  012605                   MOV     (SP)+,R5
10936
10937  041724  122700  000001   32$:    CMPB    #1,R0           ;PATTERN 1 (ALL ZEROS)?
10938  041730  001004                   BNE     55$             ;NO - SKIP
10939  041732  005025           30$:    CLR     (R5)+           ;CLEAR WORD IN BUFF
10940  041734  005301                   DEC     R1              ;DEC WORD COUNT
10941  041736  001375                   BNE     30$             ;LOOP UNTIL WORD COUNT ZERO
10942  041740  000550                   BR      22$             ;EXIT BUILD
10943
10944  041742  122700  000007   55$:    CMPB    #7,R0           ;PATTERN 7 (ALL ONES)?
10945  041746  001005                   BNE     56$             ;NO - SKIP
10946  041750  012725  177777   31$:    MOV     #-1,(R5)+       ;LOAD WORD IN BUFF
10947  041754  005301                   DEC     R1              ;DEC WORD COUNT
10948  041756  001374                   BNE     31$             ;LOOP UNTIL WORD COUNT ZERO
10949  041760  000540                   BR      22$             ;EXIT BUILD
10950
10951  041762  122700  000002   56$:    CMPB    #2,R0           ;PATTERN 2 SET UP
10952  041766  001003                   BNE     57$
10953  041770  012703  045662           MOV     #PAT02,R3
10954  041774  000504                   BR      70$
10955
10956  041776  122700  000003   57$:    CMPB    #3,R0           ;PATTERN 3 SET UP
10957  042002  001003                   BNE     58$
10958  042004  012703  045722           MOV     #PAT03,R3
10959  042010  000476                   BR      70$
10960
10961  042012  122700  000004   58$:    CMPB    #4,R0           ;PATTERN 4 SET UP
10962  042016  001003                   BNE     59$
10963  042020  012703  045762           MOV     #PAT04,R3
10964  042024  000470                   BR      70$
10965
10966  042026  122700  000005   59$:    CMPB    #5,R0           ;PATTERN 5 SET UP
10967  042032  001003                   BNE     60$
10968  042034  012703  046022           MOV     #PAT05,R3
10969  042040  000462                   BR      70$
10970
10971  042042  122700  000006   60$:    CMPB    #6,R0           ;PATTERN 6 SET UP
10972  042046  001003                   BNE     61$
10973  042050  012703  046062           MOV     #PAT06,R3
10974  042054  000454                   BR      70$
10975
10976  042056  122700  000010   61$:    CMPB    #10,R0          ;PATTERN 10 SET UP
10977  042062  001003                   BNE     62$
10978  042064  012703  046122           MOV     #PAT10,R3
10979  042070  000446                   BR      70$
```

```
10980
10981  042072  122700  000011      62$:    CMPB    #11,R0          ;PATTERN 11 SET UP
10982  042076  001003                      BNE     63$
10983  042100  012703  046162              MOV     #PAT11,R3
10984  042104  000440                      BR      70$
10985
10986  042106  122700  000012      63$:    CMPB    #12,R0          ;PATTERN 12 SET UP
10987  042112  001003                      BNE     64$
10988  042114  012703  046222              MOV     #PAT12,R3
10989  042120  000432                      BR      70$
10990
10991  042122  122700  000013      64$:    CMPB    #13,R0          ;PATTERN 13 SET UP
10992  042126  001003                      BNE     65$
10993  042130  012703  046262              MOV     #PAT13,R3
10994  042134  000424                      BR      70$
10995
10996  042136  122700  000014      65$:    CMPB    #14,R0          ;PATTERN 14 SET UP
10997  042142  001003                      BNE     66$
10998  042144  012703  046322              MOV     #PAT14,R3
10999  042150  000416                      BR      70$
11000
11001  042152  122700  000015      66$:    CMPB    #15,R0          ;PATTERN 15 SET UP
11002  042156  001003                      BNE     67$
11003  042160  012703  046362              MOV     #PAT15,R3
11004  042164  000410                      BR      70$
11005
11006  042166  122700  000016      67$:    CMPB    #16,R0          ;PATTERN 16 SET UP
11007  042172  001003                      BNE     68$
11008  042174  012703  046422              MOV     #PAT16,R3
11009  042200  000402                      BR      70$
11010
11011  042202  012703  046422      68$:    MOV     #PAT16,R3       ;SET UP FOR 16
11012
11013  042206  032700  004000      70$:    BIT     #BIT11,R0       ;FIRST WORD REPEAT?
11014  042212  001020                      BNE     73$             ;YES - SKIP
11015  042214  010446                      MOV     R4,-(SP)        ;STORE R4
11016  042216  010046                      MOV     R0,-(SP)        ;STORE R0
11017  042220  012700  000020              MOV     #16.,R0         ;PRESET COUNT FOR PATTERN LENGTH
11018  042224  010504                      MOV     R5,R4           ;STORE START OF BUFF
11019
11020  042226  012325              71$:    MOV     (R3)+,(R5)+     ;MOV WORD TO BUFF
11021  042230  005301                      DEC     R1              ;DEC WORD COUNT
11022  042232  001405                      BEQ     74$             ;EXIT IF ZERO
11023  042234  005300                      DEC     R0              ;DEC PAT LENGTH COUNT
11024  042236  001373                      BNE     71$             ;LOOP IF NOT ZERO
11025
11026  042240  012425              72$:    MOV     (R4)+,(R5)+     ;REPEAT PATTERN IN BUFFER
11027  042242  005301                      DEC     R1              ;DEC WORD COUNT
11028  042244  001375                      BNE     72$             ;LOOP UNTIL WORD COUNT ZERO
11029
11030  042246  012600              74$:    MOV     (SP)+,R0        ;RESTORE R0
11031  042250  012604                      MOV     (SP)+,R4        ;RESTORE R4
11032  042252  000403                      BR      22$             ;EXIT BUILD
11033
11034  042254  011325              73$:    MOV     (R3),(R5)+      ;MOV THE SAME WORD INTO BUFFER
11035  042256  005301                      DEC     R1              ;DEC WORD COUNT
```

```
11036  042260  001375                          BNE     73$             ;LOOP UNTIL ZERO
11037
11038  042262  032700  030000          22$:    BIT     #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
11039  042266  001402                          BEQ     34$             ;NO - SKIP
11040  042270  005337  177572                  DEC     @#SR0           ;TURN OFF MEM MANAGEMENT
11041  042274  005700                  34$:    TST     R0              ;IS COMPARE REQUIRED?
11042  042276  100012                          BPL     23$             ;NO - SKIP
11043  042300  013705  001666                  MOV     DESHLD,R5       ;RESTORE COMPARE PARAMETERS
11044  042304  013704  001670                  MOV     SRCHLD,R4
11045  042310  013703  001672                  MOV     WRDNUM,R3
11046  042314  013701  001674                  MOV     WRDCNT,R1
11047  042320  000137  041432                  JMP     13$             ;GO START COMPARE
11048  042324                          23$:
11049  042324  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11050  042326  000137  041552                  JMP     16$             ;GO TO EXIT
11051
11052                                  ;;*****************************************************************
11053                                  .SBTTL  PHASE LOCK LOOP CLOCK ADJUSTMENT ROUNTINE
11054                                  ;*      THIS ROUTINE IS ENTERED VIA A START AT LOCATION 220(8). THE
11055                                  ;*      PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
11056                                  ;*      PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
11057                                  ;*      THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
11058                                  ;*      RESETS DIAGNOSTIC MODE BIT IN MR1. INSTRUCTIONS ON WHERE TO
11059                                  ;*      SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
11060                                  ;*
11061                                  ;*      THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
11062
11063  042332  104401  051565          ADJCLK: TYPE    ,OPR019         ;TYPE ADJUSTMENT INSTRUCTIONS
11064
11065  042336  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
11066  042344  013762  001626  000010          MOV     DRVNUM,RKCS2(R2)        ;SELECT DRIVE
11067  042352  012762  000001  000000          MOV     #1,RKCS1(R2)
11068  042360  032762  000200  000000  5$:     BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
11069  042366  001774                          BEQ     5$
11070  042370  032737  100000  001656          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
11071  042376  001402                          BEQ     1$              ;NO - SKIP
11072  042400  005077  137272                  CLR     @KWLADD         ;CLEAR INTERRUPT ENABLE
11073
11074  042404  012762  000040  000026  1$:     MOV     #DMD,RKMR1(R2)  ;SET DIAG MODE
11075  042412  012701  000014                  MOV     #12.,R1         ;SET A COUNT
11076  042416  005301                  2$:     DEC     R1              ;DEC COUNT
11077  042420  001376                          BNE     2$              ;LOOP UNTIL ZERO
11078  042422  012762  000000  000026          MOV     #0,RKMR1(R2)    ;CLEAR MR1
11079  042430  012701  000014                  MOV     #12.,R1         ;SET COUNT
11080  042434  005301                  3$:     DEC     R1              ;DEC COUNT
11081  042436  001376                          BNE     3$              ;LOOP UNTIL ZERO
11082  042440  000761                          BR      1$              ;RESTART LOOP
11083
11084                                  .SBTTL  CONTROLLED HALT SUBROUTINE
11085                                  ;*      THIS ROUTINE IS ENTERED WHEN A CONTROL C IS TYPED. THE
11086                                  ;*      SUBSYSTEM IS CLEARED, THE DRIVE IS RECALIBRATED, AND, IF
11087                                  ;*      NECESSARY, CERTAIN CYLINDERS ARE REFORMATED. THE REFORMATTING
11088                                  ;*      IS CONTROLLED BY THE LOCATION REFMT WHICH CONTAINS THE ADDRESS
11089                                  ;*      OF THE CYLINDER TO BE REFORMATTED.
11090
11091  042442  012737  000112  001302  CTRHLT: MOV     #$TN,$TESTN             ;SET UP FOR HALT FAIL
```

```
11092
11093  042450  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
11094  042452  104003                  ERROR   3               ;BAD INIT ERROR
11095
11096  042454  113700  001102          MOVB    $TSTNM,R0       ;GET CURRENT TEST NUMBER
11097  042460  042700  177400          BIC     #177400,R0      ;CLEAR UNUSED BITS
11098  042464  022700  000003          CMP     #3,R0           ;TEST IF TEST NUMBER 3
11099  042470  001464                  BEQ     PROGEND         ;GO TO HALT PROG
11100  042472  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
11101  042476  000113                  RECAL                   ;RECAL
11102  042500  000000                  0                       ;0 WORDS
11103  042502  000000                  0                       ;0 IS BUFF ADDRESS
11104  042504     000                  .BYTE   0               ;SECTOR 0
11105  042505     000                  .BYTE   0               ;TRACK 0
11106  042506  000000                  0                       ;CYLINDER 0
11107
11108  042510  104417                  TLOADRK                 ;LOAD RK REGS
11109  042512  104423                  TWAT16                  ;WAIT FOR INTERRUPT
11110  042514  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
11111
11112  042516  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11113  042520  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11114
11115  042522  005037  001662          CLR     INTSET          ;CLEAR INTERRUPT FLAG
11116  042526  104437                  TWAT8S                  ;WAIT FOR SECOND INTERRUPT
11117  042530  104002                  ERROR   2
11118
11119  042532  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11120  042534  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11121
11122  042536  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
11123  042540  104003                  ERROR   3               ;BAD INIT ERROR
11124
11125  042542  005737  001664          TST     REFMT           ;TEST IF REFORMAT REQUIRED
11126  042546  001435                  BEQ     PROGEND         ;NO - GO TO HALT
11127  042550  104401  051730          TYPE    ,OPR020         ;TYPE MESSAGE
11128
11129  042554  004437  034574          JSR     R4,LRLOAD       ;LOAD "L" REGS
11130  042560  000127                  WRHEAD                  ;WRHEAD
11131  042562  177676                  -102                    ;-102 WORDS
11132  042564  062604                  OBUFF                   ;OBUFF IS BUFF ADDRESS
11133  042566     000                  .BYTE   0               ;SECTOR 0
11134  042567     000                  .BYTE   0               ;TRACK 0
11135  042570  000312                  312                     ;CYLINDER 312
11136
11137  042572  005737  001664          TST     REFMT           ;TEST IF CYL 0
11138  042576  100002                  BPL     5$              ;NO - SKIP
11139  042600  005037  001614          CLR     L.DCYL          ;ELSE LOAD FOR CYL 0
11140  042604  004437  040776  5$:     JSR     R4,GENCOM       ;GENERATE HEADERS
11141  042610  001200                  1200
11142
11143  042612  104417                  TLOADRK                 ;LOAD RK REGS
11144  042614  104434                  TWAT159                 ;WAIT FOR INTERRUPT
11145  042616  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
11146
11147  042620  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
```

```
11148  042622  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11149
11150  042624  122737  000002  001607          CMPB    #2,L.DT          ;TEST IF TRACK 2 FORMATTED
11151  042632  001403                          BEQ     PROGEND          ;YES - SKIP
11152  042634  105237  001607                  INCB    L.DT             ;ELSE BUMP TRACK
11153  042640  000761                          BR      5$               ;DO NEXT TRACK
11154
11155  042642  104401  052024          PROGEND:        TYPE    ,OPR021  ;TYPE HALT MESSAGE
11156  042646  012706  001100                  MOV     #STACK,SP        ;CLEAR STACK
11157  042652  105037  001103                  CLRB    $EPFLG           ;CLEAR ERROR FLAG
11158  042656  005037  001264                  CLR     $ESCAPE          ;CLEAR ESCAPE
11159  042662  005737  000042                  TST     @#42             ;TEST IF MONITOR PRESENT
11160  042666  001404                          BEQ     10$              ;NO - SKIP
11161  042670  005037  031270                  CLR     $EOPCT           ;SET FOR END OF PROGRAM
11162  042674  000137  031242                  JMP     $EOP             ;GO TO END OF PASS
11163
11164  042700  000000          10$:    HALT                             ;HALT PROGRAM
11165  042702  000137  001744                  JMP     START1           ;GO TO RESTART IF CONTINUE
11166
11167                                  .SBTTL  HALT FAIL ROUTINE
11168                                  ;*      THIS ROUTINE IS ENTERED IF A HARDWARE ERROR IS DETECTED WHEN
11169                                  ;*      THE CARTRIDGE IS BEING REFORMATTED PRIOR TO HALT.
11170  042706  000240          ABTFAIL:        NOP
11171  042710  104401  052063                  TYPE    ,OPR022          ;TYPE HALT FAIL MESSAGE
11172  042714  000137  042642                  JMP     PROGEND          ;GO STOP PROGRAM
11173                                  .SBTTL  TYPE ROUTINE
11174
11175                                  ;;**************************************************************
11176                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11177                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11178                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11179                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11180                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11181                                  ;*
11182                                  ;*CALL:
11183                                  ;*1) USING A TRAP INSTRUCTION
11184                                  ;*      TYPE    ,MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11185                                  ;*OR
11186                                  ;*      TYPE
11187                                  ;*      MESADR
11188                                  ;*
11189
11190  042720  105737  001157          $TYPE:  TSTB    $TPFLG           ;;IS THERE A TERMINAL?
11191  042724  100002                          BPL     1$               ;;BR IF YES
11192  042726  000000                          HALT                     ;;HALT HERE IF NO TERMINAL
11193  042730  000430                          BR      3$               ;;LEAVE
11194  042732  010046                  1$:     MOV     R0,-(SP)         ;;SAVE R0
11195  042734  017600  000002                  MOV     @2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
11196  042740  122737  000001  001316          CMPB    #APTENV,$ENV     ;;RUNNING IN APT MODE
11197  042746  001011                          BNE     62$              ;;NO,GO CHECK FOR APT CONSOLE
11198  042750  132737  000100  001317          BITB    #APTSPOOL,$ENVM  ;;SPOOL MESSAGE TO APT
11199  042756  001405                          BEQ     62$              ;;NO,GO CHECK FOR CONSOLE
11200  042760  010037  042770                  MOV     R0,61$           ;;SETUP MESSAGE ADDRESS FOR APT
11201  042764  004737  032540                  JSR     PC,$ATY3         ;;SPOOL MESSAGE TO APT
11202  042770  000000          61$:    .WORD   0                        ;;MESSAGE ADDRESS
11203  042772  132737  000040  001317  62$:    BITB    #APTCSUP,$ENVM   ;;APT CONSOLE SUPPRESSED
```

```
11204  043000  001003                    BNE     60$             ;;YES,SKIP TYPE OUT
11205  043002  112046            2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
11206  043004  001005                    BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
11207  043006  005726                    TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
11208  043010  012600            60$:    MOV     (SP)+,R0        ;;RESTORE R0
11209  043012  062716  000002    3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
11210  043016  000002                    RTI                     ;;RETURN
11211  043020  122716  000011    4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
11212  043024  001430                    BEQ     8$
11213  043026  122716  000200            CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
11214  043032  001006                    BNE     5$
11215  043034  005726                    TST     (SP)+           ;;POP  <CR><LF> EQUIV
11216  043036  104401                    TYPE                    ;;TYPE A CR AND LF
11217  043040  001273                    $CRLF
11218  043042  105037  043176            CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
11219  043046  000755                    BR      2$              ;;GET NEXT CHARACTER
11220  043050  004737  043132    5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
11221  043054  123726  001156    6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
11222  043060  001350                    BNE     2$              ;;IF NO GO GET NEXT CHAR.
11223  043062  013746  001154            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
11224                                                            ;;AND THE NULL CHAR.
11225  043066  105366  000001    7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
11226  043072  002770                    BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
11227  043074  004737  043132            JSR     PC,$TYPEC       ;;GO TYPE A NULL
11228  043100  105337  043176            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
11229  043104  000770                    BR      7$              ;;LOOP
11230
11231                                   ;HORIZONTAL TAB PROCESSOR
11232
11233  043106  112716  000040    8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
11234  043112  004737  043132    9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
11235  043116  132737  000007  043176    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
11236  043124  001372                    BNE     9$              ;;TAB STOP
11237  043126  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
11238  043130  000724                    BR      2$              ;;GET NEXT CHARACTER
11239  043132  105777  136012    $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
11240  043136  100375                    BPL     $TYPEC
11241  043140  116677  000002  136004    MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11242  043146  122766  000015  000002    CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
11243  043154  001003                    BNE     1$              ;;BRANCH IF NO
11244  043156  105037  043176            CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
11245  043162  000406                    BR      $TYPEX          ;;EXIT
11246  043164  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
11247  043172  001402                    BEQ     $TYPEX          ;;BRANCH IF YES
11248  043174  105227                    INCB    (PC)+           ;;COUNT THE CHARACTER
11249  043176  000000            $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
11250  043200  000207            $TYPEX: RTS     PC
11251
11252                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11253
11254                                   ;;*****************************************************************
11255                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
11256                                   ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
11257                                   ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
11258                                   ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
11259                                   ;*REPLACED WITH SPACES.
```

```
                                        :*CALL:
                                        :*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        :*      TYPDS                   ;;GO TO THE ROUTINE
                                        $TYPDS:
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
043214  012746  020200                          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
043220  016605  000020                          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
043224  100004                                  BPL     1$              ;;BR IF INPUT IS POS.
043226  005405                                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
043230  112766  000055  000001                  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
043236  005000                          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
043240  012703  043416                          MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
043244  112723  000040                          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
043250  005002                          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
043252  016001  043406                          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
043256  160105                          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
043260  002402                                  BLT     4$              ;;BR IF DONE
043262  005202                                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
043264  000774                                  BR      3$
043266  060105                          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
043270  005702                                  TST     R2              ;;CHECK IF BCD DIGIT=0
043272  001002                                  BNE     5$              ;;FALL THROUGH IF 0
043274  105716                                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
043276  100407                                  BMI     7$              ;;BR IF YES
043300  106316                          5$:     ASLB    (SP)            ;;MSD?
043302  103003                                  BCC     6$              ;;BR IF NO
043304  116663  000001  177777                  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
043312  052702  000060                  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
043316  052702  000040                  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
043322  110223                                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
043324  005720                                  TST     (R0)+           ;;JUST INCREMENTING
043326  020027  000010                          CMP     R0,#10          ;;CHECK THE TABLE INDEX
043332  002746                                  BLT     2$              ;;GO DO THE NEXT DIGIT
043334  003002                                  BGT     8$              ;;GO TO EXIT
043336  010502                                  MOV     R5,R2           ;;GET THE LSD
043340  000764                                  BR      6$              ;;GO CHANGE TO ASCII
043342  105726                          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
043344  100003                                  BPL     9$              ;;BR IF NO
043346  116663  177777  177776                  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
043354  105013                          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
043356  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
043360  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
043362  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
043364  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
043366  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
043370  104401  043416                          TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
043374  016666  000002  000004                  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
043402  012616                                  MOV     (SP)+,(SP)
043404  000002                                  RTI                     ;;RETURN TO USER
043406  023420                          $DTBL:  10000.
043410  001750                                  1000.
```

```
 11316  043412  000144                        100.
 11317  043414  000012                        10.
 11318  043416  000004                SDBLK:  .BLKW   4
 11319                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 11320
 11321                                ;;*****************************************************************
 11322                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 11323                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 11324                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 11325                                ;*CALL:
 11326                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 11327                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
 11328                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 11329                                ;*      .BYTE   M               ;;M=1 OR 0
 11330                                ;*                                      ;;1=TYPE LEADING ZEROS
 11331                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
 11332                                ;*
 11333                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 11334                                ;*$TYPOS OR $TYPOC
 11335                                ;*CALL:
 11336                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 11337                                ;*      TYPON                   ;;CALL FOR TYPEOUT
 11338                                ;*
 11339                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 11340                                ;*CALL:
 11341                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 11342                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
 11343
 11344  043426  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 11345  043432  116637  000001  043651        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 11346  043440  112637  043653                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 11347  043444  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 11348  043450  000406                        BR      $TYPON
 11349  043452  112737  000001  043651  $TYPOC: MOVB   #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 11350  043460  112737  000006  043653        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 11351  043466  112737  000005  043650  $TYPON: MOVB   #5,$OCNT        ;;SET THE ITERATION COUNT
 11352  043474  010346                        MOV     R3,-(SP)        ;;SAVE R3
 11353  043476  010446                        MOV     R4,-(SP)        ;;SAVE R4
 11354  043500  010546                        MOV     R5,-(SP)        ;;SAVE R5
 11355  043502  113704  043653                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 11356  043506  005404                        NEG     R4
 11357  043510  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 11358  043514  110437  043652                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 11359  043520  113704  043651                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 11360  043524  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 11361  043530  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
 11362  043532  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 11363  043534  000404                        BR      3$              ;;GO DO MSB
 11364  043536  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
 11365  043540  006105                        ROL     R5
 11366  043542  006105                        ROL     R5
 11367  043544  010503                        MOV     R5,R3
 11368  043546  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 11369  043550  105337  043652                DECB    $OMODE          ;;TYPE THIS DIGIT?
 11370  043554  100016                        BPL     7$              ;;BR IF NO
 11371  043556  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
```

```
11372  043562  001002                        BNE     4$              ;;TEST FOR 0
11373  043564  005704                        TST     R4              ;;SUPPRESS THIS 0?
11374  043566  001403                        BEQ     5$              ;;BR IF YES
11375  043570  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11376  043572  052703  000060              5$:     BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11377  043576  052703  000040                      BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
11378  043602  110337  043646                      MOVB    R3,8$           ;;SAVE FOR TYPING
11379  043606  104401  043646                      TYPE    ,8$             ;;GO TYPE THIS DIGIT
11380  043612  105337  043650              7$:     DECB    $OCNT           ;;COUNT BY 1
11381  043616  003347                        BGT     2$              ;;BR IF MORE TO DO
11382  043620  002402                        BLT     6$              ;;BR IF DONE
11383  043622  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11384  043624  000744                        BR      2$              ;;GO DO THE LAST DIGIT
11385  043626  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
11386  043630  012604                        MOV     (SP)+,R4        ;;RESTORE R4
11387  043632  012603                        MOV     (SP)+,R3        ;;RESTORE R3
11388  043634  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
11389  043642  012616                        MOV     (SP)+,(SP)
11390  043644  000002                        RTI                     ;;RETURN
11391  043646     000                 8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11392  043647     000                         .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11393  043650     000                 $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
11394  043651     000                 $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
11395  043652  000003                 $OMODE: .WORD   3               ;;NUMBER OF DIGITS TO TYPE
11396                                  .SBTTL  TTY INPUT ROUTINE
11397
11398                                  ;;*****************************************************************
11399                                  .ENABL  LSB
11400  043654  000000                 $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
11401  043656  000000                 $TKQIN: .WORD   0               ;;INPUT POINTER
11402  043660  000000                 $TKQOUT: .WORD  0               ;;OUTPUT POINTER
11403  043662  000001                 $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
11404          043663                 $TKQEND=.
11405          043664                 .EVEN
11406
11407                                  ;*TK INITIALIZE ROUTINE
11408                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
11409                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
11410                                  ;
11411                                  ;*CALL:
11412                                  ;*      JSR     PC,$TKINT
11413                                  ;*      RETURN
11414                                  ;
11415  043664  005037  043654         $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
11416  043670  012737  043662  043656          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
11417  043676  013737  043656  043660          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
11418  043704  012737  043734  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
11419  043712  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
11420  043720  005777  135222                  TST     @$TKB           ;;CLEAR DONE FLAG
11421  043724  012777  000100  135212          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
11422  043732  000207                          RTS     PC              ;;RETURN TO CALLER
11423
11424                                  ;*TK SERVICE ROUTINE
11425                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
11426                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
11427                                  ;*IT IN THE QUEUE.
```

```
11428                                   ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
11429                                   ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT).
11430                                   ;
11431  043734  117746  135206          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
11432  043740  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
11433  043744  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
11434  043750  001007                          BNE     1$              ;;BRANCH IF NO
11435  043752  104401  045050                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
11436  043756  004737  043664                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
11437  043762  005726                          TST     (SP)+           ;;CLEAN UP STACK
11438  043764  000137  042442                  JMP     CTRHLT          ;;CONTROL C RESTART
11439  043770  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
11440  043774  001004                          BNE     2$              ;;BRANCH IF NO
11441  043776  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
11442  044004  001500                          BEQ     6$              ;;GO TO SWR CHANGE
11443
11444  044006                          2$:
11445  044006  022737  000001  043654          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
11446  044014  001004                          BNE     3$              ;;BRANCH IF NO
11447  044016  104401  001266                  TYPE    ,$BELL          ;;RING THE TTY BELL
11448  044022  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
11449  044024  000451                          BR      5$              ;;EXIT
11450  044026  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
11451  044032  001021                          BNE     32$             ;;BRANCH IF NO
11452  044034  005077  135104                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11453  044040  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
11454  044042  105777  135076          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
11455  044046  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
11456  044050  117746  135072                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
11457  044054  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11458  044060  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
11459  044064  001366                          BNE     31$             ;;BRANCH IF NO
11460  044066  012777  000100  135050          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
11461  044074  000002                          RTI                     ;;RETURN
11462  044076  005237  043654          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
11463  044102  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
11464  044106  002405                          BLT     4$              ;;BRANCH IF YES
11465  044110  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
11466  044114  003002                          BGT     4$              ;;BRANCH IF YES
11467  044116  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
11468  044122  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEJE
11469  044126  005237  043656                  INC     $TKQIN          ;;UPDATE THE POINTER
11470  044132  023727  043656  043663          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
11471  044140  001003                          BNE     5$              ;;BRANCH IF NO
11472  044142  012737  043662  043656          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
11473  044150  000002                  5$:     RTI                     ;;RETURN
11474
11475                                   ;;*****************************************************************
11476                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11477                                   ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
11478                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11479                                   ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11480  044152  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
11481  044160  001124                          BNE     15$             ;;EXIT IF NOT
11482  044162  105777  134756                  TSTB    @$TKS           ;;IS A CHAR WAITING?
11483  044166  100121                          BPL     15$             ;;IF NOT, EXIT
```

```
11484  044170  117746  134752                  MOVB    @$TKB,-(SP)     ;;YES
11485  044174  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
11486  044200  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
11487  044204  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
11488                                                                  ;;AND EXIT
11489
11490                                  ;;**************************************************************
11491                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11492                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11493                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11494  044206  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
11495  044214  001674                          BEQ     2$              ;;BRANCH IF YES
11496  044216  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
11497  044220  004737  043664                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
11498  044224  005077  134714                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11499  044230  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
11500
11501  044236  104401  045062                  TYPE    .$CNTLG         ;;ECHO THE CONTROL-G (^G)
11502  044242  104401  045067          $GTSWR: TYPE    .$MSWR          ;;TYPE CURRENT CONTENTS
11503  044246  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
11504  044252  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII (ALL DIGITS)
11505  044254  104401  045100                  TYPE    .$MNEW          ;;PROMPT FOR NEW SWR
11506  044260  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
11507  044262  005046                          CLR     -(SP)           ;;THE NEW SWR
11508  044264  105777  134654          7$:     TSTB    @$TKS           ;;CHAR THERE?
11509  044270  100375                          BPL     7$              ;;IF NOT TRY AGAIN
11510
11511  044272  117746  134650                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
11512  044276  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11513
11514  044302  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
11515  044306  001015                          BNE     9$              ;;BRANCH IF NOT
11516  044310  104401  045050                  TYPE    .$CNTLC         ;;YES, ECHO CONTROL-C (^C)
11517  044314  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
11518  044320  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
11519  044326  001003                          BNE     8$              ;;BRANCH IF NO
11520  044330  012777  000100  134606          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
11521  044336  000137  042442          8$:     JMP     CTRHLT          ;;CONTROL-C RESTART
11522
11523
11524  044342  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
11525  044346  001005                          BNE     10$             ;;BRANCH IF NOT
11526  044350  104401  045055                  TYPE    .$CNTLU         ;;YES, ECHO CONTROL-U (^U)
11527  044354  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
11528  044360  000737                          BR      19$             ;;LET'S TRY IT AGAIN
11529
11530
11531  044362  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
11532  044366  001022                          BNE     15$             ;;BRANCH IF NO
11533  044370  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
11534  044374  001403                          BEQ     11$             ;;BRANCH IF YES
11535  044376  016677  000002  134534          MOV     2(SP),@SWR      ;;SAVE NEW SWR
11536  044404  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
11537  044410  104401  001273          14$:    TYPE    .$CRLF          ;;ECHO <CR> AND <LF>
11538  044414  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
11539  044422  001003                          BNE     15$             ;;BRANCH IF NOT
```

```
11540  044424  012777  000100  134512          MOV     #100,@#TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
11541  044432  000002                  15$:     RTI                     ;;RETURN
11542  044434  004737  043132          16$:     JSR     PC,$TYPEC       ;;ECHO CHAR
11543  044440  021627  000060                   CMP     (SP),#60        ;;CHAR < 0?
11544  044444  002420                           BLT     18$             ;;BRANCH IF YES
11545  044446  021627  000067                   CMP     (SP),#67        ;;CHAR > 7?
11546  044452  003015                           BGT     18$             ;;BRANCH IF YES
11547  044454  042726  000060                   BIC     #60,(SP)+       ;;STRIP-OFF ASCII
11548  044460  005766  000002                   TST     2(SP)           ;;IS THIS THE FIRST CHAR
11549  044464  001403                           BEQ     17$             ;;BRANCH IF YES
11550  044466  006316                           ASL     (SP)            ;;NO, SHIFT PRESENT
11551  044470  006316                           ASL     (SP)            ;;   CHAR OVER TO MAKE
11552  044472  006316                           ASL     (SP)            ;;   ROOM FOR NEW ONE.
11553  044474  005266  000002          17$:     INC     2(SP)           ;;KEEP COUNT OF CHAR
11554  044500  056616  177776                   BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
11555  044504  000667                           BR      7$              ;;GET THE NEXT ONE
11556  044506  104401  001272          18$:     TYPE    ,$QUES          ;;TYPE ?<CR><LF>
11557  044512  000720                           BR      20$             ;;SIMULATE CONTROL-U
11558                                  .DSABL   LSB
11559
11560
11561                                  ;;*****************************************************************
11562                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11563                                  ;*CALL:
11564                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
11565                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
11566                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
11567                                  ;
11568
11569  044514  011646                  $RDCHR:  MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
11570  044516  016666  000004  000002           MOV     4(SP),2(SP)     ;;THE PS
11571  044524  005066  000004                   CLR     4(SP)           ;;GET READY FOR A CHARACTER
11572  044530  005046                           CLR     -(SP)           ;;PUT NEW PS ON STACK
11573  044532  012746  044540                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
11574  044536  000002                           RTI                     ;;POP NEW PC AND PS
11575  044540                          64$:
11576  044540  005737  043654          1$:      TST     $TKCNT          ;;WAIT ON A CHARACTER
11577  044544  001775                           BEQ     1$
11578  044546  005337  043654                   DEC     $TKCNT          ;;DECREMENT THE COUNTER
11579  044552  117766  177102  000004           MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
11580  044560  005237  043660                   INC     $TKQOUT         ;;UPDATE THE POINTER
11581  044564  023727  043660  043663           CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
11582  044572  001003                           BNE     2$              ;;BRANCH IF NO
11583  044574  012737  043662  043660           MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
11584  044602  000002                  2$:      RTI                     ;;RETURN
11585                                  ;;*****************************************************************
11586                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
11587                                  ;*CALL:
11588                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
11589                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11590                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
11591
11592  044604  010346                  $RDLIN:  MOV     R3,-(SP)        ;;SAVE R3
11593  044606  005046                           CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
11594  044610  012703  045040          1$:      MOV     #$TTYIN,R3      ;;GET ADDRESS
11595  044614  022703  045050          2$:      CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
```

```
11596  044620  101456                          BLOS    4$              ;;BR IF YES
11597  044622  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
11598  044624  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
11599  044626  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
11600  044632  001022                          BNE     5$              ;;BR IF NO
11601  044634  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
11602  044636  001007                          BNE     6$              ;;BR IF NO
11603  044640  112737  000134  045036         MOVB    #'\,9$          ;;TYPE A BACK SLASH
11604  044646  104401  045036                  TYPE    ,9$
11605  044652  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
11606  044656  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
11607  044660  020327  045040                  CMP     R3,#STTYIN      ;;STACK EMPTY?
11608  044664  103434                          BLO     4$              ;;BR IF YES
11609  044666  111337  045036                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
11610  044672  104401  045036                  TYPE    ,9$             ;;GO TYPE
11611  044676  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
11612  044700  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
11613  044702  001406                          BEQ     7$              ;;BR IF NO
11614  044704  112737  000134  045036         MOVB    #'\,9$          ;;TYPE A BACK SLASH
11615  044712  104401  045036                  TYPE    ,9$
11616  044716  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
11617  044720  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
11618  044724  001003                          BNE     9$              ;;BR IF NO
11619  044726  104401  045055                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
11620  044732  000726                          BR      1$              ;;GO START OVER
11621  044734  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
11622  044740  001011                          BNE     3$              ;;BRANCH IF NO
11623  044742  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
11624  044744  104401  001273                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
11625  044750  104401  045040                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
11626  044754  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
11627  044756  104401  001272          4$:     TYPE    ,$QUES          ;;TYPE A '?'
11628  044762  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
11629  044764  111337  045036          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
11630  044770  104401  045036                  TYPE    ,9$
11631  044774  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
11632  045000  001305                          BNE     2$              ;;LOOP IF NOT RETURN
11633  045002  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
11634  045006  104401  001274                  TYPE    ,$LF            ;;TYPE A LINE FEED
11635  045012  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
11636  045014  012603                          MOV     (SP)+,R3        ;;RESTORE R3
11637  045016  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
11638  045020  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
11639  045026  012766  045040  000004          MOV     #STTYIN,4(SP)
11640  045034  000002                          RTI                     ;;RETURN
11641  045036     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
11642  045037     000                          .BYTE   0               ;;TERMINATOR
11643  045040  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
11644  045050  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
11645  045055     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
11646  045062  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
11647  045067     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR =
11648  045074  036440  000040
11649  045100  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
11650  045106  020075     000
11651          045112                  .EVEN
```

```
11652                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
11653
11654                                   ;;*****************************************************************
11655                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
11656                                   ;*CHANGE IT TO BINARY.
11657                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
11658                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11659                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11660                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11661                                   ;*CALL:
11662                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
11663                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
11664                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
11665
11666  045112  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
11667  045114  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
11668  045122  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11669  045124  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11670  045126  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11671  045130  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
11672  045132  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
11673  045134  010037  045240                  MOV     R0,5$           ;;AND SAVE IT
11674  045140  005001                          CLR     R1              ;;CLEAR DATA WORD
11675  045142  005002                          CLR     R2
11676  045144  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
11677  045146  001420                          BEQ     3$              ;;IF ZERO GET OUT
11678  045150  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
11679  045154  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
11680  045156  122716  000067                  CMPB    #'7,(SP)
11681  045162  002423                          BLT     4$
11682  045164  006301                          ASL     R1              ;;*2
11683  045166  006102                          ROL     R2
11684  045170  006301                          ASL     R1              ;;*4
11685  045172  006102                          ROL     R2
11686  045174  006301                          ASL     R1              ;;*8
11687  045176  006102                          ROL     R2
11688  045200  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
11689  045204  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
11690  045206  000756                          BR      2$              ;;LOOP
11691  045210  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
11592  045212  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
11693  045216  010237  045250                  MOV     R2,$HIOCT
11694  045222  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11695  045224  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11696  045226  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11697  045230  000002                          RTI                     ;;RETURN
11698  045232  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
11699  045234  105010                          CLRB    (R0)            ;;SET A TERMINATOR
11700  045236  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
11701  045240  000000                  5$:     .WORD   0
11702  045242  104401  001272                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
11703  045246  000730                          BR      1$              ;;TRY AGAIN
11704  045250  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
11705                                   .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
11706
11707                                   ;;*****************************************************************
```

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

045252                                  $SAVREG:
045252  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
045254  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
045256  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
045260  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
045262  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
045264  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
045266  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
045272  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
045276  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
045302  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
045306  000002                                  RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
045310                                  $RESREG:
045310  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
045314  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
045320  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
045324  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
045330  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
045332  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
045334  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
045336  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
045340  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
045342  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
045344  000002                                  RTI
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;**************************************************************
                                        ;POWER DOWN ROUTINE
045346  012737  045506  000024          $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
045354  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;PRIO:7
045362  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
045364  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
045366  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
045370  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
045372  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
045374  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
045376  017746  133536                          MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
045402  010637  045512                          MOV     SP,$SAVR6       ;;SAVE SP
```

```
11764  045406  012737  045420  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
11765  045414  000000                          HALT
11766  045416  000776                          BR      .-2              ;;HANG UP
11767
11768                                  ;;*****************************************************************
11769                                  ;*POWER UP ROUTINE
11770  045420  012737  045506  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
11771  045426  013706  045512                  MOV     $SAVR6,SP        ;;GET SP
11772  045432  005037  045512                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
11773  045436  005237  045512          1$:     INC     $SAVR6           ;;WAIT FOR THE INC
11774  045442  001375                          BNE     1$               ;;OF  WORD
11775  045444  012677  133470                  MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
11776  045450  012605                          MOV     (SP)+,R5         ;;POP STACK INTO R5
11777  045452  012604                          MOV     (SP)+,R4         ;;POP STACK INTO R4
11778  045454  012603                          MOV     (SP)+,R3         ;;POP STACK INTO R3
11779  045456  012602                          MOV     (SP)+,R2         ;;POP STACK INTO R2
11780  045460  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
11781  045462  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
11782  045464  012737  045346  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
11783  045472  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
11784  045500  104401                          TYPE                     ;REPORT THE POWER FAILURE
11785  045502  045514          $PWRMG: .WORD   $POWER                   ;;POWER FAIL MESSAGE POINTER
11786  045504  000002                  RTI
11787  045506  000000          $ILLUP: HALT                             ;;THE POWER UP SEQUENCE WAS STARTED
11788  045510  000776                  BR      .-2                      ;; BEFORE THE POWER DOWN WAS COMPLETE
11789  045512  000000          $SAVR6: 0                                ;;PUT THE SP HERE
11790  045514  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
11791  045522  000122
11792                                          .EVEN
11793                                  .SBTTL  TRAP DECODER
11794
11795                                  ;;*****************************************************************
11796                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
11797                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
11798                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
11799                                  ;*GO TO THAT ROUTINE.
11800
11801  045524  010046          $TRAP:  MOV     R0,-(SP)                 ;;SAVE R0
11802  045526  016600  000002          MOV     2(SP),R0                 ;;GET TRAP ADDRESS
11803  045532  005740                  TST     -(R0)                    ;;BACKUP BY 2
11804  045534  111000                  MOVB    (R0),R0                  ;;GET RIGHT BYTE OF TRAP
11805  045536  006300                  ASL     R0                       ;;POSITION FOR INDEXING
11806  045540  016000  045560          MOV     $TRPAD(R0),R0            ;;INDEX TO TABLE
11807  045544  000200                  RTS     R0                       ;;GO TO ROUTINE
11808
11809
11810                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
11811
11812  045546  011646          $TRAP2: MOV     (SP),-(SP)               ;;MOVE THE PC DOWN
11813  045550  016666  000004  000002          MOV     4(SP),2(SP)      ;;MOVE THE PSW DOWN
11814  045556  000002                  RTI                              ;;RESTORE THE PSW
11815
11816                                  .SBTTL  TRAP TABLE
11817
11818                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
11819                                  ;*BY THE "TRAP" INSTRUCTION.
```

```
11820
11821                                   ;       ROUTINE
11822                                   ;       -------
11823   045560  045546                  $TRPAD: .WORD   $TRAP2
11824   045562  042720                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
11825   045564  043452                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
11826   045566  043426                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
11827   045570  043466                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
11828   045572  043202                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
11829
11830   045574  044242                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
11831
11832   045576  044152                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
11833   045600  044514                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
11834   045602  044604                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
11835   045604  045112                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
11836   045606  045252                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
11837   045610  045310                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
11838   045612  034122                          SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERPOR
11839   045614  035722                          SSINIT  ;;CALL=TSSINIT  TRAP+16(104416) INITIALIZE SUBSYSTEM
11840   045616  034634                          LOADRK  ;;CALL=TLOADRK  TRAP+17(104417) LOAD RK611 FOR OPERATION
11841   045620  034716                          GETRK   ;;CALL=TGETRK   TRAP+20(104420) GET RK611 REGISTERS
11842   045622  036706                          CHKOP   ;;CALL=TCHKOP   TRAP+21(104421) CHECK OPREATION FOR ANY ERRORS
11843   045624  036654                          CHKWE   ;;CALL=TCHKWE   TRAP+22(104422) CHECK OPERATION FOR EXPECTED ERRORS
11844   045626  034430                          IWAT16  ;;CALL=TWAT16   TRAP+23(104423) WAIT 16 MS
11845   045630  034420                          IWAT32  ;;CALL=TWAT32   TRAP+24(104424) WAIT 32 MS
11846   045632  034410                          IWAT48  ;;CALL=TWAT48   TRAP+25(104425) WAIT 48 MS
11847   045634  034400                          IWAT64  ;;CALL=TWAT64   TRAP+26(104426) WAIT 64 MS
11848   045636  034370                          IWAT80  ;;CALL=TWAT80   TRAP+27(104427) WAIT 80 MS
11849   045640  034360                          IWAT96  ;;CALL=TWAT96   TRAP+30(104430) WAIT 96 MS
11850   045642  034350                          IWAT112 ;;CALL=TWAT112  TRAP+31(104431) WAIT 112 MS
11851   045644  034340                          IWAT128 ;;CALL=TWAT128  TRAP+32(104432) WAIT 128 MS
11852   045646  034330                          IWAT144 ;;CALL=TWAT144  TRAP+33(104433) WAIT 144 MS
11853   045650  034320                          IWAT159 ;;CALL=TWAT159  TRAP+34(104434) WAIT 160 MS
11854   045652  034310                          IWAT1S  ;;CALL=TWAT1S   TRAP+35(104435) WAIT FOR 1 SECOND
11855   045654  034300                          IWAT2S  ;;CALL=TWAT2S   TRAP+36(104436) WAIT FOR 2 SECONDS
11856   045656  034260                          IWAT8S  ;;CALL=TWAT8S   TRAP+37(104437) WAIT FOR 8 SECONDS
11857   045660  034270                          IWAT1M  ;;CALL=TWAT1M   TRAP+40(104440) WAIT FOR 1 MIN
11858           000102          $TERM=.-$TRPAD
```

```
 11859
 11860
 11861
 11862
 11863
 11864
 11865  045662
 11866  045662  177777
 11867  045664  177777
 11868  045666  177777
 11869  045670  052525
 11870  045672  052525
 11871  045674  052525
 11872  045676  177777
 11873  045700  177777
 11874  045702  052525
 11875  045704  052525
 11876  045706  177777
 11877  045710  052525
 11878  045712  177252
 11879  045714  177252
 11880  045716  172765
 11881  045720  172765
 11882
 11883
 11884
 11885  045722
 11886  045722  000000
 11887  045724  000000
 11888  045726  000000
 11889  045730  177777
 11890  045732  177777
 11891  045734  177777
 11892  045736  000000
 11893  045740  000000
 11894  045742  177777
 11895  045744  177777
 11896  045746  000000
 11897  045750  177777
 11898  045752  000000
 11899  045754  177777
 11900  045756  000000
 11901  045760  177777
 11902
 11903
 11904
 11905  045762
 11906  045762  052525
 11907  045764  052525
 11908  045766  052525
 11909  045770  125252
 11910  045772  125252
 11911  045774  125252
 11912  045776  052525
 11913  046000  052525
 11914  046002  125252
```

```
.SBTTL  DATA PATTERNS
;DATA PATTERN 1
;       PATTERN IS ALL ZEROS

;DATA PATTERN 2
;       HI-LO FREQ. MIX
PAT02:
        177777
        177777
        177777
        052525
        052525
        052525
        177777
        177777
        052525
        052525
        177777
        052525
        177252
        177252
        172765
        172765

;DATA PATERN 3
;       HI FREQ. PHASE MIX
PAT03:
        000000
        000000
        000000
        177777
        177777
        177777
        000000
        000000
        177777
        177777
        000000
        177777
        000000
        177777
        000000
        177777

;DATA PATERN 4
;       LO FREQ. PHASE MIX
PAT04:
        052525
        052525
        052525
        125252
        125252
        125252
        052525
        052525
        125252
```

```
 11915  046004  125252                          125252
 11916  046006  052525                          052525
 11917  046010  125252                          125252
 11918  046012  052525                          052525
 11919  046014  125252                          125252
 11920  046016  052525                          052525
 11921  046020  125252                          125252
 11922
 11923                                  ;DATA PATTERN 5
 11924                                  ;       MAX PRECOMP. PHASE MIX
 11925  046022                          PAT05:
 11926  046022  133333                          133333
 11927  046024  066666                          066666
 11928  046026  155555                          155555
 11929  046030  155555                          155555
 11930  046032  133333                          133333
 11931  046034  066666                          066666
 11932  046036  066666                          066666
 11933  046040  155555                          155555
 11934  046042  155555                          155555
 11935  046044  133333                          133333
 11936  046046  133333                          133333
 11937  046050  133333                          133333
 11938  046052  133333                          133333
 11939  046054  133333                          133333
 11940  046056  133333                          133333
 11941  046060  133333                          133333
 11942
 11943                                  ;DATA PATTERN 6
 11944                                  ;       ROTATING BOUNDARY PULSE PRECOMP.
 11945  046062                          PAT06:
 11946  046062  121105                          121105
 11947  046064  150442                          150442
 11948  046066  064221                          064221
 11949  046070  132110                          132110
 11950  046072  055044                          055044
 11951  046074  026422                          026422
 11952  046076  013211                          013211
 11953  046100  105504                          105504
 11954  046102  042642                          042642
 11955  046104  021321                          021321
 11956  046106  110550                          110550
 11957  046110  044264                          044264
 11958  046112  022132                          022132
 11959  046114  011055                          011055
 11960  046116  104426                          104426
 11961  046120  042213                          042213
 11962
 11963                                  ;DATA PATTERN 7
 11964                                  ;       FIELD OF ALL ONES
 11965
 11966                                  ;DATA PATTERN 10
 11967                                  ;       ROTATING CELL PULSE PRECOMP.
 11968  046122                          PAT10:
 11969  046122  026455                          026455
 11970  046124  113226                          113226
```

```
11971  046126  045513                          045513
11972  046130  122645                          122645
11973  046132  151322                          151322
11974  046134  064551                          064551
11975  046136  132264                          132264
11976  046140  055132                          055132
11977  046142  026455                          026455
11978  046144  113226                          113226
11979  046146  045513                          045513
11980  046150  122645                          122645
11981  046152  151322                          151322
11982  046154  064551                          064551
11983  046156  132264                          132264
11984  046160  055132                          055132
11985
11986                                  ;DATA PATTERN 11
11987                                  ;       SHIFTED 1 IN A FIELD OF ZEROS
11988  046162                          PAT11:
11989  046162  000001                          000001
11990  046164  000002                          000002
11991  046166  000004                          000004
11992  046170  000010                          000010
11993  046172  000020                          000020
11994  046174  000040                          000040
11995  046176  000100                          000100
11996  046200  000200                          000200
11997  046202  000400                          000400
11998  046204  001000                          001000
11999  046206  002000                          002000
12000  046210  004000                          004000
12001  046212  010000                          010000
12002  046214  020000                          020000
12003  046216  040000                          040000
12004  046220  100000                          100000
12005
12006                                  ;DATA PATTERN 12
12007                                  ;       SHIFTED 0 IN A FIELD OF ONES
12008  046222                          PAT12:
12009  046222  177776                          177776
12010  046224  177775                          177775
12011  046226  177773                          177773
12012  046230  177767                          177767
12013  046232  177757                          177757
12014  046234  177737                          177737
12015  046236  177677                          177677
12016  046240  177577                          177577
12017  046242  177377                          177377
12018  046244  176777                          176777
12019  046246  175777                          175777
12020  046250  173777                          173777
12021  046252  167777                          167777
12022  046254  157777                          157777
12023  046256  137777                          137777
12024  046260  077777                          077777
12025
12026                                  ;DATA PATTERN 13
```

DATA PATTERNS

```
12027
12028  046262
12029  046262  052525
12030  046264  052525
12031  046266  052525
12032  046270  052525
12033  046272  052525
12034  046274  052525
12035  046276  052525
12036  046300  052525
12037  046302  052525
12038  046304  052525
12039  046306  052525
12040  046310  052525
12041  046312  052525
12042  046314  052525
12043  046316  052525
12044  046320  052525
12045
12046
12047  046322
12048  046322  125252
12049  046324  125252
12050  046326  125252
12051  046330  125252
12052  046332  125252
12053  046334  125252
12054  046336  125252
12055  046340  125252
12056  046342  125252
12057  046344  125252
12058  046346  125252
12059  046350  125252
12060  046352  125252
12061  046354  125252
12062  046356  125252
12063  046360  125252
12064
12065
12066
12067  046362
12068  046362  000001
12069  046364  000003
12070  046366  000007
12071  046370  000017
12072  046372  000037
12073  046374  000077
12074  046376  000177
12075  046400  000377
12076  046402  000777
12077  046404  001777
12078  046406  003777
12079  046410  007777
12080  046412  017777
12081  046414  037777
```

```
;       ALTERNATING 0-1
PAT13:
        052525
        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
        125252

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

```
046416  077777                  077777
046420  177777                  177777

                        ;DATA PATTERN 16
                        ;       COMPOSITE ROTATING
046422                  PAT16:
046422  072307                  072307
046424  135143                  135143
046426  156461                  156461
046430  167230                  167230
046432  073514                  073514
046434  035646                  035646
046436  016723                  016723
046440  107351                  107351
046442  143564                  143564
046444  061672                  061672
046446  030735                  030735
046450  114356                  114356
046452  046167                  046167
046454  123073                  123073
046456  151453                  151453
046460  164616                  164616
```

```
                        .SBTTL  FIELDS AND VARIABLES FOR OPERATION CHECKING
                024000  CS1ERBIT=24000                  ;CS1 ERROR BITS SPAR & DTO
                177400  CS2ERBIT=177400                 ;CS2 ERROR BITS
                                                        ;DLT,WCE,UPE,NED,NEM
                                                        ;PGE,MOS,UFE
                000070  DSERBIT=70                      ;DS ERROR BITS
                                                        ;SPDLSS,DROT,ACLO

046462  000000  EXPWC:  .WORD   0                      ;EXPECTED WORD COUNT (GIVEN)
046464  000000  EXPUBA: .WORD   0                      ;EXPECTED UPPER BA (COMPUTED)
046466  000000  EXPBA:  .WORD   0                      ;EXPECTED BUS ADDRESS (COMPUTED)
046470  000000  EXPCYL: .WORD   0                      ;EXPECTED CYLINDER (COMPUTED)
046472  000000  EXPSEC: .WORD   0                      ;EXPECTED SECTOR (COMPUTED)
046474  000000  EXPTRK: .WORD   0                      ;EXPECTED TRACK (COMPUTED)

046476  000000  REALWC: .WORD   0                      ;WORD COUNT AT END OF OPERATION
046500  000000  REALUB: .WORD   0                      ;REAL UPPER BA
046502  000000  REALBA: .WORD   0                      ;BUS ADDRESS
046504  000000  REALCY: .WORD   0                      ;CYLINDER
046506  000000  REALTRK:.WORD   0                      ;TRACK
046510  000000  REALSEC:.WORD   0                      ;SECTOR

046512  000000  GRP1ER: .WORD   0                      ;GROUP 1 ERROR FIELDS
                000001          SPARERR=BIT0          ;CONTROLLER DETECTED DRIVE BUS PARITY ERR
                000002          SKIERR= BIT1          ;SEEK INCOMPLETE
                000004          NXFERR= BIT2          ;NON-EXECUTABLE DRIVE FUNCTION
                000010          DRPARERR=BIT3         ;DRIVE DETECTED DRIVE BUS PARITY ERROR
                000020          FMTERR= BIT4          ;FORMAT ERROR
                000040          DTYERR= BIT5          ;DRIVE TYPE ERROR
                000100          ACLOERR=BIT6          ;AC LOW ERROR
                000200          SPDERR= BIT7          ;SPEED LOSS ERROR
                000400          DROTERR=BIT8          ;DRIVE OFF TRACK ERROR
                001000          COERR=  BIT9          ;CYLINDER OVER FLOW ERROR
                002000          IDAERR= BIT10         ;ILLEGAL DISK ADDRESS ERROR
                004000          WLERR=  BIT11         ;WRITE LOCK ERROR
                010000          DTERR=  BIT12         ;DRIVE TIMING ERROR
                020000          NCERWE= BIT13         ;NO CERR WITH ERROR SET ERROR
                040000          UNSERR= BIT14         ;DRIVE UNSAFE ERROR
                100000          CERNER= BIT15         ;CERR BUT NO ERROR SET ERROR

046514  000000  GRP2ER: .WORD   0                      ;GROUP 2 ERROR FIELD
                000001          ECHERR= BIT0          ;ECC HARD ERROR
                000002          DCKERR= BIT1          ;DATA CHECK ERROR
                000004          WCKERR= BIT2          ;WRITE CHECK ERROR
                000010          DLTERR= BIT3          ;DATA LATE ERROR
                000020          OPIERR= BIT4          ;OPERATION INCOMPLETE ERROR
                000040          HVRCERR=BIT5          ;HEADER VRC ERROR
                000100          BSERR=  BIT6          ;BAD SECTOR ERROR

046516  000000  GRP3ER: .WORD   0                      ;GROUP 3 ERROR FIELD
                000001          NEDERR= BIT0          ;NON-EXISTANT DRIVE ERROR
                000002          CTOERR= BIT1          ;CONTROLLER TIME OUT ERROR
                000004          UFERR=  BIT2          ;UNIT FIELD ERROR
                000010          MDSERR= BIT3          ;MULTIPLE DRIVE SELECT ERROR
                000020          PGERR=  BIT4          ;PROGRAMMING ERROR
                000040          NEMERR= BIT5          ;NON-EXISTANT MEMORY ERROR
```

```
 12162                  000100                  UPERR=  BIT6            ;UNIBUS PARITY ERROR
 12163                  000200                  ILFERR= BIT7            ;ILLEGAL FUNCTION ERROR.
 12164
 12165  046520  000000  GRP4ER: .WORD   0               ;GROUP 4 ERROR FIELD
 12166                  000001          WCERR=  BIT0            ;WORD COUNT ERROR FLAG
 12167                  000002          UBAERR= BIT1            ;UPPER BA ERROR
 12168                  000004          BAERR=  BIT2            ;BUS ADDRESS ERROR FLAG
 12169                  000010          CYLERR= BIT3            ;CYL ADDRESS ERROR FLAG
 12170                  000020          TRKERR= BIT4            ;TRACK ADDRESS ERROR FLAG
 12171                  000040          SECERR= BIT5            ;SECTOR ADDRESS ERROR FLAG
 12172
 12173  046522  052707  GRP4MS: .WORD   EM10
 12174  046524  052762          .WORD   EM11A
 12175  046526  052734          .WORD   EM11
 12176  046530  053036          .WORD   EM12
 12177  046532  053071          .WORD   EM13
 12178  046534  053121          .WORD   EM14
 12179
 12180  046536  053153  GRP3MS: .WORD   EM15
 12181  046540  053176          .WORD   EM16
 12182  046542  053221          .WORD   EM17
 12183  046544  053242          .WORD   EM18
 12184  046546  053267          .WORD   EM19
 12185  046550  053311          .WORD   EM20
 12186  046552  053335          .WORD   EM21
 12187  046554  053361          .WORD   EM22
 12188
 12189  046556  053402  GRP2MS: .WORD   EM23
 12190  046560  053413          .WORD   EM24
 12191  046562  053426          .WORD   EM25
 12192  046564  053442          .WORD   EM26
 12193  046566  053454          .WORD   EM27
 12194  046570  053501          .WORD   EM28
 12195  046572  053514          .WORD   EM29
 12196
 12197  046574  053535  GRP1MS: .WORD   EM30
 12198  046576  053610          .WORD   EM31
 12199  046600  053630          .WORD   EM32
 12200  046602  053666          .WORD   EM33
 12201  046604  053734          .WORD   EM34
 12202  046606  053751          .WORD   EM35
 12203  046610  053772          .WORD   EM36
 12204  046612  054001          .WORD   EM37
 12205  046614  054024          .WORD   EM38
 12206  046616  054044          .WORD   EM39
 12207  046620  054066          .WORD   EM40
 12208  046622  054120          .WORD   EM41
 12209  046624  054141          .WORD   EM42
 12210  046626  054164          .WORD   EM43
 12211  046630  054226          .WORD   EM44
 12212  046632  054243          .WORD   EM45
 12213
 12214  046634  000000  GPSUMF: .WORD   0               ;GROUP ERROR SUMMARY FLAGS
 12215                  000001          GRP1ST= BIT0            ;GROUP 1 ERROR SET
 12216                  000002          GRP2ST= BIT1            ;GROUP 2 ERROR SET
 12217                  000004          GRP3ST= BIT2            ;GROUP 3 ERROR SET
```

```
                    000010          GP1NR=  BIT3            ;GROUP 1 ERROR NOT RECEIVED
                    000020          GP2NR=  BIT4            ;GROUP 2 ERROR NOT RECEIVED
                    000040          GP3NR=  BIT5            ;GROUP 3 ERROR NOT RECEIVED
                    040000          DRSTER= BIT14           ;ERROR IN GETTING DRIVE STATUS FLAG.
                    100000          REPNR=  BIT15           ;REPORTING NOT RECEIVED SWITCH

                                    .SBTTL  TABLE OF OPERATION MESSAGE ADDRESS
                                    ;*      THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE
                                    ;*      USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.

046636  046576          CMNDLB: .WORD   OPER00          ;ADDRESS OF SELECT MESSAGE
046640  046713                  .WORD   OPER02          ;               PACK ACK
046642  046724                  .WORD   OPER04          ;               DRIVE CLEAR
046644  046740                  .WORD   OPER06          ;               UNLOAD
046646  046747                  .WORD   OPER10          ;               START SPINDLE
046650  046765                  .WORD   OPER12          ;               RECALIBRATE
046652  047001                  .WORD   OPER14          ;               OFFSET
046654  047010                  .WORD   OPER16          ;               SEEK
046656  047015                  .WORD   OPER20          ;               READ DATA
046660  047027                  .WORD   OPER22          ;               WRITE DATA
046662  047042                  .WORD   OPER24          ;               READ HEADER
046664  047057                  .WORD   OPER26          ;               WRITE HEADER
046666  047075                  .WORD   OPER30          ;               WRITE CHECK
046670  047111                  .WORD   OPER32          ;ILLEGAL OPERATION 33
046672  047135                  .WORD   OPER34          ;                  35
046674  047161                  .WORD   OPER36          ;                  37

                                    .SBTTL  OPERATION MESSAGES
046676  051104  053111  020105  OPER00: .ASCIZ  /DRIVE SELECT/
046704  042523  042514  052103
046712      000
046713      120 041501  020113  OPER02: .ASCIZ  /PACK ACK/
046720  041501  000113
046724  051104  053111  020105  OPER04: .ASCIZ  /DRIVE CLEAR/
046732  046103  040505  000122
046740  047125  047514  042101  OPER06: .ASCIZ  /UNLOAD/
046746      000
046747      123 040524  052122  OPER10: .ASCIZ  /START SPINDLE/
046754  051440  044520  042116
046762  042514      000
046765      122 041505  046101  OPER12: .ASCIZ  /RECALIBRATE/
046772  041111  040522  042524
047000      000
047001      117 043106  042523  OPER14: .ASCIZ  /OFFSET/
047006  000124
047010  042523  045505      000 OPER16: .ASCIZ  /SEEK/
047015      122 040505  020104  OPER20: .ASCIZ  /READ DATA/
047022  040504  040524      000
047027      127 044522  042524  OPER22: .ASCIZ  /WRITE DATA/
047034  042040  052101  000101
047042  042522  042101  044040  OPER24: .ASCIZ  /READ HEADERS/
047050  040505  042504  051522
047056      000
047057      127 044522  042524  OPER26: .ASCIZ  /WRITE HEADERS/
047064  044040  040505  042504
047072  051522      000
```

```
 12274  047075      127  044522  042524  OPER30: .ASCIZ  /WRITE CHECK/
 12275  047102  041440  042510  045503
 12276  047110      000
 12277  047111      111  046114  043505  OPER32: .ASCIZ  /ILLEGAL FUNCTION 33/
 12278  047116  046101  043040  047125
 12279  047124  052103  047511  020116
 12280  047132  031463      000
 12281  047135      111  046114  043505  OPER34: .ASCIZ  /ILLEGAL FUNCTION 35/
 12282  047142  046101  043040  047125
 12283  047150  052103  047511  020116
 12284  047156  032463      000
 12285  047161      111  046114  043505  OPER36: .ASCIZ  /ILLEGAL FUNCTION 37/
 12286  047166  046101  043040  047125
 12287  047174  052103  047511  020115
 12288  047202  033463      000
 12289  047205      127  044522  042524  OPER37: .ASCIZ   WRITE DATA ABORTED WITH BSE/
 12290  047212  042040  052101  020101
 12291  047220  041101  051117  042524
 12292  047226  020104  044527  044124
 12293  047234  041040  042523      000
 12294  047241                          OPER40:
 12295  047241      127  044522  042524  OPER41: .ASCIZ  /WRITE CHECK ABORTED WITH WCE/
 12296  047246  041440  042510  045503
 12297  047254  040440  047502  052122
 12298  047262  042105  053440  052111
 12299  047270  020110  041527  000105
 12300  047276  051127  052111  020105  OPER42: .ASCIZ  /WRITE DATA ABORTED WITH NON-EXISTANT MEMORY/
 12301  047304  040504  040524  040440
 12302  047312  047502  052122  042105
 12303  047320  053440  052111  020110
 12304  047326  047516  026516  054105
 12305  047334  051511  040524  052116
 12306  047342  046440  046505  051117
 12307  047350  000131
 12308  047352  042522  042101  042040  OPER43: .ASCIZ  /READ DATA ABORTED WITH NON-EXISTANT MEMORY/
 12309  047360  052101  020101  041101
 12310  047366  051117  042524  020104
 12311  047374  044527  044124  047040
 12312  047402  047117  042455  044530
 12313  047410  052123  047101  020124
 12314  047416  042515  047515  054522
 12315  047424      000
 12316  047425      127  044522  042524  OPER44: .ASCIZ  /WRITE DATA ABORTED WITH UNIBUS PARITY ERROR
 12317  047432  042040  052101  020101
 12318  047440  041101  051117  042524
 12319  047446  020104  044527  044124
 12320  047454  052440  044516  052502
 12321  047462  020123  040520  044522
 12322  047470  054524  042440  051122
 12323  047476  051117      000
 12324
```

```
12325                                           .SBTTL  ASCII MESSAGES
12326
12327  047501  040     000040                   SPACE2: .ASCIZ  /  /
12328  047504  005015  045522  030466           OPR001: .ASCIZ  <15><12>/RK611 BUS ADDRESS ( /
12329  047512  020061  052502  020123
12330  047520  042101  051104  051505
12331  047526  020123  020050     000
12332  047533     040  020051  020075           OPR002: .ASCIZ  / ) = /
12333  047540     000
12334  047541     122  033113  030461           OPR003: .ASCIZ  /RK611 VECTOR ADDRESS ( /
12335  047546  053040  041505  047524
12336  047554  020122  042101  051104
12337  047562  051505  020123  020050
12338  047570     000
12339  047571     122  033113  030461           OPR004: .ASCIZ  /RK611 PRIORITY ( /
12340  047576  050040  044522  051117
12341  047604  052111  020131  020050
12342  047612     000
12343  047613     111  051516  043125           OPR005: .ASCIZ  /INSUFFICIENT MEMORY. PROGRAM ABORTING./<15><12>
12344  047620  044506  044503  047105
12345  047626  020124  042515  047515
12346  047634  054522  020056  051120
12347  047642  043517  040522  020115
12348  047650  041101  051117  044524
12349  047656  043516  006456  000012
12350  047664  005015  047524  041040           OPR006: .ASCII  <15><12>/TO BYPASS TESTING DRIVE 0, PLACE IT OFF-LINE/
12351  047672  050131  051501  020123
12352  047700  042524  052123  047111
12353  047706  020107  051104  053111
12354  047714  020105  026060  050040
12355  047722  040514  042503  044440
12356  047730  020124  043117  026506
12357  047736  044514  042516
12358  047742  005015  047524  052040                   .ASCIZ  <15><12>/TO TEST DRIVE 0, REPLACE PROGRAM PACK WITH SCRATCH PACK/
12359  047750  051505  020124  051104
12360  047756  053111  020105  026060
12361  047764  051040  050105  040514
12362  047772  042503  050040  047522
12363  050000  051107  046501  050040
12364  050006  041501  020113  044527
12365  050014  044124  051440  051103
12366  050022  052101  044103  050040
12367  050030  041501  000113
12368  050034  005015  051104  053111           OPR007: .ASCIZ  <15><12>/DRIVE 0 WILL NOT BE TESTED. START AT 214 TO TEST DR 1/
12369  050042  020105  020060  044527
12370  050050  046114  047040  052117
12371  050056  041040  020105  042524
12372  050064  052123  042105  020056
12373  050072  052123  051101  020124
12374  050100  052101  031040  032061
12375  050106  052040  020117  042524
12376  050114  052123  042040  020122
12377  050122  000061
12378  050124  005015  047516  042040           OPR008: .ASCIZ  <15><12>/NO DRIVES AVAILABLE FOR TESTING. PROGRAM ABORTED/
12379  050132  044522  042526  020123
12380  050140  053101  044501  040514
```

```
12381  050146  046102  020105  047506
12382  050154  020122  042524  052123
12383  050162  047111  027107  050040
12384  050170  047522  051107  046501
12385  050176  040440  047502  052122
12386  050204  042105      000
12387  050207      015  052012  042510  OPR009: .ASCIZ  <15><12>/THE FOLLOWING DRIVES WILL BE TESTED/<15><12>
12388  050214  043040  046117  047514
12389  050222  044527  043516  042040
12390  050230  044522  042526  020123
12391  050236  044527  046114  041040
12392  050244  020105  042524  052123
12393  050252  042105  005015      000
12394  050257      015  047012  020117  OPR010: .ASCII  <15><12>/NO PARITY OPTION-UNIBUS PARITY ERROR AND/<15><12>
12395  050264  040520  044522  054524
12396  050272  047440  052120  047511
12397  050300  026516  047125  041111
12398  050306  051525  050040  051101
12399  050314  052111  020131  051105
12400  050322  047522  020122  047101
12401  050330  006504      012
12402  050333      062  024064  024470          .ASCIZ  /24(8) SECTOR FORMAT DATA XFER TEST BYPASSED/<15><12>
12403  050340  051440  041505  047524
12404  050346  020122  047506  046522
12405  050354  052101  042040  052101
12406  050362  020101  043130  051105
12407  050370  052040  051505  020124
12408  050376  054502  040520  051523
12409  050404  042105  005015      000
12410  050411      015  046412  046505  OPR011: .ASCII  <15><12>/MEMORY SIZE NOT LARGE ENOUGH FOR BUS ADDRESS BITS 16 & 17 TESTS
12411  050416  051117  020131  044523
12412  050424  042532  047040  052117
12413  050432  046040  051101  042507
12414  050440  042440  047516  043525
12415  050446  020110  047506  020122
12416  050454  052502  020123  042101
12417  050462  051104  051505  020123
12418  050470  044502  051524  030440
12419  050476  020066  020046  033461
12420  050504  052040  051505  051524
12421  050512  005015
12422  050514  046101  020114  040504          .ASCIZ  /ALL DATA XFER TESTS WITH ADDR >/
12423  050522  040524  054040  042506
12424  050530  020122  042524  052123
12425  050536  020123  044527  044124
12426  050544  040440  042104  020122
12427  050552  000076
12428  050554  031063  000113          OPR012: .ASCIZ  /32K/
12429  050560  032066  000113          OPR013: .ASCIZ  /64K/
12430  050564  033071  000113          OPR014: .ASCIZ  /96K/
12431  050570  041040  050131  051501  OPR015: .ASCIZ  / BYPASSED/<15><12>
12432  050576  042523  006504  000012
12433  050604  005015  020012  020040  OPR016: .ASCII  <15><12><12>/       *** CAUTION ***/<15><12><12>
12434  050612  020040  020040  025052
12435  050620  020052  040503  052125
12436  050626  047511  020116  025052
```

```
12437  050634  006452  005012
12438  050640  044124  051511  050040          .ASCII  /THIS PROGRAM SHOULD BE HALTED BY TYPING ↑C./<15><12>
12439  050646  047522  051107  046501
12440  050654  051440  047510  046125
12441  050662  020104  042502  044040
12442  050670  046101  042524  020104
12443  050676  054502  052040  050131
12444  050704  047111  020107  041536
12445  050712  006456     012
12446  050715     111  020106  040510          .ASCII  /IF HALTED USING THE HALT KEY, THE STATE OF THE DRIVE/<15><12>
12447  050722  052114  042105  052440
12448  050730  044523  043516  052040
12449  050736  042510  044040  046101
12450  050744  020124  042513  026131
12451  050752  052040  042510  051440
12452  050760  040524  042524  047440
12453  050766  020106  044124  020105
12454  050774  051104  053111  006505
12455  051002     012
12456  051003     117  020122  040503          .ASCII  /OR CARTRIDGE CANNOT BE PREDICTED./<15><12><12>
12457  051010  052122  044522  043504
12458  051016  020105  040503  047116
12459  051024  052117  041040  020105
12460  051032  051120  042105  041511
12461  051040  042524  027104  005015
12462  051046     012
12463  051047     101  046114  042040          .ASCII  /ALL DRIVES TO BE TESTED MUST BE ON-LINE,/<15><12>
12464  051054  044522  042526  020123
12465  051062  047524  041040  020105
12466  051070  042524  052123  042105
12467  051076  046440  051525  020124
12468  051104  042502  047440  026516
12469  051112  044514  042516  006454
12470  051120     012
12471  051121     122  040505  054504          .ASCII  /READY, AND WRITE LOCK RESET./<15><12>
12472  051126  020054  047101  020104
12473  051134  051127  052111  020105
12474  051142  047514  045503  051040
12475  051150  051505  052105  006456
12476  051156     012
12477  051157     101  054516  042040          .ASCII  /ANY DRIVE NOT TO BE TESTED MUST BE OFF-LINE./<15><12><12>
12478  051164  044522  042526  047040
12479  051172  052117  052040  020117
12480  051200  042502  052040  051505
12481  051206  042524  020104  052515
12482  051214  052123  041040  020105
12483  051222  043117  026506  044514
12484  051230  042516  006456  005012
12485  051236  047516  042524  020072          .ASCII  /NOTE: 2ND AND SUBSEQUENT PASS RUN TIME IS/<15><12>
12486  051244  047062  020104  047101
12487  051252  020104  052523  051502
12488  051260  050505  042525  052116
12489  051266  050040  051501  020123
12490  051274  052522  020116  044524
12491  051302  042515  044440  006523
12492  051310     012
```

```
 12493  051311     040  020040  020040          .ASCIZ  /      APPROX 7 MINUTES FOR EACH DRIVE./<15><12>
 12494  051316  040440  050120  047522
 12495  051324  020130  020067  044515
 12496  051332  052516  042524  020123
 12497  051340  047506  020122  040505
 12498  051346  044103  042040  044522
 12499  051354  042526  006456  000012
 12500  051362  005015  044506  051522  OPR017: .ASCII  <15><12>/FIRST 256 SECTORS NOT BSE ERROR FREE./
 12501  051370  020124  032462  020066
 12502  051376  042523  052103  051117
 12503  051404  020123  047516  020124
 12504  051412  051502  020105  051105
 12505  051420  047522  020122  051106
 12506  051426  042505     056
 12507  051431     115  054101  046511          .ASCIZ  /MAXIMUM DATA TRANSFER TEST BYPASSED/<15><12>
 12508  051436  046525  042040  052101
 12509  051444  020101  051124  047101
 12510  051452  043123  051105  052040
 12511  051460  051505  020124  054502
 12512  051466  040520  051523  042105
 12513  051474  005015     000
 12514  051477     040  020040  020040  OPR018: .ASCIZ  /     /<15><12>/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/<15><12>
 12515  051504  005015  047117  054514
 12516  051512  030440  042040  044522
 12517  051520  042526  020056  053117
 12518  051526  051105  040514  050120
 12519  051534  042105  047440  042520
 12520  051542  040522  044524  047117
 12521  051550  020123  054502  040520
 12522  051556  051523  042105  005015
 12523  051564     000
 12524  051565     015  051412  047503  OPR019: .ASCII  <15><12>@SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)@
 12525  051572  042520  020072  044103
 12526  051600  020061  052050  044522
 12527  051606  024507  020054  032505
 12528  051614  026463  035470  041440
 12529  051622  031110  020054  032105
 12530  051630  026471  020062  040450
 12531  051636  020103  047503  050125
 12532  051644  042514  020054  031056
 12533  051652  027526  046503     051
 12534  051657     015  040412  045104          .ASCIZ  <15><12>/ADJUST R72 FOR CONSTANT LEVEL ON CH2/<15><12>
 12535  051664  051525  020124  033522
 12536  051672  020062  047506  020122
 12537  051700  047503  051516  040524
 12538  051706  052116  046040  053105
 12539  051714  046105  047440  020116
 12540  051722  044103  006462  000012
 12541  051730  005015  051120  043517  OPR020: .ASCIZ  <15><12>/PROGRAM HALT PENDING - CARTRIDGE FORMAT BEING CORRECTED/<15><12
 12542  051736  040522  020115  040510
 12543  051744  052114  050040  047105
 12544  051752  044504  043516  026440
 12545  051760  041440  051101  051124
 12546  051766  042111  042507  043040
 12547  051774  051117  040515  020124
 12548  052002  042502  047111  020107
```

```
12549  052010  047503  051122  041505
12550  052016  042524  006504  000012
12551  052024  025052  025052  020052  OPR021: .ASCIZ  /*****  PROGRAM HALTED  *****/<15><12>
12552  052032  050040  047522  051107
12553  052040  046501  044040  046101
12554  052046  042524  020104  025040
12555  052054  025052  025052  005015
12556  052062     000
12557  052063     103  051101  051124  OPR022: .ASCIZ  /CARTRIDGE FORMAT CORRECTION FAILED/<15><12>
12558  052070  042111  042507  043040
12559  052076  051117  040515  020124
12560  052104  047503  051122  041505
12561  052112  044524  047117  043040
12562  052120  044501  042514  006504
12563  052126  000012
12564  052130  005015  051120  043517  ABORT:  .ASCIZ  <15><12>/PROGRAM ABORTING BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
12565  052136  040522  020115  041101
12566  052144  051117  044524  043516
12567  052152  041040  041505  052501
12568  052160  042523  042440  051122
12569  052166  051117  052040  051110
12570  052174  051505  047510  042114
12571  052202  042440  041530  042505
12572  052210  042504  006504  000012
12573
12574                          .SBTTL ERROR MESSAGES
12575  052216  040506  040524  026514  EM1:    .ASCIZ  /FATAL-NON EXISTANT MEMORY AT RK611 BASE ADDRESS/
12576  052224  047516  020116  054105
12577  052232  051511  040524  052116
12578  052240  046440  046505  051117
12579  052246  020131  052101  051040
12580  052254  033113  030461  041040
12581  052262  051501  020105  042101
12582  052270  051104  051505  000123
12583  052276  040506  040524  026514  EM2:    .ASCIZ  /FATAL-WRITE READY AND IE DID NOT CAUSE INTERRUPT/
12584  052304  051127  052111  020105
12585  052312  042522  042101  020131
12586  052320  047101  020104  042511
12587  052326  042040  042111  047040
12588  052334  052117  041440  052501
12589  052342  042523  044440  052116
12590  052350  051105  052522  052120
12591  052356     000
12592  052357     106  052101  046101  EM3:    .ASCIZ  /FATAL-PARITY ERROR TRAP. PC AT ERROR = /
12593  052364  050055  051101  052111
12594  052372  020131  051105  047522
12595  052400  020122  051124  050101
12596  052406  020056  041520  040440
12597  052414  020124  051105  047522
12598  052422  020122  020075     000
12599  052427     105  050130  041505  EM4:    .ASCIZ  /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
12600  052434  042524  020104  047111
12601  052442  042524  051122  050125
12602  052450  020124  044504  020104
12603  052456  047516  020124  041517
12604  052464  052503  020122  051117
```

```
 12605  052472  053440  051501  046040
 12606  052500  052101  027105  041440
 12607  052506  046517  040515  042116
 12608  052514  053440  051501  000072
 12609  052522  052523  051502  051531  EM5:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT RESET ERROR/
 12610  052530  042524  020115  046103
 12611  052536  040505  020122  044504
 12612  052544  020104  047516  020124
 12613  052552  042522  042523  020124
 12614  052560  051105  047522  000122
 12615  052566  052523  051502  051531  EM6:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
 12616  052574  042524  020115  046103
 12617  052602  040505  020122  044504
 12618  052610  020104  047516  020124
 12619  052616  042522  042523  020124
 12620  052624  042504  044526  04-503
 12621  052632  044440  052116  051105
 12622  052640  052522  052120     000
 12623  052645     123  041125  054523  EM7:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT SET READY/
 12624  052652  052123  046505  041440
 12625  052660  042514  051101  042040
 12626  052666  042111  047040  052117
 12627  052674  051440  052105  051040
 12628  052702  040505  054504     000
 12629  052707     127  051117  020104  EM10:   .ASCIZ  /WORD COUNT INCORRECT/
 12630  052714  047503  047125  020124
 12631  052722  047111  047503  051122
 12632  052730  041505  000124
 12633  052734  052502  020123  042'01  EM11:   .ASCIZ  /BUS ADDRESS INCORRECT/
 12634  052742  051104  051505  020123
 12635  052750  047111  047503  051122
 12636  052756  041505  000124
 12637  052762  050125  042520  020122  EM11A:  .ASCIZ  /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
 12638  052770  052502  020123  042101
 12639  052776  051104  051505  020123
 12640  053004  044502  051524  044440
 12641  053012  041516  051117  042522
 12642  053020  052103  024040  040502
 12643  053026  033061  020054  033461
 12644  053034  000051
 12645  053036  054503  044514  042116  EM12:   .ASCIZ  /CYLINDER ADDRESS INCORRECT/
 12646  053044  051105  040440  042104
 12647  053052  042522  051523  044440
 12648  053060  041516  051117  042522
 12649  053066  052103     000
 12650  053071     124  040522  045503  EM13:   .ASCIZ  /TRACK ADDRESS INCORRECT/
 12651  053076  040440  042104  042522
 12652  053104  051523  044440  041516
 12653  053112  051117  042522  052103
 12654  053120     000
 12655  053121     123  041505  047524  EM14:   .ASCIZ  /SECTOR ADDRESS INCORRECT./
 12656  053126  020122  042101  051104
 12657  053134  051505  020123  047111
 12658  053142  047503  051122  041505
 12659  053150  027124     000
 12660  053153     116  047117  042455  EM15:   .ASCIZ  /NON-EXISTANT DRIVE/
```

```
12661  053160  044530  052123  047101
12662  053166  020124  051104  053111
12663  053174  000105
12664  053176  047503  052116  047522  EM16:   .ASCIZ  /CONTROLLER TIMEOUT/
12665  053204  046114  051105  052040
12666  053212  046511  047505  052125
12667  053220     000
12668  053221     125  044516  020124  EM17:   .ASCIZ  /UNIT FIELD ERROR/
12669  053226  044506  046105  020104
12670  053234  051105  047522  000122
12671  053242  052515  044514  046120  EM18:   .ASCIZ  /MULIPLE DRIVE SELECT/
12672  053250  020105  051104  053111
12673  053256  020105  042523  042514
12674  053264  052103     000
12675  053267     120  047522  051107  EM19:   .ASCIZ  /PROGRAMMING ERROR/
12676  053274  046501  044515  043516
12677  053302  042440  051122  051117
12678  053310     000
12679  053311     116  047117  042455  EM20:   .ASCIZ  /NON-EXISTANT MEMORY/
12680  053316  044530  052123  047101
12681  053324  020124  042515  047515
12682  053332  054522     000
12683  053335     125  044516  052502  EM21:   .ASCIZ  /UNIBUS PARITY ERROR/
12684  053342  020123  040520  044522
12685  053350  054524  042440  051122
12686  053356  051117     000
12687  053361     111  046114  043505  EM22:   .ASCIZ  /ILLEGAL FUNCTION/
12688  053366  046101  043040  047125
12689  053374  052103  047511  000116
12690  053402  041505  020103  040510  EM23:   .ASCIZ  /ECC HARD/
12691  053410  042122     000
12692  053413     104  052101  020101  EM24:   .ASCIZ  /DATA CHECK/
12693  053420  044103  041505  000113
12694  053426  051127  052111  020105  EM25:   .ASCIZ  /WRITE CHECK/
12695  053434  044103  041505  000113
12696  053442  040504  040524  046040  EM26:   .ASCIZ  /DATA LATE/
12697  053450  052101  000105
12698  053454  050117  051105  052101  EM27:   .ASCIZ  /OPERATION INCOMPLETE/
12699  053462  047511  020116  047111
12700  053470  047503  050115  042514
12701  053476  042524     000
12702  053501     110  040505  042504  EM28:   .ASCIZ  /HEADER VRC/
12703  053506  020122  051126  000103
12704  053514  040502  020104  042523  EM29:   .ASCIZ  /BAD SECTOR ERROR/
12705  053522  052103  051117  042440
12706  053530  051122  051117     000
12707  053535     103  047117  051124  EM30:   .ASCIZ  /CONTROLLER DETECTED DRIVE BUS PARITY ERROR/
12708  053542  046117  042514  020122
12709  053550  042504  042524  052103
12710  053556  042105  042040  044522
12711  053564  042526  041040  051525
12712  053572  050040  051101  052111
12713  053600  020131  051105  047522
12714  053606  000122
12715  053610  042523  045505  044440  EM31:   .ASCIZ  /SEEK INCOMPLETE/
12716  053616  041516  046517  046120
```

```
12717  053624  052105  000105
12718  053630  047516  026516  054105  EM32:   .ASCIZ  /NON-EXECUTABLE DRIVE FUNCTION/
12719  053636  041505  052125  041101
12720  053644  042514  042040  044522
12721  053652  042526  043040  047125
12722  053660  052103  047511  000116
12723  053666  051104  053111  020105  EM33:   .ASCIZ   DRIVE DETECTED DRIVE BUS PARITY ERROR/
12724  053674  042504  042524  052103
12725  053702  042105  042040  044522
12726  053710  042526  041040  051525
12727  053716  050040  051101  052111
12728  053724  020131  051105  047522
12729  053732  000122
12730  053734  047516  046522  052101  EM34:   .ASCIZ  /FORMAT ERROR/
12731  053742  042440  051122  051117
12732  053750     000
12733  053751     104  044522  042526  EM35:   .ASCIZ  /DRIVE TYPE ERROR/
12734  053756  052040  050131  020105
12735  053764  051105  047522  000122
12736  053772  041501  046040  053517  EM36:   .ASCIZ  /AC LOW/
12737  054000     000
12738  054001     123  044520  042116  EM37:   .ASCIZ  /SPINDLE SPEED LOSS/
12739  054006  042514  051440  042520
12740  054014  042105  046040  051517
12741  054022  000123
12742  054024  051104  053111  020105  EM38:   .ASCIZ  /DRIVE OFF TRACK/
12743  054032  043117  020106  051124
12744  054040  041501  000113
12745  054044  054503  044514  042116  EM39:   .ASCIZ  /CYLINDER OVERFLOW/
12746  054052  051105  047440  042526
12747  054060  043122  047514  000127
12748  054066  046111  042514  040507  EM40:   .ASCIZ  /ILLEGAL DISK PACK ADDRESS/
12749  054074  020114  044504  045523
12750  054102  050040  041501  020113
12751  054110  042101  051104  051505
12752  054116  000123
12753  054120  051127  052111  020105  EM41:   .ASCIZ  /WRITE LOCK ERROR/
12754  054126  047514  045503  042440
12755  054134  051122  051117     000
12756  054141     104  044522  042526  EM42:   .ASCIZ  /DRIVE TIMING ERROR/
12757  054146  052040  046511  047111
12758  054154  020107  051105  047522
12759  054162  000122
12760  054164  047516  041440  051105  EM43:   .ASCIZ  /NO CERR WITH SOME OTHER ERROR SET/
12761  054172  020122  044527  044124
12762  054200  051440  046517  020105
12763  054206  052117  042510  020122
12764  054214  051105  047522  020122
12765  054222  042523  000124
12766  054226  051104  053111  020105  EM44:   .ASCIZ  /DRIVE UNSAFE/
12767  054234  047125  040523  042506
12768  054242     000
12769  054243     103  051105  020122  EM45:   .ASCIZ  /CERR SET BUT NO OTHER ERROR SET/
12770  054250  042523  020124  052502
12771  054256  020124  047516  047440
12772  054264  044124  051105  042440
```

```
 12773  054272  051122  051117  051440
 12774  054300  052105     000
 12775
 12776  054303     126  020126  044504  EM46:   .ASCIZ  /VV DID NOT SET WITH PACK ACK
 12777  054310  020104  047516  020124
 12778  054316  042523  020124  044527
 12779  054324  044124  050040  041501
 12780  054332  020113  041501  000113
 12781  054340  052123  052101  051525  EM47:   .ASCIZ  /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
 12782  054346  053040  046101  042111
 12783  054354  051440  052105  047440
 12784  054362  020116  042523  042514
 12785  054370  052103  052040  020117
 12786  054376  047516  026516  054105
 12787  054404  051511  040524  052116
 12788  054412  042040  044522  042526
 12789  054420     000
 12790  054421     123  040524  052524  EM48:   .ASCIZ  /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
 12791  054426  020123  040526  044514
 12792  054434  020104  042522  042523
 12793  054442  020124  047117  051440
 12794  054450  046105  041505  020124
 12795  054456  047524  042440  044530
 12796  054464  052123  047101  020124
 12797  054472  051104  053111  000105
 12798  054500  052123  052101  051525  EM49:   .ASCIZ  /STATUS VALID NOT RESET ON DRIVE RELEASE/
 12799  054506  053040  046101  042111
 12800  054514  047040  052117  051040
 12801  054522  051505  052105  047440
 12802  054530  020116  051104  053111
 12803  054536  020105  042522  042514
 12804  054544  051501  000105
 12805  054550  054105  042520  052103  EME3:   .ASCIZ  /EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
 12806  054556  042105  031040  042116
 12807  054564  044440  052116  051105
 12808  054572  052522  052123  042040
 12809  054600  042111  047040  052117
 12810  054606  047440  041503  051125
 12811  054614  047440  020122  040527
 12812  054622  020123  040514  042524
 12813  054630  020056  047503  046515
 12814  054636  047101  020104  040527
 12815  054644  035123     000
 12816  054647     103  047101  047516  EM51:   .ASCII   /CANNOT READ BAD SECTOR FILE/<15><12>
 12817  054654  020124  042522  042101
 12818  054662  041040  042101  051440
 12819  054670  041505  047524  020122
 12820  054676  044506  042514  005015
 12821  054704  042524  052123  047111          .ASCIZ  /TESTING ABORTED ON THIS DRIVE/
 12822  054712  020107  041101  051117
 12823  054720  042524  020104  047117
 12824  054726  052040  044510  020123
 12825  054734  051104  053111  000105
 12826  054742  046101  043511  046516  EME2:   .ASCIZ  /ALIGNMENT PACK ON THIS DRIVE. TESTING ABORTED/
 12827  054750  047105  020124  040520
 12828  054756  045503  047440  020116
```

```
 12829  055764  044124  051511  042040
 12830  055772  044522  042526  020056
 12831  056000  042524  052123  047111
 12832  056006  020107  041101  051117
 12833  056014  042524  000104
 12834  056020  047524  046440  047101  EM53:   .ASCIZ  /TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
 12835  056026  020131  040502  020104
 12836  056034  042523  052103  051117
 12837  056042  027123  052040  051505
 12838  056050  044524  043516  040440
 12839  056056  047502  052122  042105
 12840  056064  047440  020116  044124
 12841  056072  051511  042040  044522
 12842  056100  042526     000
 12843  056103     110  040505  042504  EM54:   .ASCIZ  /HEADER WORD MISCOMPARE/
 12844  056110  020122  047527  042122
 12845  056116  046440  051511  047503
 12846  056124  050115  051101  000105
 12847  056132  040504  040524  053440  EM55:   .ASCIZ  /DATA WORD MISCOMPARE/
 12848  056140  051117  020104  044515
 12849  056146  041523  046517  040520
 12850  056154  042522     000
 12851
 12852  056157     103  047101  047516  EM56:   .ASCIZ  /CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/
 12853  056164  020124  044506  042116
 12854  056172  044040  040505  042504
 12855  056200  020122  020060  047111
 12856  056206  051040  040505  020104
 12857  056214  047101  020104  047523
 12858  056222  052122  044040  040505
 12859  056230  042504  051522     000
 12860  056235     123  040524  052524  EMSVAL: .ASCIZ  /STATUS VALID/
 12861  056242  020123  040526  044514
 12862  056250  000104
 12863  056252  042523  042514  052103  EMNZPR: .ASCIZ  /SELECT WITH NON-ZERO STATUS PAIR/
 12864  056260  053440  052111  020110
 12865  056266  047516  026516  042532
 12866  056274  047522  051440  040524
 12867  056302  052524  020123  040520
 12868  056310  051111     000
 12869  056313     127  044522  044524  EMWCS2: .ASCIZ  /WRITING COMMAND STATUS REGISTER 2/
 12870  056320  043516  041440  046517
 12871  056326  040515  042116  051440
 12872  056334  040524  052524  020123
 12873  056342  042522  044507  052123
 12874  056350  051105  031040     000
 12875          053751                  EMDTPE= EM35                    ;DRIVE TYPE ERROR
 12876  056355     104  044522  042526  EMDI:   .ASCIZ  /DRIVE INTERRUPT/
 12877  056362  044440  052116  051105
 12878  056370  052522  052120     000
 12879          053666                  EMDPAR= EM33                            ;DRIVE DETECTED DRIVE BUS PARITY ERROR
 12880  056375     104  044522  042526  EMDSC:  .ASCIZ  /DRIVE STATUS CHANGE/
 12881  056402  051440  040524  052524
 12882  056410  020123  044103  047101
 12883  056416  042507     000
 12884  056421     104  044522  042526  EMDA:   .ASCIZ  /DRIVE ATTENTION/
```

```
12885  055426  040440  052124  047105
12886  055434  044524  047117     000
12887  055441     103  047117  051124  EMCCLR: .ASCIZ  /CONTROLLER CLEAR/
12888  055446  046117  042514  020122
12889  055454  046103  040505  000122
12890           046676                  EMSELD= OPER00          ;DRIVE SELECT
12891           046724                  EMDCLR= OPER04                  ;DRIVE CLEAR
12892           046765                  EMRCAL= OPER12                  ;RECALIBRATE
12893  055462  042523  047503  042116  EM2INT: .ASCIZ  /SECOND INTERRUPT/
12894  055470  044440  052116  051105
12895  055476  052522  052120     000
12896  055503     123  042505  020113  EMSKSF: .ASCIZ  /SEEK TO SELF/
12897  055510  047524  051440  046105
12898  055516  000106
12899           047010                  EMSK=   OPER16                  ;SEEK
12900  055520  047125  054105  042520  EMUXIT: .ASCIZ  /UNEXPECTED INTERRUPT/
12901  055526  052103  042105  044440
12902  055534  052116  051105  052522
12903  055542  052120     000
12904  055545     125  044516  052502  EMUR:   .ASCIZ  /UNIBUS RESET/
12905  055552  020123  042522  042523
12906  055560  000124
12907  055562  042522  042523  000124  EMRSET: .ASCIZ  /RESET/
12908           053442                  EMDLT=  EM26                    ;DATA LATE
12909  055570  042522  042101  047111  EMRDB:  .ASCIZ  /READING DATA BUFFER/
12910  055576  020107  040504  040524
12911  055604  041040  043125  042506
12912  055612  000122
12913  055614  047503  052116  047522  EMCERR: .ASCIZ  /CONTROLLER ERROR/
12914  055622  046114  051105  042440
12915  055630  051122  051117     000
12916  055635     123  041125  054523  EMSCLR: .ASCIZ  /SUBSYSTEM CLEAR/
12917  055642  052123  046505  041440
12918  055650  042514  051101     000
12919  055655     115  046125  044524  EMMI:   .ASCIZ  /MULTIPLE INTERRUPTS/
12920  055662  046120  020105  047111
12921  055670  042524  051122  050125
12922  055676  051524     000
12923  055701     102  042101  051440  DRVABT: .ASCII  'BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(8))'<15><12>
12924  055706  041505  047524  051522
12925  055714  047440  020116  040520
12926  055722  045503  044440  020116
12927  055730  051101  040505  020123
12928  055736  051525  042105  041040
12929  055744  020131  042524  052123
12930  055752  024040  054503  020114
12931  055760  030463  024062  024470
12932  055766  006451     012
12933  055771     124  051505  044524          .ASCIZ  'TESTING ABORTED ON THIS DRIVE'
12934  055776  043516  040440  047502
12935  056004  052122  042105  047440
12936  056012  020116  044124  051511
12937  056020  042040  044522  042526
12938  056026     000
12939
12940                                   .SBTTL  DATA HEADERS FOR ERROR REPORTS
```

```
12941  056027     124  052123  047040  DH001:  .ASCIZ  /TST NUM ERR PC  DRIVE/
12942  056034  046525  042440  051122
12943  056042  050040  020103  042040
12944  056050  044522  042526     000
12945  056055     124  052123  047040  DH002A: .ASCIZ  /TST NUM ERR PC  DRIVE/
12946  056062  046525  042440  051122
12947  056070  050040  020103  042040
12948  056076  044522  042526     000
12949  056103     122  041513  030523  DH002B: .ASCIZ  /RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA/
12950  056110  020040  051040  041513
12951  056116  031123  020040  051040
12952  056124  042113  020123  020040
12953  056132  051040  042513  020122
12954  056140  020040  051040  040513
12955  056146  047523  020106  051040
12956  056154  042113  054503  020114
12957  056162  051040  042113  000101
12958  056170  045522  040502  020040  DH002C: .ASCIZ  /RKBA    RKWC/
12959  056176  020040  045522  041527
12960  056204     000
12961  056205     101  030060  020040  DH002D: .ASCIZ  /A00     B00     A01     B01     A02     B02     A03     B03/
12962  056212  020040  041040  030060
12963  056220  020040  020040  040440
12964  056226  030460  020040  020040
12965  056234  041040  030460  020040
12966  056242  020040  040440  031060
12967  056250  020040  020040  041040
12968  056256  031060  020040  020040
12969  056264  040440  031460  020040
12970  056272  020040  041040  031460
12971  056300     000
12972  056301     122  041513  030523  DH003B: .ASCIZ  /RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1/
12973  056306  020040  051040  041513
12974  056314  031123  020040  051040
12975  056322  042113  020123  020040
12976  056330  051040  042513  020122
12977  056336  020040  051040  040513
12978  056344  047523  020106  051040
12979  056352  046513  030522     000
12980  056357     124  042510  040440  DH005:  .ASCIZ  /THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:/
12981  056364  047502  042526  040440
12982  056372  042522  042440  050130
12983  056400  041505  042524  020104
12984  056406  051105  047522  051522
12985  056414  052040  040510  020124
12986  056422  044504  020104  047516
12987  056430  020124  042523  020124
12988  056436  047111  047440  042520
12989  056444  040522  044524  047117
12990  056452  000072
12991  056454  044124  020105  041101  DH006:  .ASCIZ  /THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:/
12992  056462  053117  020105  051101
12993  056470  020105  047125  054105
12994  056476  042520  052103  042105
12995  056504  042440  051122  051117
12996  056512  020123  042523  020124
```

```
 12997  056520  047111  047440  042520
 12998  056526  040522  044524  047117
 12999  056534  000072
 13000  056536  044124  020105  041101  DH007:  .ASCIZ  /THE ABOVE ARE ERRORS SET IN OPERATION:/
 13001  056544  053117  020105  051101
 13002  056552  020105  051105  047522
 13003  056560  051522  051440  052105
 13004  056566  044440  020116  050117
 13005  056574  051105  052101  047511
 13006  056602  035116     000
 13007  056605     101  054516  043040  DH005A: .ASCIZ  /ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID/
 13008  056612  042511  042114  053440
 13009  056620  052111  020110  046101
 13010  056626  020114  047117  051505
 13011  056634  046440  051525  020124
 13012  056642  042502  041440  047117
 13013  056650  044523  042504  042522
 13014  056656  020104  047111  040526
 13015  056664  044514  000104
 13016  056670  051105  047522  020122  DH013:  .ASCIZ  /ERROR AT COMPLETION OF OPERATION/
 13017  056676  052101  041440  046517
 13018  056704  046120  052105  047511
 13019  056712  020116  043117  047440
 13020  056720  042520  040522  044524
 13021  056726  047117     000
 13022  056731     105  050130  042124  DH010A: .ASCIZ  /EXPTD   IS/
 13023  056736  020040  044440  000123
 13024  056744  042522  042101  042040  DH010B: .ASCIZ  /READ DATA WITH FORCED CYLINDER OVERFLOW/
 13025  056752  052101  020101  044527
 13026  056760  044124  043040  051117
 13027  056766  042503  020104  054503
 13028  056774  044514  042116  051105
 13029  057002  047440  042526  043122
 13030  057010  047514  000127
 13031  057014  047516  020124  042523  DH011:  .ASCIZ  /NOT SET AS A RESULT OF/
 13032  057022  020124  051501  040440
 13033  057030  051040  051505  046125
 13034  057036  020124  043117     000
 13035  057043     116  052117  051040  DH012:  .ASCIZ  /NOT RESET AS A RESULT OF/
 13036  057050  051505  052105  040440
 13037  057056  020123  020101  042522
 13038  057064  052523  052114  047440
 13039  057072  000106
 13040  057074  042523  020124  051501  DH013:  .ASCIZ  /SET AS A RESULT OF/
 13041  057102  040440  051040  051505
 13042  057110  046125  020124  043117
 13043  057116     000
 13044  057117     122  051505  052105  DH014:  .ASCIZ  /RESET AS A RESULT OF/
 13045  057124  040440  020123  020101
 13046  057132  042522  052523  052114
 13047  057140  047440  000106
 13048  057144  047507  042117  020040  DH015:  .ASCIZ  /GOOD    BAD     WORD NUM/
 13049  057152  020040  040502  020104
 13050  057160  020040  020040  047527
 13051  057166  042122  047040  046525
 13052  057174     000
```

```
13053  057175     123  041125  054523  DH016:  .ASCIZ  /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/
13054  057202  052123  046505  041440
13055  057210  042514  051101  052040
13056  057216  020117  042522  042523
13057  057224  020124  042523  045505
13058  057232  046040  046511  052111
13059  057240  040440  046114  053517
13060  057246  047111  020107  042510
13061  057254  042101  020123  047524
13062  057262  051040  046105  040517
13063  057270  000104
13064  057272  042523  042514  052103  DH017:  .ASCIZ  /SELECT AFTER RECAL/
13065  057300  040440  052106  051105
13066  057306  051040  041505  046101
13067  057314     000
13068  057315     122  041505  046101  DH018:  .ASCIZ  /RECAL COMPLETE ATTN AFTER SELECT/
13069  057322  041440  046517  046120
13070  057330  052105  020105  052101
13071  057336  047124  040440  052106
13072  057344  051105  051440  046105
13073  057352  041505  000124
13074
13075                                .SBTTL  DATA TABLES FOR ERROR REPORTS
13076                                        .EVEN
13077  057356  001302  001116  001626  DT003:  .WORD   $TESTN,$ERRPC,DRVNUM,T.CS1,T.CS2,T.DS,T.ER,T.ASOF,T.MR1
13078  057364  001540  001550  001552
13079  057372  001554  001556  001566
13080  057400  001302  001116  001626  DT001:  .WORD   $TESTN,$ERRPC,DRVNUM
13081  057406                          DT002:
13082  057406                          DT004:
13083  057406                          DT005:
13084  057406                          DT006:
13085  057406                          DT007:
13086  057406  001302  001116  001626  DT010:  .WORD   $TESTN,$ERRPC,DRVNUM
13087  057414  001540  001550  001552          .WORD   T.CS1,T.CS2,T.DS,T.ER,T.ASOF
13088  057422  001554  001556
13089  057426  001560  001546          .WORD   T.DCYL,T.DA
13090  057432  001544  001542          .WORD   T.BA,T.WC
13091  057436  001202  001204  001206  .WORD   $REG10,$REG11,$REG12,$REG13,$REG14,$REG15,$REG16,$REG17
13092  057444  001210  001212  001214
13093  057452  001216  001220
13094  057456  001302  001116  001626  DT015:  .WORD   $TESTN,$ERRPC,DRVNUM
13095  057464  001202  001204  001206  DT015A: .WORD   $REG10,$REG11,$REG12
13096                                .SBTTL  DATA FORMATS FOR ERROR REPORTS
13097  057472  000001                  DF001:  .WORD   1
13098  057474     003     000                  .BYTE   3,0
13099
13100  057476  000004                  DF002:  .WORD   4
13101  057500     000     000                  .BYTE   0,0
13102  057502  056055                       .WORD   DH002A
13103  057504     003     000                  .BYTE   3,0
13104  057506  056103                       .WORD   DH002B
13105  057510     007     000                  .BYTE   7,0
13106  057512  056170                       .WORD   DH002C
13107  057514     002     000                  .BYTE   2,0
13108
```

```
13109
13110  057516  000002              DF003:  .WORD   2
13111  057520     003      000             .BYTE   3,0
13112  057522  056301                      .WORD   DH003B
13113  057524     006      000             .BYTE   6,0
13114
13115  057526  000006              DF004:  .WORD   6
13116  057530     000      000             .BYTE   0,0
13117  057532  000000              DF004A: .WORD   0
13118  057534     000      000             .BYTE   0,0
13119  057536  056055                      .WORD   DH002A
13120  057540     003      000             .BYTE   3,0
13121  057542  056103                      .WORD   DH002B
13122  057544     007      000             .BYTE   7,0
13123  057546  056170                      .WORD   DH002C
13124  057550     002      000             .BYTE   2,0
13125  057552  056205                      .WORD   DH002D
13126  057554     010      000             .BYTE   10,0
13127
13128  057556  000007              DF005:  .WORD   7
13129  057560     000      000             .BYTE   0,0
13130  057562  000000              DF005A: .WORD   0
13131  057564     000      000             .BYTE   0,0
13132  057566  056055                      .WORD   DH002A
13133  057570     003      000             .BYTE   3,0
13134  057572  056103                      .WORD   DH002B
13135  057574     007      000             .BYTE   7,0
13136  057576  056170                      .WORD   DH002C
13137  057600     002      000             .BYTE   2,0
13138  057602  056205                      .WORD   DH002D
13139  057604     010      000             .BYTE   10,0
13140  057606  056605                      .WORD   DH005A
13141  057610     000      000             .BYTE   0,0
13142
13143  057612  000005              DF006:  .WORD   5
13144  057614     000      000             .BYTE   0,0
13145  057616  000000              DF006A: .WORD   0
13146  057620     000      000             .BYTE   0,0
13147  057622  056055                      .WORD   DH002A
13148  057624     003      000             .BYTE   3,0
13149  057626  056103                      .WORD   DH002B
13150  057630     007      000             .BYTE   7,0
13151  057632  056170                      .WORD   DH002C
13152  057634     002      000             .BYTE   2,0
13153
13154  057636  000004              DF007:  .WORD   4
13155  057640     000      000             .BYTE   0,0
13156  057642  000000              DF007A: .WORD   0
13157  057644     000      000             .BYTE   0,0
13158  057646  056055                      .WORD   DH002A
13159  057650     003      000             .BYTE   3,0
13160  057652  056301                      .WORD   DH003B
13161  057654     006      000             .BYTE   6,0
13162
13163  057656  000004              DF010:  .WORD   4
13164  057660     000      000             .BYTE   0,0
```

```
13165  057662  000000                DF010A: .WORD   0
13166  057664     000      000              .BYTE   0,0
13167  057666  056055                       .WORD   DH002A
13168  057670     003      000              .BYTE   3,0
13169  057672  056731                       .WORD   DH010A
13170  057674     002      000              .BYTE   2,0
13171
13172  057676  000004                DF011:  .WORD   4
13173  057700     000      000              .BYTE   0,0
13174  057702  000000                DF011A: .WORD   0
13175  057704     000      000              .BYTE   0,0
13176  057706  056055                       .WORD   DH002A
13177  057710     003      000              .BYTE   3,0
13178  057712  056301                       .WORD   DH003B
13179  057714  000006  000000               .WORD   6,0
13180
13181  057720  000002                DF015:  .WORD   2
13182  057722     003      000              .BYTE   3,0
13183  057724  057144                       .WORD   DHC15
13184  057726     003      000              .BYTE   3,0
13185
13186  057730  000001                DF016:  .WORD   1
13187  057732     003      000              .BYTE   3,0
13188
13189  057734  000052                BS26:   .BLKW   52
13190  060060  000052                BS24:   .BLKW   52
13191  060204  000200                PATCH:  .BLKW   200
13192  060604  001000                IBUFF:  .BLKW   1000
13193  062604  001000                OBUFF:  .BLKW   1000
13194           000001                .END
```

```
ABASE = 177440          1956#   2278    2319
ABORT   052130          9449   12564#
ABTFAI  042705          9244   11170#
ACDW1 = 000000          2278    2321
ACDW2 = 000000          2278
ACLO  = 000010          2051#  10266
ACLOER= 000100         10268   12135#
ACPUOP= 000000          2278    2293
ADDW0 = 000000          2278
ADDW1 = 000000          2278
ADDW10= 000000          2278
ADDW11= 000000          2278
ADDW12= 000000          2278
ADDW13= 000000          2278
ADDW14= 000000          2278
ADDW15= 000000          2278
ADDW2 = 000000          2278
ADDW3 = 000000          2278
ADDW4 = 000000          2278
ADDW5 = 000000          2278
ADDW6 = 000000          2278
ADDW7 = 000000          2278
ADDW8 = 000000          2278
ADDW9 = 000000          2278
ADEVCT= 000000          2278    2284
ADEVM = 000000          2278    2320
ADJCLK  042332          3185   11063#
AENV  = 000000          2278    2289
AENVM = 000000          2278    2290
AFATAL= 000000          2278    2281
AMADR1= 000000          2278    2306
AMADR2= 000000          2278    2310
AMADR3= 000000          2278    2313
AMADR4= 000000          2278    2316
AMAMS1= 000000          2278    2300
AMAMS2= 000000          2278    2308
AMAMS3= 000000          2278    2311
AMAMS4= 000000          2278    2314
AMSGAD= 000000          2278    2286
AMSGLG= 000000          2278    2287
AMSGTY= 000000          2278    2280
AMTYP1= 000000          2278    2301
AMTYP2= 000000          2278    2309
AMTYP3= 000000          2278    2312
AMTYP4= 000000          2278    2315
APASS = 000000          2278    2283
APRIOR= 000240          1955#   2278
APTCSU= 000040          9302#  11203
APTENV= 000001          9258    9300#   9357   11196
APTSIZ= 000200          2775    9299#
APTSPO= 000100          9260    9301#  11198
ASWREG= 000000          2278    2291
ATESTN= 000000          2278    2282
AUNIT = 000000          2278    2285
AUSWR = 000000          2278    2292
AVECT1= 000210          1954#   2278    2317
```

```
AVECT2= 000000          2278    2318
BAERR = 000004         10069   12168#
BAI   = 000020          2014#   4704    7376    7391    7476    7547    7603   10037
BA16  = 000400          1999#   4938    5024    5364    5417
BA17  = 001000          2000#   5084    5175    5364    5417
BDSRCK  040676         10638#  10807
BITCNT  034774          8638    9763#  10411   10421   10429   10441   10451   10462
BIT0  = 000001          1904#   1996    2029    2048    2686    3614    3851    4307    4495    4497    4499    5514    5517
                        6527    6530    6940    8748    8928    9921   12129   12147   12156   12166   12215
BIT00 = 000001          1894#   1904
BIT01 = 000002          1893#   1903
BIT02 = 000004          1892#   1902
BIT03 = 000010          1891#   1901
BIT04 = 000020          1890#   1900
BIT05 = 000040          1889#   1899
BIT06 = 000100          1888#   1898
BIT07 = 000200          1887#   1897
BIT08 = 000400          1886#   1896    9125
BIT09 = 001000          1885#   1895    9143    9368
BIT1  = 000002          1903#   2030    2687    3614    3853    4553    8759    8939   12130   12148   12157   12167   12216
BIT10 = 002000          1884#   2001    2020    2039    2072    2086    2099    2113    2127    9322   10931   12139
BIT11 = 004000          1883#   2002    2021    2040    2057    2073    2087    2100    2114    2128    9150   11013   12140
BIT12 = 010000          1882#   2004    2022    2041    2074    2088    2101    2115    2129   10850   10858   10876   10900
                       10904   10928   11038   12141
BIT13 = 020000          1881#   2005    2023    2042    2058    2075    2089    2102    2116    2130    9329    9479   10850
                       10858   10876   10900   10928   11038   12142
BIT14 = 040000          1880#   2006    2024    2043    2059    2076    2090    2103    2117    2131    4235    4236    6670
                        6671    9111   10780   10825   10841   10893   12143   12221
BIT15 = 100000          1879#   2007    2008    2025    2044    2060    2077    2136    2694    4232    4233    4385    7047
                        7182   10780   10802   10823   12144   12222
BIT2  = 000004          1902#   2031    2050    2688    5510    5511    6523    6524   12131   12149   12158   12168   12217
BIT3  = 000010          1901#   2013    2032    2051    2689   12132   12150   12159   12169   12218
BIT4  = 000020          1900#   2014    2033    2052    2066    2107    2121    3106   12133   12151   12160   12170   12219
BIT5  = 000040          1899#   2015    2034    2053    2067    2081    2094    2108    2122    3106   12134   12152   12161
                       12171   12220
BIT6  = 000100          1898#   1997    2016    2035    2054    2068    2082    2095    2109    2123    2691    9530    9864
                       12135   12153   12162
BIT7  = 000200          1897#   1998    2017    2036    2055    2069    2083    2096    2110    2124    2690    2692    2800
                        3002    3093    8470    9492   10757   12136   12163
BIT8  = 000400          1896#   1999    2018    2037    2056    2070    2084    2097    2111    2125    2693    3073    3142
                        3392    8477    8504    8573    8732    8742    8765    8773    8912    8922    8945    8953   10763
                       12137
BIT9  = 001000          1895#   2000    2019    2038    2071    2085    2098    2112    2126    4437    4438    4492    4493
                       10763   10778   10795   12138
BPTVEC= 000014          1911#
BSE   = 000200          2036#   7328    7486   10260   10307
BSERPT= 000200          2692#   7330    7332    7489    7491
BSERR = 000100          6702    6778   10309   12153#
BSF24P  001636          2672#   4047   10805
BSF26P  001640          2674#   3953   10801
BSS24P  001642          2676#   4053*  10829
BSS26P  001644          2678#   4038*  10826
BS24    060060          2672    4062   13190#
BS24CT  001650          2681#   4058*
BS26    057734          2674    4061   13189#
BS26CT  001646          2680#   4045*
```

```
CCLR   = 100000          2008#   3426    3480    7890    9910    9937
CDT    = 002000          2001#   3336
CERNER= 100000          10205   12144#
CERR   = 100000          2007#   3023    3062    4150    9925    9981    9985   10203   10278
CFMT   = 010000          2004#   6361    6389    6447    6462    6539    6590    6605    8272    9820   10120   10775   10803
                        10827   10834
CHKBA   036154          10034#  10228
CHKCTS  036366           6706   10096#  10229
CHKOP   036706          10186#  11842
CHKST   036714          10184   10189#
CHKWC   036122          10015#  10226
CHKWE   036654          10177#  11843
CKSWR  = 104407           9105    9318    9367   11832#
CLEAR  = 000105           1981#   3514    7523    7582    7656    7993    8097    8136    8257    8391    8826    8845
CLKCAL  034146           9563#   9658
CMNDLB  046636           9669   10232   10524   12228#
COE    = 001000          2038#
COERR  = 001000           6336   12138#
CR     = 000015           1819#  11242   11252
CRLF   = 000200           1820#   2799   11213   11252
CS1ERB= 024000          10195   12107#
CS2ERB= 177400          10197   12108#
CTO    = 004000           2002#   9927   10317
CTOERR= 000002           7697   10319   12157#
CTRHLT  042442           2915    2944    3091   11091#  11438   11521
CYLERR= 000010          10146   12169#
DCK    = 100000           2044#  10260   10288
DCKERR= 000002           7906   10290   12148#
DDISP  = 177570           1826#   2225    2763
DDT    = 000400           2056#
DESHLD  001666           2703#  10843   10867*  10919*  11043
DF001   057472           2372   13097#
DF002   057476           2387   13100#
DF003   057516           2398   13110#
DF004   057526           2479   13115#
DF004A  057532          10456   10467   13117#
DF005   057556           2493   13128#
DF005A  057562          10476   13130#
DF006   057612           2518   13143#
DF006A  057616          10434   10446   13145#
DF007   057636           2544   13154#
DF007A  057642          10408   13156#
DF010   057656           2567   13163#
DF010A  057662           5380*   5433*   5542*   6712*   6719*   7017*   7023*   7034*   7040*   7152*   7158*   7169*   7175*
                        10232*  13165#
DF011   057676           2581    2590    2599    2607   13172#
DF011A  057702           3171*   3254*   3270*   3305*   3316*   3352*   3403*   3408*   3414*   3420*   3436*   3442*   3459*
                         3465*   3471*   3477*   3499*   3505*   3531*   3537*   3665*   3680*   3687*   3704*   3733*   3745*
                         3774*   3794*   3801*   4146*   4153*   8550*   8616*   8686*   8703*   8883*   9888*  13174#
DF015   057720           2619   13181#
DF016   057730           2627   13186#
DH001   056027           2370   12941#
DH002A  056055           2396    2617   12945#  13102   13119   13132   13147   13158   13167   13176
DH002B  056103          12949#  13104   13121   13134   13149
DH002C  056170          12958#  13106   13123   13136   13151
DH002D  056205          12961#  13125   13138
```

```
DH0038  056301          12972*  13112   13160   13178
DH005   056357          10517   12980*
DH005A  056605          13007*  13140
DH006   056454          10520   10577   12991*
DH007   056536          10515   13000*
DH010   056670           2565   13016*
DH010A  056731          13022*  13169
DH010B  056744          13024*
DH011   057014           2579   13031*
DH012   057043           2588   13035*
DH013   057074           2597   13040*
DH014   057117           2605   13044*
DH015   057144          13048*  13183
DH016   057175           7795   13053*
DH017   057272           8695   13064*
DH018   057315           8703   13068*
DH2N    001372           2385*   7795*   8695*   9669*   9846*   9874*
DH4N    001412           2477*  10457   10468
DH5N    001422           2491*  10477
DH6N    001432           2516*  10435   10447
DH7N    001442           2542*  10409
DI    = 040000           2006*   3405    3439    3462    3502    3534    3730    3771    3798    8833    8852    9981
DISPLA  001142           2225*   2763*   2771*   9165*   9321*
DISPRE  000174           2166*   2771
DLT   = 100000           2025*   4143   10295
DLTERR= 000010          10297   12150*
DMD   = 000040           2067*   3567    3575    3614    3623    7967    8049    8057    8157    8305   11074
DRA   = 000001           2048*
DRDY  = 000200           2055*   3036    3041    3064    8443
DROT  = 000040           2053*  10274
DROTER= 000400           8210    8344   10276   12137*
DRPAR = 000010           2032*
DRPARE= 000010           3387   12132*
DRSTER= 040000          10472   10478   10612   12221*
DRVABT  055701           4071   12923*
DRVBIT  001630           2668*   3115*   3175   3177*   3982   4031   8470   8474*   8475   8477   8483*
DRVMSK= 000007           2012*
DRVNUM  001626           2667*   3121*   3181*   3201   3241   3291   3308   3334   3378   3394   3556   3604   3654
                         3720    3762    3826   3843   3887   3904   6362   6390   6816   7322   8441   8473*  8484*
                         8661*   8673*   8694*  8814*  8835*  8855*  8864*  8879*  9702   9828  11066  13077  13080
                        13086   13094
DSC   = 040000           2059*   3168    3411   3468   3684
DSERBI= 000070          10199   12111*
DSWR  = 177570           1825*   2224    2762
DTE   = 010000           2041*
DTERR = 010000           7978   12141*
DTYE  = 000040           2034*
DTYERR= 000040          12134*
DT001   057400           2371   13080*
DT002   057406           2386   13081*
DT003   057356           2397   13077*
DT004   057406           2478   13082*
DT005   057406           2492   13083*
DT006   057406           2517   13084*
DT007   057406           2543   13085*
DT010   057406           2580    2589    2598    2606   13086*
```

```
DTC15   057456          2566    2618   13094#
DTC15A  057464          2626   13095#
DC'ST = 000001          2686#   3011    3032    3043    3048
ECCM  = 020000          2075#
ECH   = 000100          2035#  10260   10284
ECHERR= 000001          1906   10286   12147#
EMCCLR  055441          3436    3442   12887#
EMCERR  055614          4152   12913#
EMCM    055421          3402    3435    3458    3498    3530    8549    8615    8892   12884#
EMDCLR= 046724          3531    3537   12891#
EMDI    055355          3407    3441    3464    3504    3536    3732    3773    3800   12876#
EMDLY = 053442          4145    9887   12908#
EMDPAR= 053666          3403    3408    3414    3420    3459    3465    3471    3477   12879#
EMDSC   055316          3170    3413    3470    3686   12880#
EMDTPE= 053751          3352   12875#
EMHI    055655          3744   12919#
EMMJPR  055252          3270    3499    3505   12863#
EMRCAL= 046765          3665   12892#
EMROB   055570          4146    4153    9888   12909#
EMRSET  055562          3794    3801   12907#
EMSCLR  055635          3171    8616   12916#
EMSELD= 046676          3254    3305    3704   12890#
EMSK  = 047010          8683   12899#
EMSKSF  055503          3733    3745    3774   12896#
EMSVAL  055235          3253    3269    3304    3315    3351    3419    3476    3664    3679    3703    8685    8702   12860#
EMTVEC= 000030          1914#   2746*   2747*
EMUR    055545          8550   12904#
EMUXIT  055520          3793   12900#
EMWCS2  055313          3316   12869#
EM1     052216          2913    2942   12575#
EM1N    001360          2369#   2913*   2942*   3980*   3990*   4000*   4007*   4016*   4029*   4071*   7333*   7492*
EM1C    052707          5397    5451    5557    7022    7039    7157    7174   12173   12629#
EM1DN   001450          2564#   5379*   5390*   5397*   5432*   5444*   5451*   5543*   5557*   6709*   6718*   7014*   7022*
                        7031*   7039*   7149*   7157*   7166*   7174*  10239#
EM11    052734          5390    5444    5543    7014    7031    7149    7166   12175   12633#
EM11A   052762          5379    5432   12174   12637#
EM11N   001460          2578#   3253*   3304*   3351*   3402*   3407*   3413*   3419*   3458*   3464*   3470*   3476*   3664*
                        3686*   3703*   3732*   3773*   8685*   8882*
EM12    053036         12176   12645#
EM12N   001470          2587#   3170*   3269*   3315*   3530*   3536*   3679*   3800*   8549*   8615*
EM13    053071          4145*   6709   12177   12650#
EM13N   001500          2596#   3744*   3793*   4152*   9887*
EM14    053121          6718   12178   12655#
EM14N   001510          2604#   3435*   3441*   3504*   8702*
EM15    053153         12180   12660#
EM15N   001520          2616#   10756*  10759*
EM16    053176         12181   12664#
EM17    053221         12182   12668#
EM18    053242         12183   12671#
EM19    053267         12184   12675#
EM2     052276         12583#
EM2INT  055462          3680    3687   12893#
EM2N    001370          2384#   3673*   9670*   9845*   9873*
EM2C    053311         12185   12679#
EM21    053335         12186   12683#
EM22    053361         12187   12687#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM23    053402          12189   12690#
EM24    053413          12190   12692#
EM25    053426          12191   12694#
EM26    053442          12192   12696#  12908
EM27    053454          12193   12698#
EM28    053501          12194   12702#
EM29    053514          12195   12704#
EM3     052357           9481   12592#
EM3N    001400           2395#   3138*   3150*   3164*   3221*   9987*   9989*  10000*
EM30    053535          12197   12707#
EM31    053610          12198   12715#
EM32    053630          12199   12718#
EM33    053666          12200   12723#  12879
EM34    053734          12201   12730#
EM35    053751          12202   12733#  12875
EM36    053772          12203   12736#
EM37    054001          12204   12738#
EM38    054024          12205   12742#
EM39    054044          12206   12745#
EM4     052427           9670    9845   12599#
EM4N    001410           2476#   3498*
EM40    054066          12207   12748#
EM41    054120          12208   12753#
EM42    054141          12209   12756#
EM43    054164          12210   12760#
EM44    054226          12211   12766#
EM45    054243          12212   12769#
EM46    054303           3164   12776#
EM47    054340           3138   12781#
EM48    054421           3150   12790#
EM49    054500           3221   12798#
EM5     052522           9987   12609#
EM5N    001420           2490#
EM50    054550           3673   12805#
EM51    054647           3980    3990    4000    4007   12816#
EM52    054742           4016   12826#
EM53    055020           4029   12834#
EM54    055103          10756   12843#
EM55    055132          10759   12847#
EM56    055157           9873   12852#
EM6     052566           9989   12615#
EM6N    001430           2515#
EM7     052645          10000   12623#
EM7N    001440           2541#
ERRCNT  001632           2669#   9444*   9447
ERRLMT  001634           2671#   4175    4671    4744    4825    4894    4975    5044    5124    5195    5269    5340    5646
                         5734    5817    5902    5987    6046    6103    6161    6217    6273    6407    6482    6625    6750
                         6788    7419
ERRVEC= 000004           1907#   2760    2761*   2772*   9033    9034    9038*   9043*   9055*   9060*   9069*   9079*   9080*
                         9116    9117*   9119*   9122*
EXPBA   046466          10039*  10066*  12116#
EXPCYL  046470          10125*  10139*  10144   12117#
EXPSEC  046472           6720   10143*  12118#
EXPTRK  046474           6710   10142*  12119#
EXPUBA  046464          10065*  10073   12115#
EXPWC   046462          10015*  10017   10240   12114#
```

D04


```
FMTE   = 000020          2033#
FMTERR= 000020          12133#
FPFMT  = 002400          2693#   7262
GENCOM  045776           4132    4170    4178    4228    4303    4382    4434    4488    4549    4610    4639    4665    4674
                         4705    4739    4747    4790    4803    4819    4828    4865    4888    4897    4932    4951    4966
                         4978    5013    5037    5047    5090    5117    5127    5164    5188    5198    5227    5247    5263
                         5272    5301    5319    5334    5343    5499    5595    5606    5640    5649    5679    5702    5728
                         5737    5765    5781    5811    5820    5850    5862    5896    5905    5935    5947    5981    5990
                         6017    6021    6040    6049    6075    6079    6097    6106    6132    6136    6155    6164    6188
                         6192    6211    6220    6244    6248    6267    6276    6368    6402    6410    6438    6442    6476
                         6485    6512    6581    6585    6619    6628    6667    6680    6684    6744    6753    6760    6782
                         6791    6823    6895    6944    7077    7203    7276    7361    7413    7422    7852    7871    8405
                         8449   10750#  11140
GETCHS  035E06           9909#  10473
GETRK   034716           9739#  11841
GNS    = ******          2165    2798    8987    8994   11824   11825   11826   11827   11828   11830   11832   11833   11834
                        11835   11836   11837   11838   11839   11840   11841   11842   11843   11844   11845   11846   11847
                        11848   11849   11850   11851   11852   11853   11854   11855   11856   11857
GO     = 000001          1996#
GPSUMF  046634          10194*  10407   10578*  12214#
GP1NR  = 000010          10465   10470   10518   10572   10609   12218#
GP2NR  = 000020          10444   10518   10572   10605   12219#
GP3NR  = 000040          10424   10518   10572   12220#
GRP1ER  046512          10190*  10205*  10252   10358   10450   10560   12128#
GRP1MS  046574          10561   10592   12197#
GRP1ST= 000001          10454   10470   10526   10574   10609   12215#
GRP2ER  046514          10191*  10282   10428   10554   12146#
GRP2MS  046556          10555   10586   12189#
GRP2ST= 000002          10432   10526   10574   10605   12216#
GRP3ER  046516          10192*  10311   10410   10531   12155#
GRP3MS  046536          10532   10581   12180#
GRP3ST= 000004          10415   10526   10574   12217#
GRP4ER  046520           6705*   6707    6716   10019*  10069*  10075*  10146*  10149*  10152*  10193*  10230   10233   12165#
GRP4MS  046522          10239   12173#
GTSWR  = 104406           2793   11830#
HT     = 000011          1817#   11211   11252
HVRC   = 000400          2037#   10260   10303
HVRCER= 000040          10305   12152#
IBUFF   060604           3961    4014    4020    4121    4653    4719    4810    4958    5105    5242    5315    5613    5628
                         5697    5716    5788    5802    5869    5884    5954    5969    6028    6085    6143    6199    6255
                         6299    6326    6464    6607    6732    6767    6849    7399    7817    7895    9823    9877   10838
                        10899   13192#
IDAE   = 002000          2039#
IDAERR= 002000          3857    3865    3913   12139#
IE     = 000100          1997#   2932    3787    7702    7788    7986    8383
ILF    = 000001          2029#  10260   10341
ILFERR= 000200          10343   12163#
INTHLR  033626           2879    2997    9472#
INTR   = 000300          1992#
INTSET  001662           2697#   2906*   2911    2931*   2936    3660*   3742    3778*   3833*   3894*   7321*   7356*   7511*
                         7571*   7649*   7701*   7738*   7790*   7885    7944*   7987*   8091*   8127*   8189*   8249*   8376*
                         8515*   8583*   8670*   8689*   8821*   8842*   9463*   9472*   9506*   9511    9518*   9523    9526*
                         9534    9539*   9568    9591    9659    9714*   9833*   9837   11115*
IOTVEC= 000020          1912#   2744*   2745*
IP     = 000100          2016#
IWAIT1  034270          9615#  11857
```

```
IWAT1S  034310          9621#  11854
IWAT11  034350          9633#  11850
IWAT12  034340          9630#  11851
IWAT14  034330          9627#  11852
IWAT1S  034320          9624#  11853
IWAT16  034430          9651#  11844
IWAT2S  034300          9618#  11855
IWAT3Z  034420          9648#  11845
IWAT48  034410          9645#  11846
IWAT64  034400          9642#  11847
IWAT8S  034260          9612#  11856
IWAT8Q  034370          9639#  11848
IWAT96  034360          9636#  11849
KIPAR0= 172340          1945#   9019    9047
KIPAR1= 172342          1946#
KIPAR2= 172344          1947#
KIPAR3= 172346          1948#
KIPAR4= 172350          1949#
KIPARS= 172352          1950#
KIPAR6= 172354          1951#  10903*
KIPAR7= 172356          1952#   9063
KIPDR0= 172300          1934#
KIPDR1= 172302          1935#
KIPDR2= 172304          1936#
KIPDR3= 172306          1937#
KIPDR4= 172310          1938#
KIPDRS= 172312          1939#
KIPDR6= 172314          1940#
KIPDR7= 172316          1941#
KWLADD  001676          2707#   9492*   9530*   9818*   9864*  11072*
KWLVEC  001700          2708#   9527
LCKHLR  033670          9491#   9528
LCLKPR= 100000          2694#   9537    9564    9661    9816    9862  11070
LC_KTK  001660          2696#   3782*   3784    9491*   9563*   9570   9572*   9597*   9664   9834*   9840
LF    = 000012          1818#  11246   11252
LOADRK  034634          9714#  11840
LRLOAD  034574          3958    4094    4122    4219    4247    4295   4328    4374    4394   4426    4447   4480   4508
                        4541    4567    4602    4631    4650    4696   4716    4783    4807   4858    4875   4937   4955
                        5006    5023    5083    5102    5157    5174   5219    5238    5294   5312    5363   5416   5486
                        5587    5610    5625    5671    5694    5713   5757    5785    5799   5842    5866   5881   5927
                        5951    5966    6025    6082    6140    6196   6252    6296    6323   6446    6461   6539   6589
                        6604    6659    6688    6729    6764    6846   6887    6906    6948   6963    6983   7081   7096
                        7118    7207    7224    7268    7302    7345   7368    7396    7469   7500    7522   7537   7561
                        7581    7596    7634    7655    7670    7682   7726    7745    7760   7814    7856   7875   7893
                        7933    7948    7960    7992    8024    8042   8080    9096    8111   8135    8150   8196   8234
                        8256    8271    8286    8298    8329    8352   8390    8409    8452   9694#  11100  11129
L.ASOF  001612          2656#   8118*
L.BA    001604          2651#   6365*   6819*   8868*  10039   10054
L.CS1   001600          2649#   3016*   3055*   3122*   3153*   3202*  3242*   3292*   3335*  3336*   3379*  3426*  3445*
                        3480*   3486*   3514*   3557*   3605*   3655*  3691*   3721*   3763*  3825*   3844*  3886*  3905*
                        6361*   6389*   6817*   8509*   8578*   8662*  8674*   8816*   8826*  8837*   8845*  8859*  8869*
                        9697    9729    9820    9835    9973   10043  10120   10775   10803  10827   10834
L.CS2   001610          2655#   3014*   3054*   3120*   3201*   3211*  3241*   3291*   3334*  3378*   3556*  3566*  3604*
                        3615*   3654*   3720*   3762*   3826*   3843*  3887*   3904*   4004*  6362*   6390*  6816*  7376*
                        7391*   7476*   7547*   7603*   8050*   8508*  8577*   9660*   9672*  9815*   9836*  9856*  9865*
                        9919   10037
```

```
L.DA    001606          2652#   9830
L.DCYL  001614          2657#   3906*   6366*   6392*   6821*   8857*   8866*   9829   10125   10644   10767   11139*
L.DS    001606          2653#   6854*  10128
L.DT    001607          2654#   3845*   4186    6364*   6391*   6818*  10126   10647   10766   11150   11152*
L.MR1   001616          2658#   3243*   3257*   3380*   3424*   3488*   3512*   3567*   3614*   7752*   7967*   8049*   8157*
                        8203*   8305*   8337*   8360*   8388*
L.WC    001602          2650#   6363*   6820*   8867*   9717   10041   10097
MCLK  = 000400          2070#   9794    9795
MCLOCK  035050          3571    3619    7970    8054    8161    8308    8313    8318    9785#
MDS   = 001000          2019#  10325
MDSERR= 000010         10327   12159#
MEMPYB= 000004          2688#   5479    5482
MEMSZB= 000002          2687#   4770    4775    4918    4924    5069    5075
MERO  = 001000          2071#
MESMSK= 000017          2064#
MEWC  = 002000          2072#
MILCNT  001652          2682#   9590
MINC  = 000200          2069#   8311    8316
MMCSR1  001702          2709#   5494*   5511*   5514*   5527*   6524*   6527*   6556*   9507*
MMCSR2  001704          2710#   5497*   5510*   5517*   5530*   6523*   6530*   6559*   9519*
MMVEC = 000250          1923#
MMVECA  001706          2711#   9515
MSP   = 000100          2068#
MYTIME  034214          3783    9589#   9663    9839
M.CADD= 017760          2139#
M.CDIF= 017760          2138#
M.DRV = 000007          2135#
M.HEAD= 007000          2142#
M.ID  = 000003          2137#
M.PAR = 100000          2136#
M.SECT= 000760          2141#
M.SER = 077770          2140#
NCERWE= 020000         10280   12142#
NED   = 010000          2022#   3025    3077   10313
NEDERR= 000001         10315   12156#
NEM   = 004000          2021#  10333
NEMERR= 000040          5377    5430   10335   12161#
NEWDRV  027236          3983    4032    8468#
NEXINT  033620          2903    9463#   9505
NXF   = 000004          2031#
NXFERR= 000004         12131#
OBJFF   062604          4097    4222    4231    4250    4298    4306    4331    4377    4385*   4397    4429    4437*   4438*
                        4450    4483    4491    4511    4544    4552    4570    4605    4634    4699    4725    4726    4861
                        5009    5160    5222    5258    5297    5329    5489    5512*   5513*   5531*   5532*   5544    5546
                        5551    5553    5590    5674    5760    5776    5845    5930    6365    6449    6525*   6526*   6541
                        6560*   6561*   6592    6662    6670*   6671*   6691    6819    6890    6909    6951    6966    6980*
                        6986    7011    7015    7029    7032    7084    7099    7114*   7121    7146    7150    7164    7167
                        7210    7222*   7227    7271    7305    7319    7371    7467*   7472    7540    7545*   7599    7763
                        7859    7878    8153    8301    8411    8455    8868   10791   10818*  10819*  10839   10898   11132
                       13193#
OFFSET= 000115          1985#   8112
OFST  = 000004          2050#
OPER00  046676          8686   12228   12246#  12890
OPER02  046713         12229   12249#
OPER04  046724         12230   12251#  12891
OPER06  046740         12231   12253#
```

```
OPER10  046747          12232   12255#
OPER12  046765          12233   12258#  12892
OPER14  047001          12234   12261#
OPER16  047010          12235   12263#  12899
OPER20  047015          12236   12264#
OPER22  047027          12237   12266#
OPER24  047042           9846   12238   12268#
OPER26  047057          12239   12271#
OPER30  047075          12240   12274#
OPER32  047111          12241   12277#
OPER34  047135          12242   12281#
OPER36  047161          12243   12285#
OPER37  047205           6712    6719   12289#
OPER40  047241          12294#
OPER41  047241           7017    7023    7034    7040    7152    7158    7169    7175   12295#
OPER42  047276           5380   12300#
OPER43  047352           5433   12308#
OPER44  047425           5542   12316#
OPI     =  020000           2042#  10260   10299
OPIERR= 000020          10301   12151#
OPR001  047504           2829   12328#
OPR002  047533           2832    2843    2860   12332#
OPR003  047541           2839   12334#
OPR004  047571           2851   12339#
OPR005  047613           2805   12343#
OPR006  047664           3013   12350#
OPR007  050034           3010    3031   12368#
OPR008  050124           3090   12378#
OPR009  050207           3095   12387#
OPR010  050257           5481   12394#
OPR011  050411           4773    4921    5072   12410#
OPR012  050554           4774   12428#
OPR013  050560           4922   12429#
OPR014  050564           5073   12430#
OPR015  050570           4776    4923    5074   12431#
OPR016  050604           2824   12433#
OPR017  051362           7333    7492   12500#
OPR018  051477           8644   12514#
OPR019  051565          11063   12524#
OPR020  051730          11127   12541#
OPR021  052024          11155   12551#
OPR022  052063          11171   12557#
OPTFLG  001656           2684#   2822    2825*   3011*   3032*   3043*   3048    4770    4775*   4918    4924*   5069    5075*
                         5477    5479    5482*   5495    5508    5515    5528    6509    6521    6528    6557    7262    7330
                         7332*   7489    7491*   9514*   9525*   9537*   9564    9661    9816    9862   11070
OPTTST  033704           2875    3788    5534    6562    6577    8542   9500#
OR      =  000200        2017#
PACK    =  000103        1980#   3153
PARBK0= 000200           2690#   5477    6509    9514
PARBK1= 000100           2691#   5495    5508    5515    5528    6521    6528    6557    9525
PARM    001720           2172    2721#
PAT     =  000020        2066#   3380    7752
PATCH   060204          13191#
PAT02   045662          10953   11865#
PAT03   045722          10958   11885#
PAT04   045762          10963   11905#
```

```
PAT05   046022          10968   11925#
PAT06   046062          10973   11945#
PAT10   046122          10978   11968#
PAT11   046162          10983   11988#
PAT12   046222          10988   12008#
PAT13   046262          10993   12028#
PAT14   046322          10998   12048#
PAT15   046362          11003   12068#
PAT16   046422          11008   11011   12088#
PCA     = 004000         2073#
PCD     = 010000         2074#
PERHLR  033634           9479#   9516
PGE     = 002000         2020#  10329
PGERR   = 000020         7832   10331   12160#
PIP     = 020000         2058#
PIRQ    = 177772         1824#
PIRQVE= 000240           1918#
PROGEN  042642          11099   11126   11151   11155#  11172
PR0     = 000000         1841#   2883    2992    9653
PR1     = 000040         1842#
PR2     = 000100         1843#
PR3     = 000140         1844#
PR4     = 000200         1845#
PR5     = 000240         1846#   1955
PR6     = 000300         1847#
PR7     = 000340         1848#   2730    2904    2998    5503    6516    9504    9517    9529
PS      = 177776         1821#   1822
PSW     = 177776         1822#
PWRVEC= 000024           1913#   2750*   2751*   11754*  11755*  11764*  11770*  11782*  11783*
RDCHR   = 104410        11597   11833#
RDDATA= 000121           1987#   3959    4651    4717    4808    4876    4956    5024    5103    5175    5240    5313    5417
                         5611    5626    5695    5714    5786    5800    5867    5882    5952    5967    6026    6083    6141
                         6197    6253    6297    6324    6462    6605    6730    6765    6847    7397    7815    7893
RDGATE= 100000           2077#
RDHEAD= 000125           1989#   4123    6389    7683    7961    8272
RDLIN   = 104411        11671   11834#
RDOCT   = 104412         2833    2844    2861   11835#
RDSTHD  035114           4163    6394    9815#
RDY     = 000200         1998#   2932    9998   11068
REALBA  046502          10070*  12123#
REALCY  046504          12124#
REALSE  046510           6721   12126#
REALTR  046506           6711   12125#
REALUB  046500          10076*  12122#
REALWC  046476          10020*  10153   10241   12121#
RECAL   = 000113         1984#   3655    3825    3886    7671    7727    7949    8081    8235    8509    8578    8662    8816
                         9837   11101
REFMT   001664           2699#   2735*   4088*   4215*   4291*   4370*   4422*   4476*   4536*   4595*   4679*   6355*   6434*
                         6508*   6576*   6655*   6870*   8230*   8466*  11125   11137
REPNR   = 100000        10578   12222#
RESREG= 104414           9443    9542    9677    9896    9946   10160   10242   10246   10600   10620   11837#
RESTRT  001730           2170    2724#
RESVEC= 000010           1908#
RKASOF= 000016           1967#   8518    8586
RKBA    = 000004         1962#   7319*
RKCS1   = 000000         1960#   2907*   2932*   3787*   7323*   7702*   7788*   7890*   7986*   8383*   8442*   8517*   8585*
```

```
                    9835*   9849    9910*   9921*   9925    9927    9937*   9980   11067*  11068
RKCS2 = 000010       1964*   2929*   3308*   7322*   8441*   8605*   9828*   9856    9919*   9929    9978*  11065*  11066*
RKDA  = 000006       1963*   9830*
RKDB  = 000024       1969*   4120    9827
RKDCYL= 000020       1968*   9829*
RKDS  = 000012       1965*   8443
RKECPS= 000030       1973*
RKECPT= 000032       1974*
RKER  = 000014       1966*   7328    7486
RKMR1 = 000026       1970*   3575*   3623*   7973*   8057*   8164*   8311*   8316*   8321*   9794*   9795*   9920*  11074*
                    11078*
RKMR2 = 000034       1971*   9935
RKMR3 = 000036       1972*   9936
RKPRI   001622       2662*   2880*   5519    6532
RKSPAR= 000022       1975*   7824*
RKVEC   001620       2661*   2878*   2996
RKWC  = 000002       1961*   7320*   7887
RLS   = 000010       2013*   3211    3566
SAVREG= 104413       9390    9500    9612    9615    9618    9621    9624    9627    9630    9633    9636    9639    9642
                     9645    9648    9651    9815    9909   10096   10177   10186   11836*
SCLR  = 000040       2015*   8605    9978   11065
SCOP1 = 104415       3861    3869    3971    4116    4148    4155    4184    4268    4349    4801    6378    6401    6417
                     6724    6833    6863    6978    7001    7045    7111    7136    7180    7317    7340    7386    7411
                     7429    8706    8886   11838*
SCOP1$  034122       9551*  11838
SECERR= 000040       6716   10152   12171*
SEEK  = 000117       1986*   3721    3763    3844    3905    7346    7501    7562    7635    7746    7934    8858
SELDRV= 000101       1979*   3016    3055    3122    3202    3242    3292    3335    3379    3445    3486    3557    3605
                     3691    8025    8043    8197    8287    8330    8353    8674
SETCLK  001710       2174    2718*
SKI   = 000002       2030*
SKIERR= 000002       3865    7773   12130*
SPACE2  047501       3109    9426    9438   12327*
SPAR  = 020000       2005*   9927   10262
SPARER= 000001       3387   10264   12129*
SPDERR= 000200      10272   12136*
SPDLSS= 000020       2052*  10270
SRCHLD  001670       2704*  10844   10868*  10920*  11044
SRTFLG  001624       2663*   2718*   2721*   2724*   2727*   2827    3008    3183
SRTINS= 000010       2689*   2822    2825
SRTSPL= 000111       1983*
SR0   = 177572       1927*   9044    9059*   9066*  10852*  10860*  10878*  10930*  11040*
SR1   = 177574       1928*
SR2   = 177576       1929*
SR3   = 172516       1930*   9056*
SSINIT  035722       9969*  11839
STACK = 001100       1812*   2729    2742    2898    9453    9484   11156
START   001740       2163    2169    2727*
START1  001744       2719    2722    2725    2728*  11165
STKLMT= 177774       1823*
SVAL  = 100000       2060*   3136    3148    3219    3250    3266    3301    3312    3348    3417    3474    3662    3677
                     3701    8683    8699
SWR     001140       2224*   2740    2762*   2764    2770*   2777*   2791    9111    9125    9128    9143    9150    9322
                     9329    9364    9368    9445    9479    9551   10507   11441   11480   11535*  11762   11775*
SWREG   000176       2167*   2770    2791   11441   11480   11503
SWC   = 000001       1876*
```

```
SW00  = 000001            1866#   1876
SW01  = 000002            1865#   1875
SW02  = 000004            1864#   1874
SW03  = 000010            1863#   1873
SW04  = 000020            1862#   1872
SW05  = 000040            1861#   1871
SW06  = 000100            1860#   1870
SW07  = 000200            1859#   1869
SW08  = 000400            1858#   1868
SW09  = 001000            1857#   1867
SW1   = 000002            1875#
SW10  = 002000            1856#
SW11  = 004000            1855#
SW12  = 010000            1854#   9445
SW13  = 020000            1853#  10507
SW14  = 040000            1852#
SW15  = 100000            1851#
SW2   = 000004            1874#
SW3   = 000010            1873#
SW4   = 000020            1872#
SW5   = 000040            1871#
SW6   = 000100            1870#
SW7   = 000200            1869#
SW8   = 000400            1868#
SW9   = 001000            1867#   9551
S.ACLO= 000100            2095#
S.BRHM= 000100            2109#
S.BRKE= 040000            2131#
S.CART= 000400            2111#
S.DIB = 002000            2127#
S.DOOR= 000200            2110#
S.DRA = 000040            2081#
S.DROT= 020000            2102#
S.DRY = 000200            2083#
S.DSC = 040000            2090#
S.FLT = 000200            2096#
S.FORM= 001000            2085#
S.FWD = 002000            2113#
S.HDFL= 000200            2124#
S.HDHM= 000040            2108#   8370
S.ICYL= 000040            2094#
S.ILF = 000400            2097#
S.LIMD= 020000            2130#
S.LOAD= 010000            2115#
S.MHD = 000400            2125#
S.NMOV= 010000            2129#
S.OFF = 002000            2086#
S.PAR = 001000            2098#
S.PIP = 020000            2089#
S.PLO = 004000            2128#
S.REV = 004000            2114#
S.RTZ = 020000            2116#
S.SECT= 000020            2121#
S.SKI = 002000            2099#
S.SPIN= 010000            2088#
S.SPLS= 010000            2101#
```

```
S.SPOK= 001000  2112#
S.TYPE= 000400  2084#
S.UNLD= 040000  2117#
S.UNS = 040000  2103#
S.VV  = 000100  2082#
S.WCLK= 000040  2122#
S.WGAT= 000100  2123#
S.WLE = 004000  2100#
S.WRL = 004000  2087#
S.XCOK= 000020  2107#
S.XERR= 001000  2126#
TBITVE= 000014  1909#
*CHKOP= 104421  3028  3080  3145  3159  3208  3217  3247  3263  3298  3430  3493  3520  3563
                3611  3697  3748  3969  4113  4138  4164  4242  4275  4321  4355  4391  4444
                4505  4564  4578  4617  4647  4662  4713  4736  4799  4817  4873  4886  4949
                4966  5021  5035  5099  5114  5172  5186  5235  5255  5309  5326  5603  5622
                5637  5687  5710  5725  5773  5797  5858  5878  5893  5944  5963  5978  6037
                6094  6152  6208  6264  6308  6375  6395  6458  6473  6550  6601  6616  6677
                6741  6830  6860  6903  6918  6960  6975  7093  7108  7219  7226  7283  7314
                7338  7357  7383  7408  7497  7513  7519  7534  7559  7573  7578  7593  7616
                7646  7652  7667  7742  7868  7912  3003  8036  8062  8108  8124  8132  8147
                8189  8246  8253  8268  8283  8326  8367  8402  8420  8464  8532  8600  8667
                8680  8831  8850  8876  11112 11119 11147 11842#
TCHKWE= 104422  3130  3342  3386  3451  3580  3628  3856  3864  3912  4262  4343  4406  4459
                4520  5374  5427  5536  6335  6700  6776  6995  7130  7694  7772  7829  7904
                7977  8209  8343  11843#
*GETRK= 104420  3022  3061  3310  3527  3661  3676  3728  3769  3797  4142  8544  8610  8698
                9666  9844  9890  9990  10189 11841#
TIMCNT  001654  2683#  9566*  9595*  9598*  9657*  9832*
TKVEC = 000060  1916#  11418*  11419*
TLOADR= 104417  3017  3056  3124  3154  3204  3213  3244  3259  3294  3338  3382  3427  3447
                3482  3489  3516  3559  3569  3607  3617  3657  3693  3723  3765  3828  3847
                3889  3908  3966  4109  4134  4238  4255  4318  4336  4387  4402  4440  4455
                4501  4516  4560  4574  4613  4643  4658  4709  4732  4795  4814  4869  4883
                4945  4962  5017  5031  5095  5110  5168  5182  5231  5251  5305  5323  5371
                5424  5524  5599  5618  5633  5683  5706  5721  5769  5793  5807  5854  5874
                5889  5939  5959  5974  6033  6090  6148  6204  6260  6304  6331  6371  6454
                6469  6546  6597  6612  6673  6696  6737  6772  6826  6856  6899  6914  6956
                6971  6991  7089  7104  7126  7215  7232  7279  7310  7353  7378  7404  7477
                7508  7530  7549  7568  7589  7606  7642  7663  7678  7690  7734  7753  7768
                7822  7864  7883  7900  7941  7956  7968  8000  8032  8052  8088  8104  8120
                8143  8159  8205  8242  8264  8279  8294  8307  8339  8363  8398  8416  8460
                8511  8579  8664  8676  8818  8827  8839  8846  8860  8871  11108 11143 11840#
TPVEC = 000064  1917#
TRAPVE= 000034  1915#  2748*  2749*
TRKERR= 000020  6707  10149  12170#
*RTVEC= 000014  1910#
TSSINI= 104416  2989  3051  3117  3198  3238  3288  3331  3375  3553  3586  3601  3651  3717
                3759  3822  3840  3883  3901  3955  4091  4107  4132  4161  4216  4245  4292
                4325  4371  4423  4477  4538  4596  4693  4780  4856  4930  5004  5081  5154
                5216  5291  5360  5413  5474  5584  5668  5754  5839  5924  6011  6070  6127
                6184  6240  6293  6320  6359  6387  6435  6536  6578  6656  6727  6813  6843
                6884  6942  6981  7075  7115  7200  7300  7366  7394  7464  7631  7699  7723
                7757  7785  7811  7849  7930  7984  8021  8039  8077  8231  8349  8373  8380
                8438  8446  8498  8567  8657  8811  11093 11122 11839#
TSTLUP  004450  3225#  8482
```

TST1	003116	2896#	9010	9170
TST10	005060	3373#	9177	
TST100	025306	7847#	9233	
TST101	025500	7927#	9234	
TST102	025724	8019#	9235	
TST103	026052	8075#	9236	
TST104	026376	8228#	9237	
TST105	027130	8435#	9238	
TST106	027332	8493#	9239	
TST107	027552	8562#	9240	
TST11	006004	3551#	9178	
TST110	027766	8635#	9241	
TST111	030464	8799#	9242	
TST12	006122	3599#	9179	
TST13	006232	3649#	9180	
TST14	006474	3715#	9181	
TST15	006626	3757#	9182	
TST16	007036	3818#	9183	
TST17	007226	3881#	9184	
TST2	003234	2927#	9171	
TST20	007350	3944#	9185	
TST21	010136	4086#	9186	
TST22	010534	4213#	9187	
TST23	010746	4289#	9188	
TST24	011164	4368#	9189	
TST25	011310	4420#	9190	
TST26	011442	4474#	9191	
TST27	011620	4534#	9192	
TST3	003322	2987#	9172	
TST30	011752	4593#	9193	
TST31	012246	4690#	9194	
TST32	012442	4765#	9195	
TST33	012676	4946#	9196	
TST34	013064	4914#	9197	
TST35	013316	4995#	9198	
TST36	013504	5065#	9199	
TST37	013726	5145#	9200	
TST4	004346	3196#	9173	
TST40	014114	5214#	9201	
TST41	014306	5289#	9202	
TST42	014502	5358#	9203	
TST43	014706	5411#	9204	
TST44	015112	5472#	9205	
TST45	015572	5582#	9206	
TST46	015776	5666#	9207	
TST47	016202	5752#	9208	
TST5	004450	3236#	9174	
TST50	016420	5837#	9209	
TST51	016624	5922#	9210	
TST52	017030	6009#	9211	
TST53	017154	6068#	9212	
TST54	017300	6125#	9213	
TST55	017424	6182#	9214	
TST56	017550	6238#	9215	
TST57	017674	6291#	9216	
TST6	004626	3286#	9175	

```
TST60   017740          6318#   9217
TST61   020012          6353#   9218
TST62   020270          6432#   9219
TST63   020452          6506#   9220
TST64   020710          6574#   9221
TST65   021076          6653#   9222
TST66   021602          6807#   9223
TST67   022012          6882#   9224
TST7    004754          3329#   9176
TST70   022116          6938#   9225
TST71   022536          7071#   9226
TST72   023156          7198#   9227
TST73   023266          7260#   9228
TST74   024120          7462#   9229
TST75   024576          7629#   9230
TST76   025012          7720#   9231
TST77   025226          7809#   9232
TWAT1M= 104440         11857#
TWAT1S= 104435         11854#
TWAT11= 104431          3967    4110    4239    4319    4388    4441    4502    4561    4614    4644    6674    6827    7280
                       11850#
TWAT12= 104432         11851#
TWAT14= 104433         11852#
TWAT15= 104434          6305    6332    6372    7311    7478    7550    7607    8377   11144   11853#
TWAT16= 104423          3019    3058    3125    3156    3205    3214    3245    3260    3295    3339    3383    3448    3490
                        3517    3560    3577    3608    3625    3658    3694    3724    3766    3790    3830    3848    3891
                        3909    4135    7354    7509    7531    7569    7590    7643    7664    7679    7735    7754    7769
                        7826    7942    7957    8001    8033    8089    8105    8121    8144    8192    8206    8243    8265
                        8295    8340    8364    8399    8512    8524    8580    8592    8665    8677    8819    8828    8840
                        8847    8861   11109   11844#
TWAT2S= 104436          7691    8690   11855#
TWAT32= 104424          4256    4337    4403    4575    4659    4733    4815    4884    4963    5032    5111    5183    5252
                        5324    5619    5634    5707    5722    5794    5808    6470    6697    6738    6773    6857    6915
                        6972    6992    7105    7127    7233    7975    8059    8129    8165    8280   11845#
TWAT48= 104425          4456    4517    5875    5890    5960    5975    7325    7405    7901   11846#
TWAT64= 104426          6613    8323    8417    8461   11847#
TWAT8S= 104437          3669    3835    3896    7379    7703    7739    7791    7989    8093    8250    8395    8823    8843
                       11116   11856#
TWAT80= 104427          7359    7516    7576    7650    8873   11848#
TWAT96= 104430          4710    4796    4870    4946    5018    5096    5169    5232    5306    5372    5425    5525    5600
                        5684    5770    5855    5940    6034    6091    6149    6205    6261    6455    6547    6598    6900
                        6957    7090    7216    7865    7945   11849#
TYPDS = 104405          8991    8998   11828#
TYPE  = 104401          2785    2805    2824    2829    2832    2839    2843    2851    2860    3010    3013    3031    3090
                        3095    3107    3109    4773    4774    4776    4921    4922    4923    5072    5073    5074    5481
                        8644    8985    8992    8999    9324    9336    9339    9344    9355    9402    9403    9407    9408
                        9415    9426    9428    9438    9439    9441    9449    9481   10543   10544   10576   10577   11063
                       11127   11155   11171   11216   11310   11379   11435   11447   11501   11502   11505   11516   11526
                       11537   11556   11604   11610   11615   11619   11624   11625   11627   11630   11634   11700   11702
                       11784   11824#
TYPERR  033322          9354    9390#
TYPOC = 104402          2831    2842    2859    9338    9343    9423    9482   11504   11825#
TYPON = 104404         11827#
TYPOS = 104403         11826#
T.ASOF  001556          2638#   3400    3433    3456    3496    3528    8519*   8519    9546    8586*   8587    9612    8880
                       13077   13087
```

```
T.BA    001544          2633*  5388   5392   5442   5446   5544   5547   5551   5554   7011   7016   7029   7033
                        7146   7151   7164   7168  10067  10070  10155  13090
T.CS1   001540          2631*  3023   3062   3405   3439   3462   3502   3534   3730   3771   3798   4150   5381
                        5434   8833   8852   9667   9742   9980*  9981   9985   9998  10071  10195  10203  10220
                       10253  10521  13077  13087
T.CS2   001550          2635*  3025   3077   4143  10197  10254  13077  13087
T.DA    001546          2634*  10147  10150  10157  10159  13089
T.DB    001564          2641*
T.DCYL  001560          2639*  10144  10156  13089
T.DS    001552          2636*  3036   3041   3064   3066   3136   3148   3162   3168   3219   3250   3266   3301
                        3312   3348   3411   3417   3468   3474   3662   3677   3684   3701   8683   8699  10199
                       10255  13077  13087
T.ECPS  001570          2643*
T.ECPT  001572          2644*
T.ER    001554          2637*  10201  10256  13077  13087
T.MR1   001566          2642*  13077
T.MR2   001574          2645*  8370
T.MR3   001576          2646*
T.SPAR  001562          2640*
T.WC    001542          2632*  5395   5399   5449   5453   5558   5561   5564   5567   7020   7025   7037   7042
                        7155   7160   7172   7177  10017  10020  10056  10099  10154  13090
UBAERR= 000002         10075  12167*
UFE   = 000400          2018*  10321
UFERR = 000004          3583   3631  10323  12158*
UNLOAD= 000107          1982*
UNS   = 040000          2043*
UNSERR= 040000          8344  12143*
UPE   = 020000          2023*  10337
UPERR = 000100          5539  10339  12162*
VV    = 000100          2054*  3162
WATSRT  034436          9614   9617   9620   9623   9626   9629   9632   9635   9638   9641   9644   9647   9650
                        9653*
WCE   = 040000          2024*  10292
WCERR = 000001         10019  12166*
WCKERR= 000004          6997   7132  10294  12149*
WLE   = 004000          2040*
WLERR = 004000         12140*
WRDATA= 000123          1988*  4248   4329   4395   4448   4509   4568   4632   4697   4784   4859   4938   5007
                        5084   5158   5220   5295   5364   5487   5588   5672   5758   5843   5928   6447   6539
                        6590   6689   6888   6949   7082   7208   7303   7323   7369   7470   7538   7761   7857
                        7876   8151   8869
WRDCNT  001674          2706*  10846  10874*  10922*  11046
WRDNUM  001672          2705*  10845  10875*  10921*  11045
WRHEAD= 000127          1990*  4095   4220   4296   4375   4427   4481   4542   4603   6361   6660   6817   7269
                        8299   8409   8453  11130
WRL   = 004000          2057*  3066
WRTCHK= 000131          1991*  6907   6964   6984   7097   7119   7225   7597
WRTGAT= 040000          2076*
SAPTHD  000224          2184   2190*
SASTAT= ****** U        9280   9295
SATYC   032556          9251   9253*
SATY1   032532          9249*
SATY3   032540          9250*  11201
SATY4   032550          9252*  9360
SAUTOB  001134          2221*  2795*  11494  11651
SBASE   001352          2319*  2830   2838*  2881   2905
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$BOARDR  001122          2216#
$BOOTAT  001126          2218#
$BELL    001266          2270#  9324    9380    11447   11644
$CDWI    001356          2321#
$CHARRC  043176         11218#  11228#  11235   11244#  11249#
$CKSWR   044152         11480#  11832
$CMTAG   001100          2204#   2737    2738    2746    2752    2753    2754
$CM1   = 000020          2236#   2237#   2238#   2239#   2240#   2241#   2242#   2243#   2244#   2245#   2246#   2247#   2248#
                         2249#   2250#   2251#   2252#
$CM2   = 000040          2236#   2237#   2238#   2239#   2240#   2241#   2242#   2243#   2244#   2245#   2246#   2247#   2248#
                         2249#   2250#   2251#   2252#
$CM3   = 000020          2234#   2236
$CM4   = 000020          2252#   2253#   2254#   2255#   2256#   2257#   2258#   2259#   2260#   2261#   2262#   2263#   2264#
                         2265#   2266#   2267#   2268#
$CNTLC   045050         11435   11516   11644#
$CNTLG   045062         11501   11646#
$CNTLJ   045055         11526   11619   11645#
$CORE    031660          9042    9069#
$CPUOP   001324          2293#
$CRLF    001273          2272#   8999    9344    9355    9380    9402    9407    9415    9428    9441    10543   10576   11217
                        11252   11537   11624   11644   11705
$CROUT   031710          9069    9076#
$DBLK    043416         11276   11310   11318#
$DEVCT   001306          2284#
$DEVM    001354          2320#   2999    3982#   4031#   8475    8497    8566    8637    8648    9803
$DOMGN   031442          8981    9002    9008#
$DTBL    043406         11279   11314#
$ENDAD   031432          2783    9004#   9375
$ENDCT   031276          2752    8983#
$ENULL   031446          9011#
$ENV     001316          2289#   2789    9258    9282    9357   11196
$ENVM    001317          2290#   2775    2800    3002    9260   11198   11203
$EOP     031242          2807    8645    8973#   9243    9455    9485   11162
$EOPCT   031270          2752#   2806#   8980#   8984    9454#   9483#  11161#
$ERFLG   001103          2207#   2938    3973    4011#   9095    9139    9141    9147#   9169    9319#   9380    9553    11157#
$ERMAX   001115          2213#   2755#   9141    9164#   9169
$ERPOR   033000          2746    9110    9317#
$ERRPC   001116          2214#   9326#   9327#   9328    9334#   9335#   9337    9341#   9342    9380    13077   13090   13086
                        13094
$ERRTB   001360          2338#   9399
$ERTTL   001112          2211#   8996    9000#   9325#   9380
$ESCAP   001264          2269#   2754#   2882#   9163#   9371    9373    9380    9673#   9674#   9893#   9894#   9993#   9994#
                        10618#  10619#  11158#
$ETABL   001316          2288#
$ETEND   001360          2196    2322#
$FATAL   001300          2281#   9286#
$FFLG    032776          9249#   9252#   9280    9289#   9297#
$FILLC   001156          2232#   11221   11252
$FILLS   001155          2231#   11252
$GOADR   001120          2215#
$GOOAT   001124          2217#
$GET42   031422          9001#
$GTSWR   044242         11502#  11830
$HD    = 000000          1797
$HIBTS   000224          2191#
$HIOCT   045250         11693#  11704#
```

```
$ICNT   001104          2208#  9154*  9155   9157*  9168
$ILLUP  045506         11754  11770  11787#
$INTAG  001135          2222#  11499* 11518  11538  11651
$ITEMB  001114          2212#  9328*  9359   9380   9393
$KTNEX  031652          9043   9068#
$KTOUT  031642          9060   9065#
$KTII   031510          2812   9041#  9045*  9068*
$LF     001274          2273#  9380   11252  11634  11644  11705
$LFLG   032775          9290*  9296#
$LPADR  001106          2209#  2756*  9136*  9145*  9161*  9166   9168
$LPERR  001110          2210#  2757*  3838*  3899*  3954*  4104*  4118*  4158*  4244*  4324*  4642*  4778*  4853*
                        4927*  5001*  5078*  5151*  6356*  6384*  6725*  6810*  6840*  6979*  7113*  7237*  7310*
                        8653*  8808*  9145   9162*  9168   9370   9555
$LSTAD  031754          2810   9081*  9088#
$LSTBK  031756          2803   2808   9082*  9089#
$MADR1  001330          2306#  2821*
$MADR2  001334          2310#
$MADR3  001340          2313#
$MADR4  001344          2316#
$MAIL   001276          2192   2196   2279#  2774   2789   9160   9357   11196
$MAMS1  001326          2300#  2820*  4767   4848   4916   4997   5067   5147
$MAMS2  001332          2308#
$MAMS3  001336          2311#
$MAMS4  001342          2314#
$MBADR  000226          2192#
$MFLG   032774          9250*  9256   9291*  9295#
$MNEW   045100         11505  11649#
$MSGAD  001312          2286#  9266*  9269
$MSGLG  001314          2287#  9271*
$MSGTY  001276          2280#  9264   9272*  9284   9288*
$MSWR   045067         11502  11647#
$MTYP1  001327          2301#  2819*
$MTYP2  001333          2309#
$MTYP3  001337          2312#
$MTYP4  001343          2315#
$MXCNT  032302          9158   9168#
$NULL   001154          2230#  11223  11252
$NWTST= 000001          2889#  2891   2920#  2922   2950#  2952   3189#  3191   3236#  3228   3278#  3280   3322#
                        3324   3355#  3357   3541#  3543   3588#  3590   3636#  3638   3707#  3709   3750#  3752
                        3808#  3810   3874#  3876   3920#  3922   4077#  4079   4197#  4199   4279#  4281   4359#
                        4361   4412#  4414   4464#  4466   4525#  4527   4584#  4586   4681#  4683   4755#  4757
                        4836#  4838   4904#  4906   4985#  4987   5055#  5057   5135#  5137   5206#  5208   5279#
                        5281   5351#  5353   5404#  5406   5458#  5460   5574#  5576   5657#  5659   5744#  5746
                        5828#  5830   5913#  5915   5998#  6000   6057#  6059   6114#  6116   6171#  6173   6227#
                        6229   6283#  6285   6310#  6312   6344#  6346   6424#  6426   6493#  6495   6565#  6567
                        6639#  6641   6799#  6801   6874#  6876   6920#  6922   7053#  7055   7188#  7190   7241#
                        7243   7446#  7448   7620#  7622   7711#  7713   7801#  7803   7836#  7838   7917#  7919
                        8008#  8010   8064#  8066   8214#  8216   8423#  8425   8486#  8488   8554#  8556   8620#
                        8622   8784#  8786
$OCNT   043650         11351* 11380* 11393#
$OMODE  043652         11346* 11350* 11355  11358* 11369* 11395#
$OVER   032266          9112   9137   9146   9156   9165#
$PASS   001304          2283#  2774*  3000   3947   8641   8977*  8978*  8989   9011   9152   9169
$PASTM  000232          2194#
$POWER  045514         11785  11790#
$PWRDN  045346          2750   11754# 11782
```

```
$PURMG  045502          11785#
$PURUP  045420          11764   11770#
$QUES   001272           2271#   9380   11252   11556   11627   11644   11702   11705
$RDCHR  044514          11569#  11833
$RDDEC= ****** J        11836
$RDLIN  044604          11592#  11834
$RDOCT  045112          11666#  11835
$RDSZ = 000010          11585#
$REGAC  001160           2234#
$REG0   001162           2236#
$REG1   001164           2237#
$REG10  001202           2244#   5386*   5391*   5398*   5438*   5445*   5452*   5546*   5553*   5560*   5566*   6710*   6720*
                         7015*   7024*   7032*   7041*   7150*   7159*   7167*   7176*   9911    9944   10240*  10869*  13091
                        13095
$REG11  001204           2245#   5385*   5392*   5399*   5439*   5446*   5453*   5547*   5554*   5561*   5567*   6711*   6721*
                         7016*   7025*   7033*   7042*   7151*   7160*   7168*   7177*  10241*  10870*  13091   13095
$REG12  001206           2246#   9916   10871*  13091   13095
$REG13  001210           2247#  13091
$REG14  001212           2248#  13091
$REG15  001214           2249#  13091
$REG16  001216           2250#  13091
$REG17  001220           2251#  13091
$REG2   001166           2238#
$REG3   001170           2239#
$REG4   001172           2240#
$REG5   001174           2241#
$REG6   001176           2242#
$REG7   001200           2243#
$RESRE  045310          11738#  11837
$RTNAD  031444           9010#
$R2R  = ****** J        11838
$SAVRE  045252          11722#  11836
$SAVR6  045512          11763*  11771   11772*  11773*  11789#
$SCOPE  031760           2744    9104#
$SETUP= 000137           2716#   2743    2744    2746    2748    2750    2752    2753    2754    2756    2783    2786    9975
                         9105    9318    9367    9375   11439   11444   11445   11475   11651
$SIZE   031452           2802    9029#
$SIZEX  031714           9067    9077#
$STJP = 177777           2716#
$SVLAC  032232           9120    9159#
$SWR  = 167400           1785#   1797    1801    1802    1803    1804    1805    1806    1807    1808    2268    2269    2270
                         2753    2754    2756    2757    2897    2928    2988    3197    3237    3287    3330    3374    3552
                         3600    3650    3716    3758    3819    3882    3945    4087    4214    4290    4369    4421    4475
                         4535    4594    4691    4766    4847    4915    4996    5066    5146    5215    5290    5359    5412
                         5473    5583    5667    5753    5838    5923    6010    6069    6126    6183    6239    6292    6319
                         6354    6433    6507    6575    6654    6808    6883    6939    7072    7199    7261    7463    7630
                         7721    7810    7848    7928    8020    8076    8229    8436    8494    8563    8636    8800    8970
                         8976    9003    9009    9011    9096    9097    9098    9099    9100    9106    9123    9125    9126
                         9139    9140    9141    9148    9149    9150    9162    9165    9168    9309    9310    9311    9312
                         9313    9322    9329    9364    9368    9380   11786
$SWREG  001320           2291#   2777
$SWRMK= 000000           1808    1809    9100    9101    9129
$SWOBT  032304           9135    9169#   9673    9893    9993   10618
$TERM = 000102          11858#
$TESTN  001302           2282#   9160*   9391*   9392*   9671    9891    9991   10616   11091*  13077   13080   13086   13094
$TIMES  001262           2268#   2753*   2897*   2928*   2988*   3197*   3237*   3287*   3330*   3374*   3552*   3600*   3650*
```

```
                    3716*  3758*  3819*  3882*  3945*  4087*  4214*  4290*  4369*  4421*  4475*  4535*  4594*
                    4691*  4766*  4847*  4915*  4996*  5066*  5146*  5215*  5290*  5359*  5412*  5473*  5583*
                    5567*  5753*  5838*  5923*  6010*  6069*  6126*  6183*  6239*  6296*  6319*  6354*  6433*
                    6507*  6575*  6654*  6808*  6883*  6939*  7072*  7199*  7261*  7463*  7630*  7721*  7810*
                    7848*  7928*  8020*  8076*  8229*  8436*  8469*  8494*  8563*  8636*  8800*  8976*  9148*
                    9155   9158*  9168
$TKB     001146     2227*  11399  11420  11431  11456  11484  11511
$TKCNT   043654    11400*  11415* 11445  11462* 11576  11578*
$TKINT   043664     2734    3780   8537  11415* 11436  11497
$TKQEN=  043663    11404*  11470  11581
$TKQIN   043656    11401*  11416* 11417  11468* 11469* 11470  11472*
$TKQOU   043660    11402*  11417* 11579  11580* 11581  11583*
$TKQSR   043662    11403*  11416  11472  11583
$TKS     001144     2226*  11399  11421* 11452* 11454  11460* 11482  11498* 11508  11520* 11540*
$TKSRV   043734    11418   11431*
$TMPC    001222     2252*  2834*  2836   2838   2845*  2847   2850   2862*  2864   2866   2868*  2869*  2870*
                    2871*  2872*  2874   9979*  9983*
$TMP1    001224     2253*  8647*  8723*  8748   8759   8781*  8802*  8903*  8928   8939   8961*  10405  10784*
                   10786*  10788  10802* 10818  10819  10823  10825*
$TMP10   001242     2260*  10179* 10359  10461  10591
$TMP11   001244     2261*  10180* 10440  10585
$TMP12   001246     2262*  10181* 10420  10580
$TMP13   001250     2263*  10183* 10187* 10217  10356  10374  10418  10438  10459  10510  10565
$TMP14   001252     2264*  10533* 10552  10556* 10558  10562* 10583  10587* 10589  10593*
$TMP15   001254     2265*  10534* 10539*
$TMP16   001256     2266*  9763*  9768*
$TMP17   001260     2267*  9765*  9773   9775*
$TMP2    001226     2254*  10406
$TMP3    001230     2255*  9918*  9932*  9947
$TMP4    001232     2256*  3015*  3034   3038*
$TMP5    001234     2257*  10641* 10651* 10658
$TMP6    001236     2258*  10638* 10659
$TMP7    001240     2259*  3114*  3178*  3181   10639* 10660
$TN    = 000112     1786*  1797   2889   2897*  2920   2928*  2950   2988*  3189   3197*  3226   3237*  3278
                    3287*  3322   3330*  3355   3374*  3541   3552*  3588   3600*  3636   3650*  3707   3716*
                    3750   3758*  3808   3819*  3874   3882*  3920   3945*  4077   4087*  4197   4214*  4279
                    4290*  4359   4369*  4412   4421*  4464   4475*  4525   4535*  4584   4594*  4681   4691*
                    4755   4766*  4836   4847*  4904   4915*  4985   4996*  5055   5066*  5135   5146*  5206
                    5215*  5279   5290*  5351   5359*  5404   5412*  5458   5473*  5574   5583*  5657   5667*
                    5744   5753*  5828   5838*  5913   5923*  5998   6010*  6057   6069*  6114   6126*  6171
                    6183*  6227   6239*  6283   6292*  6310   6319*  6344   6354*  6424   6433*  6493   6507*
                    6565   6575*  6639   6654*  6799   6808*  6874   6883*  6920   6939*  7053   7072*  7188
                    7199*  7241   7261*  7446   7463*  7620   7630*  7711   7721*  7801   7810*  7836   7848*
                    7917   7928*  8008   8020*  8064   8076*  8214   8229*  8423   8436*  8486   8494*  8554
                    8563*  8620   8636*  8784   8800*  9130   9170  11091
$TPB     001152     2229*  11241* 11252
$TPFLG   001157     2233*  11190  11252
$TPS     001150     2228*  11239  11252
$TRAP    045524     2748   11801*
$TRAP2   045546    11812*  11823
$TRP   = 000041    11816*  11825* 11826* 11827* 11828* 11829* 11830  11831* 11832  11833* 11834* 11835* 11836*
                   11837*  11838* 11839* 11840* 11841* 11842* 11843* 11844* 11845* 11846* 11847* 11848* 11849*
                   11850*  11851* 11852* 11853* 11854* 11855* 11856* 11857* 11858*
$TRPAD   045560    11806   11823* 11858
$TSTM    000230     2193*
$TSTNM   001102     2206*  8481*  8975*  9095   9132*  9159*  9160   9165   9169   9321   9380   9391   11096
```

```
$TTYIN  045040          11594   11595   11607   11625   11639   11643#
$TYPBN= ****** U        11829
$TYPDS  043202          11264#  11828
$TYPE   042720           9277   11190#  11816   11824
$TYPEC  043132          11220   11227   11234   11239#  11240   11542
$TYPEX  043200          11245   11247   11250#
$TYPOC  043452          11349#  11825
$TYPON  043466          11348   11351#  11827
$TYPOS  043426          11344#  11826
$UNIT   001310           2285#
$UNITM  000234           2195#
$USWR   001322           2292#
$VECT1  001346           2317#   2840    2849*   2850*   2853    2873*   2874*   2876    2880
$VECT2  001350           2318#
$XTSTR  032004           9114#
$$GET4= 000000           9003#
$$SWDB= 000112           9169#   9170    9171#   9172#   9173#   9174#   9175#   9176#   9177#   9178#   9179#   9180#   9181#
                         9182#   9183#   9184#   9185#   9186#   9187#   9188#   9189#   9190#   9191#   9192#   9193#   9194#
                         9195#   9196#   9197#   9198#   9199#   9200#   9201#   9202#   9203#   9204#   9205#   9206#   9207#
                         9208#   9209#   9210#   9211#   9212#   9213#   9214#   9215#   9216#   9217#   9218#   9219#   9220#
                         9221#   9222#   9223#   9224#   9225#   9226#   9227#   9228#   9229#   9230#   9231#   9232#   9233#
                         9234#   9235#   9236#   9237#   9238#   9239#   9240#   9241#   9242#   9243#
$OFILL  043651          11345*  11349*  11359   11394#
$40CAT  000000           2160#   9106    9331
.       = 064604         2145#   2161    2165#   2171#   2173#   2180    2181#   2183#   2185#   2203#   2274    2741    2756
                         2757    2799#   8988#   9011    9012#   9168    9169    9298#   9380    11252   11318#  11399   11403#
                        11404   11405#  11643#  11644   11651#  11705   11766   11788   11858   13189#  13190#  13191#  13192#
                        13193#
.$ASTA= ****** U         9250    9253
.$X   = 000224           2180#   2185
```

```
COMMEN  1919#
ENDCOM  1919#
ERROR   1813#  2914  2943  2990  3020  3029  3052  3059  3081  3118  3126  3134  3139  3146  3151
               3157  3160  3165  3172  3199  3206  3209  3215  3218  3222  3239  3246  3248  3255  3261
               3264  3271  3289  3296  3299  3306  3317  3332  3340  3346  3353  3376  3384  3390  3404
               3409  3415  3422  3431  3437  3443  3449  3455  3460  3466  3472  3478  3491  3494  3500
               3506  3518  3521  3532  3538  3554  3561  3564  3578  3584  3587  3602  3609  3612  3626
               3632  3652  3659  3666  3674  3681  3688  3695  3699  3705  3718  3725  3734  3746  3749
               3760  3767  3775  3795  3802  3823  3831  3836  3841  3849  3860  3868  3884  3832  3897
               3902  3910  3916  3956  3968  3970  3981  3991  4001  4008  4017  4030  4072  4092  4108
               4111  4114  4133  4136  4139  4147  4154  4162  4165  4166  4167  4173  4182  4217  4240
               4243  4246  4257  4266  4276  4293  4320  4322  4326  4338  4347  4356  4372  4389  4392
               4404  4410  4424  4442  4445  4457  4463  4478  4503  4506  4518  4524  4539  4562  4565
               4576  4579  4597  4615  4618  4645  4648  4660  4663  4669  4677  4694  4711  4714  4734
               4737  4743  4750  4781  4797  4800  4816  4818  4824  4831  4857  4871  4874  4885  4887
               4893  4900  4931  4947  4950  4964  4967  4973  4981  5005  5019  5022  5033  5036  5042
               5050  5082  5097  5100  5112  5115  5122  5130  5155  5170  5173  5184  5187  5193  5201
               5217  5233  5236  5253  5256  5267  5275  5292  5307  5310  5325  5327  5338  5346  5361
               5373  5378  5387  5393  5400  5414  5426  5431  5440  5447  5454  5475  5526  5540  5548
               5555  5562  5568  5585  5601  5604  5620  5623  5635  5638  5644  5652  5669  5685  5688
               5708  5711  5723  5726  5732  5740  5755  5771  5774  5795  5798  5809  5815  5823  5840
               5856  5859  5876  5879  5891  5894  5900  5908  5925  5941  5945  5961  5964  5976  5979
               5985  5993  6012  6035  6038  6044  6052  6071  6092  6095  6101  6109  6128  6150  6153
               6159  6167  6185  6206  6209  6215  6223  6241  6262  6265  6271  6279  6294  6306  6309
               6321  6333  6339  6360  6373  6376  6388  6396  6397  6398  6405  5413  6436  6456  6459
               6471  6474  6480  6488  6537  6548  6551  6579  6599  6602  6614  6617  6623  6631  6657
               6675  6678  6698  6704  6713  6722  6728  6739  6742  6748  6756  6774  6780  6786  6794
               6814  6828  6831  6844  6858  6861  6885  6901  6904  6916  6919  6943  6958  6961  6973
               6976  6982  6993  6999  7018  7026  7035  7043  7076  7091  7094  7106  7109  7116  7128
               7134  7153  7161  7170  7178  7201  7217  7220  7234  7237  7281  7284  7301  7312  7315
               7326  7334  7339  7355  7358  7360  7367  7380  7384  7395  7406  7409  7417  7425  7465
               7484  7493  7498  7510  7514  7517  7520  7532  7535  7556  7560  7570  7574  7577  7579
               7591  7594  7613  7617  7632  7644  7647  7651  7653  7665  7668  7680  7692  7698  7700
               7704  7724  7736  7740  7743  7755  7758  7770  7776  7786  7797  7812  7827  7833  7850
               7866  7869  7902  7908  7913  7931  7943  7946  7958  7976  7981  7985  7990  8002  8004
               8022  8034  8037  8040  8060  8063  8078  8090  8094  8106  8109  8122  8125  9130  8133
               8145  8148  8166  8190  8194  8207  8213  8232  8244  8247  8251  8254  8266  8269  8281
               8284  8296  8324  8327  8341  8347  8350  8365  8368  8374  8331  8386  8400  8403  8418
               8421  8439  8447  8462  8465  8499  8513  8529  8533  8551  8568  8581  8597  8601  8617
               8658  8666  8668  8678  8681  8687  8696  8704  8812  8820  8824  8829  8832  8841  8844
               8848  8851  8862  8874  8877  8884  11094 11110 11113 11117 11120 11123 11145 11148
ESCAPE  1919#
GETPRI  1919#  9037
GETSWR  1919#  2786#
INITSS  2712#  4106  4131  4160  4693  4779  4855  4929  5003  5080  5153  5291  5360  5413  5474
               5584  5668  5754  5839  5924  6011  6070  6127  6184  6240  6293  6320  6358  6386  6435
               6536  6578  6656  6726  6812  6842  6884  6942  6981  7075  7115  7200  7299  7365  7393
               7464  7631  7699  7723  7757  7785  7811  7849  7930  7983  8021  8039  8077  8231  8349
               8373  8380  8438  8446  8498  8567  8656  8810  11093 11122
LOADLS  2714#  4696  4716  4807  4858  4955  5006  5102  5157  5294  5312  5485  5587  5610  5625
               5671  5694  5713  5757  5785  5799  5842  5866  5881  5927  5951  5966  6025  6082  6140
               6196  6252  6296  6323  6446  6461  6538  6589  6604  6659  6688  6729  6764  6845  6887
               6906  6948  6963  6983  7081  7096  7118  7207  7224  7302  7345  7368  7469  7500  7522
               7537  7561  7581  7596  7634  7655  7670  7682  7726  7745  7760  7814  7856  7875  7892
               7933  7948  7960  7992  8024  8042  8080  8096  8111  8135  8150  8196  8234  8256  8271
               8286  8298  8329  8352  8390  8408  8452  11100 11129
```

```
MSG     2889#  2891   2920#  2922   2949#  2952   3189#  3191   3226#  3228   3278#  3280   3322#  3324   3355#
        3357   3541#  3543   3588#  3590   3636#  3638   3707#  3709   3750#  3752   3808#  3810   3874#  3876
        3920#  3922   4077#  4079   4197#  4199   4279#  4281   4359#  4361   4412#  4414   4464#  4466   4525#
        4527   4584#  4586   4681#  4683   4755#  4757   4836#  4838   4904#  4906   4985#  4987   5055#  5057
        5135#  5137   5206#  5208   5279#  5281   5351#  5353   5404#  5406   5458#  5460   5574#  5576   5657#
        5659   5744#  5746   5828#  5830   5913#  5915   5998#  6000   6057#  6059   6114#  6116   6171#  6173
        6227#  6229   6283#  6285   6310#  6312   6344#  6346   6424#  6426   6493#  6495   6565#  6567   6639#
        6641   6799#  6801   6874#  6876   6920#  6922   7053#  7055   7188#  7190   7241#  7243   7446#  7448
        7620#  7622   7710#  7713   7801#  7803   7835#  7838   7916#  7919   8008#  8010   8064#  9066   9214#
        8216   8423#  8425   8486#  8488   8554#  8556   8620#  8622   8784#  8786
MULT    1919#
NEWTST  1919#  2889   2920   2950   3189   3226   3278   3322   3355   3541   3589   3636   3707   3750   3808
        3874   3920   4077   4197   4279   4359   4412   4464   4525   4584   4681   4755   4836   4904   4985
        5055   5135   5206   5279   5351   5404   5458   5574   5657   5744   5828   5913   5999   6057   6114
        6171   6227   6283   6310   6344   6424   6493   6565   6639   6799   6874   6920   7053   7188   7241
        7446   7620   7711   7801   7836   7917   8008   8064   8214   8423   8486   8554   8620   8784
OPCHK   2716#  4713   4736   4799   4817   4873   4886   4949   4966   5021   5035   5099   5114   5172   5186
        5309   5326   5603   5622   5637   5687   5710   5725   5773   5797   5858   5879   5893   5944   5963
        5978   6037   6094   6152   6208   6264   6308   6375   6458   6473   6550   6601   6616   6677   6741
        6830   6860   6903   6918   6960   6975   7093   7108   7219   7236   7283   7314   7337   7357   7382
        7408   7496   7513   7519   7534   7558   7573   7578   7593   7615   7646   7652   7667   7742   7868
        7911   8003   8036   8062   8108   8124   8132   8147   8189   8246   8253   8268   8283   8326   8367
        8402   8420   8464   8531   8599   8667   8680   8830   8849   8876   11112  11119  11147
POP     1919#  9292   9293   9705   9798   9995   10861  10880  10884  11048  11305  11694  11743  11775  11776
PUSH    1919#  9253   9255   9276   9694   9785   9969   10750  10761  11264  11668  11723  11756  11762
REPORT  1919#
RESDC   2713#  4744   4825   4894   4975   5044   5124   5195   5340   5646   5734   5817   5902   5987   6046
        6103   6161   6217   6273   6407   6482   6625   6750   6788   7419
RKLOAD  2715#  4709   4732   4795   4814   4869   4883   4945   4962   5017   5031   5095   5110   5168   5182
        5305   5323   5371   5424   5524   5599   5618   5633   5683   5706   5721   5769   5793   5807   5854
        5874   5889   5939   5959   5974   6033   6090   6148   6204   6260   6304   6331   6454   6469   6546
        6597   6612   6673   6696   6737   6772   6826   6856   6899   6914   6956   6971   6991   7089   7104
        7126   7215   7232   7279   7310   7353   7404   7508   7530   7568   7589   7642   7663   7678   7690
        7734   7753   7768   7864   7900   7941   7956   8000   8032   8088   8104   8120   8143   8205   8242
        8264   8279   8294   8339   8362   8398   8416   8460   8511   8579   8664   8676   8818   8827   9839
        8846   8860   11108  11143
SCOPE   1814#  2896   2927   2987   3196   3236   3286   3329   3373   3551   3599   3649   3715   3757   3818
        3881   3944   4086   4213   4289   4368   4420   4474   4534   4593   4690   4765   4846   4914   4995
        5065   5145   5214   5289   5358   5411   5472   5582   5666   5752   5837   5922   6009   6068   6125
        6182   6238   6291   6318   6353   6432   6506   6574   6653   6807   6882   6939   7071   7198   7260
        7462   7629   7720   7809   7847   7927   8019   8075   8228   8435   8468   8493   8562   8635   8799
        8974
SETPRI  1919#  11572
SETTRA  11816#  11825  11826  11827  11828  11830  11832  11833  11834  11835  11836  11837  11838  11839  11840
        11841  11842  11843  11844  11845  11846  11847  11848  11849  11850  11851  11852  11953  11854  11855
        11856  11857
SETUP   1919#  2736
SKIP    1919#
SLASH   1919#
SPACE   1919#
STARS   1919#  2177   2179   2186   2199   2274   2277   2889   2895   2920   2926   2950   2986   3189   3195
        3226   3235   3278   3285   3322   3328   3355   3372   3541   3550   3588   3598   3636   3648   3707
        3714   3750   3756   3808   3817   3874   3880   3920   3943   4077   4085   4197   4212   4279   4288
        4359   4367   4412   4419   4464   4473   4525   4533   4584   4592   4681   4689   4755   4764   4836
        4845   4904   4913   4985   4994   5055   5064   5135   5144   5206   5213   5279   5288   5351   5357
        5404   5410   5458   5471   5574   5581   5657   5665   5744   5751   5828   5936   5913   5921   5993
```

```
        6008   6057   6067   6114   6124   6171   6181   6227   6237   6283   6290   6310   6317   6344   6352
        6424   6431   6493   6505   6565   6573   6639   6652   6799   6806   6874   6881   6920   6937   7053
        7070   7188   7197   7241   7259   7446   7461   7620   7628   7711   7719   7801   7808   7836   7846
        7917   7926   8008   8018   8064   8074   8168   8169   8182   8183   8214   8227   8423   8434   8486
        8492   8554   8561   8620   8634   8784   8798   8966   9015   9092   9248   9305   9380   9389   9495
        9545  10623  10636  10664  11052  11175  11254  11321  11398  11475  11490  11561  11585  11654  11707
       11752  11768  11795
SWRSU   1919#  2758#
TRMTRP 11816#  11858
TYPBIN  1919#
TYPDEC  1919#  8989   8996
TYPNAM  1785#  1919#  2779
TYPNUM  1919#
TYPOCS  1919#
TYPOCT  1919#  2830   9337   9342  11503
TYPTXT  1919#  8985   8992
$$CMRE  2197#  2236   2237   2238   2239   2240   2241   2242   2243   2244   2245   2246   2247   2248   2249
        2250   2251
$$CMTM  2197#  2252   2253   2254   2255   2256   2257   2258   2259   2260   2261   2262   2263   2264   2265
        2266   2267
$$ESCA  1919#
$$NEWT  1919#  2889   2920   2950   3189   3226   3278   3322   3355   3541   3588   3636   3707   3750   3808
        3874   3920   4077   4197   4279   4359   4412   4464   4525   4584   4681   4755   4836   4904   4985
        5055   5135   5206   5279   5351   5404   5458   5574   5657   5744   5828   5913   5998   6057   6114
        6171   6227   6283   6310   6344   6424   6493   6565   6639   6799   6874   6920   7053   7188   7241
        7446   7620   7711   7801   7836   7917   8008   8064   8214   8423   8486   8554   8620   8784
$$SET  11816#  11825  11826  11827  11828  11830  11832  11633  11834  11835  11836  11837  11838  11839  11840
       11841  11842  11843  11844  11845  11846  11847  11848  11849  11850  11851  11852  11853  11854  11855
       11856  11857
$$SETM  2774#
$$SKIP  1919#
.EQUAT  1785#  1809
.HEADE  1785#  1787
.KTII   1785#  1919
.SETUP  1785#  2716
.SWRHI  1785#  1797
.SWRLO  1785#  1809#
.SACTI  1785#
.SAPTB  2275#
.SAPTH  1785#  2175
.SAPTY  1785#  9246
.SCMTA  1785#  2197
.SEOP   1785#  8964
.SERRO  1785#  9303
.SERRT  1785#
.SPOWE  1785#  11750
.SRDCC  1785#  11652
.SREAC  1785#  11396
.SSAVE  1785#  11705
.SSCOP  1785#  9090
.SSIZE  1785#  9013
.STRAP  1785#  11793
.STYPD  1785#  11252
.STYPE  1785#  11173
.STYPO  1785#  11319
.S40CA  1785#  2143
```

. ABS. 064604   000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DZR6KC,DSKZ:DZR6KC.SEQ/CRF/SOL/DOC=DZR6KC
RUN-TIME: 103 112 10 SECONDS
RUN-TIME RATIO: 398/226=1.7
CORE USED: 30K (59 PAGES)

DOCUMENT PAGES: 268